PART TWO

ELEMENTARY

PROCESS DYNAMICS AND CONTROL

A. Johnson
W. de Koning

Delftse Uitgevers Maatschappij B.V. — 1977
part two

ELEMENTARY PROCESS DYNAMICS AND CONTROL

PREFACE

This second, and final part of the book, introduces processes dynamics and control as an introduction to some of the simple techniques and approaches to problem solving. A number of the problems in Part I are reconsidered in the numbers of worked examples, questions and solutions. The topics covered in this elementary text will give the student a good understanding of the subject.

The contents of Part II are arranged as follows: a brief introduction to the study of dynamical systems; the solution of linear ordinary differential equations; the solution of linear systems of equations; the solution of a linear system with constant coefficients; and the solution of a linear system with variable coefficients. A. Johnson

A. Johnson
W. de Koning
PREFACE

This second, and final, volume of elementary process dynamics and control is an introduction to some of the more modern topics and approaches to problem solving. A knowledge of the material of Part I is assumed and constant reference is made to it. As previously, there are a number of worked examples, questions and bibliographical notes concerning the references in each chapter.

The contents of Part II range from the state space representation of dynamical systems to the control of chemical reactors. Some novel, as far as is known, proofs of elementary relationships appear, for example concerning multivariable systems in Chapter 13. Furthermore, half of Chapter 14 is devoted to the study of the structure of mathematical models of systems—a topic which is not treated elsewhere at this elementary level. It has been included in the firm belief that it will attain a significant importance in the future.

Once again, it is a pleasure to record the encouragement, help and tolerance shown by so many people, and to express gratitude to the VSSD for their support and effort. The fact that this second part represents one of the products of a most stimulating and rewarding collaboration between two engineers originally from different disciplines and countries merits also, we feel, note.

Delft, January 1977
PLATE II

The purpose of Part II is to give the reader a comprehensive view of the various components of a computer system and their interrelations. The first chapter, on the hardware, provides an overview of the various components and their functions. The second chapter, on the software, explores the different types of software used in computer systems and their roles. The third chapter, on the operating systems, discusses the various operating systems and their importance in computer systems.

The information presented in this part is intended to provide a comprehensive understanding of computer systems and their components. It is hoped that this part will serve as a valuable resource for anyone interested in computer systems.

February 1977

Copyright © 1973
Verlagting van Normen
voor Standaardiserings
in Den
Eenheden
Bilt, Netherlands
Set in Bask pica
contents

Preface iii
Contents v
Additional nomenclature Part 2 x
Additional symbols Part 2 x

12. THE STATE SPACE APPROACH 129
12.1. Introduction 129
12.2. The state and output equations 129
12.3. Disturbances 134
12.4. Relationship of the state space description with other model descriptions 136
12.4.1. Differential operator form of model 136
12.4.2. Transfer function model 137
12.5. Canonical forms 137
12.5.1. Jordan canonical form 138
12.5.2. The phase variable canonical form 139
12.6. The dynamic response of the linear homogeneous state model 141
12.7. The dynamic response of the linear non-homogeneous state model 143
12.8. Eigenvalues, time constants and stability 144
12.9. References and bibliographical notes 146

13. MULTIVARIABLE SYSTEMS 147
13.1. Introduction 147
13.2. Multivariable state space representations 147
13.3. System matrix representation 149
13.4. Closed-loop systems 155
13.5. Poles and zeros of multivariable systems 160
13.6. Interaction and pole placement 163
13.7. Reconstructing the state 167
13.8. References and bibliographical notes 171

14. SIMULATION AND STRUCTURAL ANALYSIS 172
14.1. Introduction 172
14.2. Analogue simulations 173
14.3. Digital computer simulations 179
14.4. The structural analysis of systems 185
14.4.1. The macroscale: assigning distinct representatives 186
14.4.2. Partitioning into subsystems 189
14.4.3. Disjointing the model 194
14.4.4. The hierarchical structure of a system 195
14.4.5. The macroscale 197
14.5. References and bibliographical notes 197

15. DYNAMIC OPTIMIZATION 201
15.1. Introduction 201
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.2</td>
<td>Variational calculus</td>
<td>202</td>
</tr>
<tr>
<td>15.2.1</td>
<td>Optimization without constraints</td>
<td>202</td>
</tr>
<tr>
<td>15.2.2</td>
<td>Optimization with constraints</td>
<td>203</td>
</tr>
<tr>
<td>15.3</td>
<td>Pontryagin's maximum principle</td>
<td>206</td>
</tr>
<tr>
<td>15.4</td>
<td>Dynamic programming</td>
<td>209</td>
</tr>
<tr>
<td>15.5</td>
<td>Linear optimal control systems</td>
<td>213</td>
</tr>
<tr>
<td>15.5.1</td>
<td>Optimal feedback laws</td>
<td>213</td>
</tr>
<tr>
<td>15.5.2</td>
<td>State estimation</td>
<td>216</td>
</tr>
<tr>
<td>15.6</td>
<td>References and bibliographical notes</td>
<td>219</td>
</tr>
<tr>
<td>16</td>
<td>PROCESS CONTROL BY COMPUTER</td>
<td>221</td>
</tr>
<tr>
<td>16.1</td>
<td>Introduction</td>
<td>221</td>
</tr>
<tr>
<td>16.2</td>
<td>Sampling and reconstruction of signals</td>
<td>222</td>
</tr>
<tr>
<td>16.2.1</td>
<td>Sampling and samplers</td>
<td>222</td>
</tr>
<tr>
<td>16.2.2</td>
<td>Sampling and continuous elements</td>
<td>223</td>
</tr>
<tr>
<td>16.2.3</td>
<td>Sampling and discrete elements</td>
<td>225</td>
</tr>
<tr>
<td>16.2.4</td>
<td>Reconstruction</td>
<td>226</td>
</tr>
<tr>
<td>16.3</td>
<td>The z-transformation</td>
<td>227</td>
</tr>
<tr>
<td>16.3.1</td>
<td>Impulse trains and continuous elements</td>
<td>227</td>
</tr>
<tr>
<td>16.3.2</td>
<td>Number series and discrete elements</td>
<td>229</td>
</tr>
<tr>
<td>16.4</td>
<td>Control algorithms</td>
<td>232</td>
</tr>
<tr>
<td>16.5</td>
<td>The state space description of sampled data systems</td>
<td>236</td>
</tr>
<tr>
<td>16.5.1</td>
<td>Sampled signals applied to discrete elements</td>
<td>236</td>
</tr>
<tr>
<td>16.5.2</td>
<td>Continuous elements preceded by a zero-order hold circuit</td>
<td>237</td>
</tr>
<tr>
<td>16.6</td>
<td>References and bibliographical notes</td>
<td>238</td>
</tr>
<tr>
<td>17</td>
<td>THE DYNAMICS AND CONTROL OF DISTILLATION COLUMNS</td>
<td>240</td>
</tr>
<tr>
<td>17.1</td>
<td>Introduction</td>
<td>240</td>
</tr>
<tr>
<td>17.2</td>
<td>Model of an ideal binary distillation plate</td>
<td>243</td>
</tr>
<tr>
<td>17.2.1</td>
<td>Overall mass balance round plate</td>
<td>243</td>
</tr>
<tr>
<td>17.2.3</td>
<td>Equilibrium relationship</td>
<td>244</td>
</tr>
<tr>
<td>17.2.4</td>
<td>Hydraulic relationship</td>
<td>245</td>
</tr>
<tr>
<td>17.3</td>
<td>Extension to a column model</td>
<td>246</td>
</tr>
<tr>
<td>17.4</td>
<td>Distributed parameter models</td>
<td>247</td>
</tr>
<tr>
<td>17.5</td>
<td>Models for control purposes</td>
<td>247</td>
</tr>
<tr>
<td>17.6</td>
<td>The basic principles of column control</td>
<td>248</td>
</tr>
<tr>
<td>17.7</td>
<td>Feedforward control</td>
<td>251</td>
</tr>
<tr>
<td>17.8</td>
<td>Analysis of the structure of a distillation column model</td>
<td>251</td>
</tr>
<tr>
<td>17.9</td>
<td>A survey of recent literature</td>
<td>257</td>
</tr>
<tr>
<td>17.10</td>
<td>References</td>
<td>257</td>
</tr>
<tr>
<td>18</td>
<td>DYNAMICS AND CONTROL OF CHEMICAL REACTORS</td>
<td>262</td>
</tr>
<tr>
<td>18.1</td>
<td>Introduction</td>
<td>262</td>
</tr>
<tr>
<td>18.2</td>
<td>The dynamics of an ideal tank reactor</td>
<td>262</td>
</tr>
<tr>
<td>18.2.1</td>
<td>Isothermal reaction</td>
<td>262</td>
</tr>
<tr>
<td>18.2.2</td>
<td>Non-isothermal reaction</td>
<td>266</td>
</tr>
</tbody>
</table>
18.3. The dynamics of an ideal pipe reactor 270
18.3.1. Isothermal reaction 270
18.3.2. Non-isothermal reaction 272
18.4. Non-ideal reactors 275
18.5. The control of chemical reactors 279
18.6. References and bibliographical notes 281

APPENDIX C.
THE EXTENSIONS OF THE EULER-LAGRANGE METHOD OF DYNAMIC OPTIMIZATION TO MULTIDIMENSIONAL CONTROL PROCESSES 283

APPENDIX D.
SOME REMARKS CONCERNING VECTORS AND MATRICES 285
ADDITIONAL NOMENCLATURE PART 2.

B  bottom product molar flowrate (moles/s)
c  heat capacity (J/C m³)
C  number of components
D  top product molar flowrate (moles/s)
f  frequency (rad/s, Hz)
G  molar feedrate (moles/s)
L  molar liquid flowrate (moles/s)
L_r molar liquid flowrate in rectifying section (moles/s)
L_s molar liquid flowrate in stripping section (moles/s)
M  molar liquid accumulation (moles)
n  (subscript) tray number
N  number of trays in column
r  reaction rate (moles/m³ s)
s  Laplace parameter
S  separation factor
V  molar vapour flowrate (moles/s)
w  noise or disturbance
λ  thermal conductivity (J/m s°C)
ξ  system variable
φ  cross sectional area (m²)

ADDITIONAL SYMBOLS PART 2

X(ω) Fourier transform of x(t)
x* sampled form of x
|A| determinant of matrix A
Z{x} z-transform of x*
\hat{x}(z) z-transform of x*
\{f(t)\} number series
"What for do you want new ideas?" asked Mr. Schultz. "Cheaper fuel, cheaper wages, harder work, that is all the new ideas I want."

Evelyn Waugh – The Loved One

12

THE STATE SPACE APPROACH

12.1. introduction

The great advantage that working in the Laplace domain offers, it will be recalled, is that manipulations with transfer functions occur in an algebraic fashion. Any alternative approach in the time domain clearly can expect some tough competition from the "classical" techniques described in previous chapters. Some of the points that make the so-called state space approach a serious contender for the description of dynamic systems are the following:

(i) The extension from s.i.s.o. (Ch. 2, p. 11) to multivariable, m.i.m.o. (multi input – multi output) systems is straightforward,

(ii) Systems having parameters which change with time (time varying systems) can be accommodated,

(iii) controllers, which make use of the powerful mathematics of optimal control theory, can easily be designed for the dynamic process,

(iv) Liapunov methods, for the analysis of the stability of the system, are applicable.

Other very desirable features could be added to the above list.

In essence, the state space description of any dynamic system consists of one or more sets of differential equations. We have already (Ch. 5, p. 59) come across another time domain representation where integral equations – convolution integrals – were used, and the two must not be confused. Nor is the differential operator form of model [1], also in the time domain, meant here. In the following section we’ll specify precisely what we understand by a state space model; so that comparison with the transfer function techniques already described is facilitated we will treat here the s.i.s.o. case, deferring our comments on m.i.m.o. systems until the next chapter.

12.2. the state and output equations

If the dynamic relationship between the input, \( u \), and the output, \( y \), of a system can be expressed in the form

\[ \dot{x} = A(t)x + b(t)u \]  

\[ y = c(t)x, \]  

(12-1)  

(12-2)
then the system is said to be in the (strictly proper) state space form. Equations (12-1) and (12-2) are frequently referred to as the state and output equations respectively. Here, the \((n \times 1)\) column vector \(\mathbf{x}\) contains the \(n\) states (or state variables) of the system; \(A(t)\) is a matrix of time varying parameters, \(a_{ij}(t)\), called the \((n \times n)\) plant or state matrix; \(b\) is a vector of time varying parameters, \(b_i\), called the \((n \times 1)\) input or driving vector and \(c\) is the \((1 \times n)\) output vector. To complete the description of the dynamics a third equation is necessary, wherein the initial conditions (i.e. the values at \(t = 0^+\)) of the states are set, for example:

\[
\mathbf{x}(0^+) = \mathbf{x}_0. \tag{12-3}
\]

A significant simplification occurs whenever the system parameters \((a_{ij}, b, c)\) are time invariant. For the most simple types of problems this can be assumed to be the case. We then have the following continuous, time-invariant, linear (strictly proper) state space form:

\[
\begin{align*}
\dot{\mathbf{x}} &= A\mathbf{x} + b u \\
y &= c\mathbf{x} \\
\mathbf{x}(0^+) &= \mathbf{x}_0. \tag{12-4-5}
\end{align*}
\]

These equations, defining a system \(S_1\) and sometimes written \(S_1(A,b,c)\) for the sake of brevity, will be found time and again in the literature, and in the rest of this book.

It is important not to underestimate the difficulties [2] in transforming our model equations (EPDC p. 30) into the state space form. In section 12.4 we give some suggestions for special cases, although no attempt is made to outline the systematic procedures which do exist [2] to cope with these difficulties. Note also that the state, \(\mathbf{x}\), of the system is not unique, in other words it is often just as possible to base a state space representation upon a number of, say, pressures, as it is to base it upon a number of temperatures. Wiberg [3] discusses some factors affecting the choice of \(\mathbf{x}\).

The order of the system described by eqs (12-1), (12-2) and (12-3) is equal to \(n\). This stems from the fact (Ch. 2, p. 11) that an \(n\)-th order o.d.e. can be written as \(n\) simultaneous first order o.d.e.'s. Equivalently, we could say that we are working in an \(n\) dimensional space, since \(n = \dim \{x\}\).

Example 12.1.

Derive a state space model for a pneumatic control valve, whose transfer function is given [Ch. 3, p. 35] by

\[
\frac{\Delta x(s)}{\Delta P(s)} = \frac{k_2^{-1}}{k_4 s^2 + k_3 s + 1}
\]

The dynamic behaviour in the time domain can be expressed as

\[
k_4 \Delta \dot{x} + k_3 \dot{x} + k_2 \Delta x = \Delta P
\]
Defining \( x_1 = \Delta x \) deviation in stem travel
\[
x_2 = \frac{\partial x}{\partial t} = x_1
\]
\[
u = \Delta P
\]
allows us to write this as
\[
\dot{x}_1 = x_2
\]
\[
\dot{x}_2 = -\frac{k_2}{k_4} x_2 - \frac{k_3}{k_4} x_1 + \frac{1}{k_4} u
\]
\[
y = x_1
\]
which in matrix notation is:
\[
\dot{x} = \begin{bmatrix} 0 & 1 \\ \frac{k_3}{k_4} & \frac{-k_2}{k_4} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{k_4} \end{bmatrix} u
\]
\[
y = \begin{bmatrix} 1 & 0 \end{bmatrix} x
\]

Q 57: Suppose that the system under consideration contains a pure time delay. Can this be accommodated in a state space model? Suggest possible approximate state space models [see Ch. 3, p. 34]. In Chapter 16 a discrete-time version of eqs (12-3) to (12-5) will be presented which could easily handle time delay (delayor) systems.

It is vitally important that we glean as much information possible concerning the general behaviour of the state space model to help in coming to quick prognoses in actual problem situations. Suppose, then, that \( S_1 (A,b,c) \) is at rest (i.e. \( \dot{x}, \dot{u} \) and \( \dot{y} \equiv 0 \)) at some equilibrium value for \( t < 0 \). Following accepted practice, and our previous remarks concerning perturbation variables (Ch. 2, p. 22), this equilibrium is taken as \( x = 0, u = 0 \). Now at \( t = 0 \) a disturbance is assumed to affect the system; in principle this disturbance could come either through \( u \), or through a sudden change in the initial state of the system \( x(0) = x_0 \), or through both. To keep the picture simple yet meaningful we'll take it that at time \( t = 0 \) \( x(0+) = x_0 \neq 0 \) is the only disturbing factor. The one dimensional (first order) case where \( x = [x_1] \) is sketched in Fig. 12.1.

We see that what eq. (12-3) means is that a sudden step-like change occurs in the state at time \( t = 0 \). Notice that an instantaneous change also manifests itself in the velocity of the state \( (\dot{x}_1) \), a fact easily induced from eq. (12-4). Now if the system \( S_1 \) is asymptotically stable or b.i.b.o. stable (Ch. 11, p. 115) – stability being determined by the poles of \( S_1 \), which are in turn solely positioned by the coefficients \( a_{ij} \) of the plant matrix, \( A \) --, then the states will tend to return of their own volition to the equilibrium state \( (x_1 = 0) \). The output variable, \( y \), can be seen to play only a secondary role in the story – it can be thought of as being on a lower hierarchical level than that of the state variables.

To complete the picture we have shown in Fig. 12.1 the effect of an (external to the system \( S_1 \)) induced step change in \( u \). This produces in the state variables a response similar to that caused by the non-zero initial conditions \( x_0 \). However, the ultimate rest state \( (\dot{x} = 0) \) is now non-zero, in accordance with eq. (12-4).
We have deliberately adopted a rather pedantic style to firmly establish some general characteristics of the dynamic behaviour of our state space model. Discussion of the analytical solution of $S_1(A,b,c)$ is deferred until Section 12.6. With these understood, we turn our attention to two variations on the theme of $S_1(A,b,c)$.

Consider firstly the homogeneous form of eq. (12-1), $S_1(A,c)$:

$$\frac{d}{dt}x = A(t)x$$  \hspace{1cm} (12-6)


Together with eq. (12-2) and (12-3). In the time-invariant form

$$\frac{d}{dt}x = Ax$$  \hspace{1cm} (12-7)
we find many applications where batch chemical reactions take place.

**Example 12.2.**

The equations describing a batch reactor wherein two first order consecutive reactions take place have already been derived (Ch. 4, p. 52). Cast these into a state space model.

Defining the three state variables \( x_1, x_2, x_3 \) by the relations

\[
\begin{align*}
    x_1 &= C_A \\
    x_2 &= C_B \\
    x_3 &= C_C
\end{align*}
\]

we transform eqs (4-39) to (4-41) (Ch. 4, pp. 52-3) into:

\[
\begin{bmatrix}
    \dot{x}_1 \\
    \dot{x}_2 \\
    \dot{x}_3
\end{bmatrix} =
\begin{bmatrix}
    -k_1 & 0 & 0 \\
    k_1 & -k_2 & 0 \\
    0 & k_2 & 0
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3
\end{bmatrix}
\]

that is,

\[
\dot{x} = Ax
\]

To complete the state space representation we need

\[
y = cx
\]

where the nature of \( c \) would depend upon which states were available for measurement, and

\[
x(0) = [x_1(0) \quad 0 \quad 0]^T ~ *
\]

Discussion of the solution of eq. (12-6) is left until section 12.5.

**Q 58. What is the matrix state equation of the Hamiltonian system [3]:**

\[
H = \frac{1}{2} q^T V q + \frac{1}{2} p^T T p,
\]

where the equations of motion are

\[
\dot{q}_i = \frac{\partial H}{\partial p_i} \quad \text{and} \quad \dot{p}_i = -\frac{\partial H}{\partial q_i}.
\]

To close this section we study briefly instantaneous (memoryless) systems. Consider, for example, a subsystem \( S_i \) which comprises of a gain element only. This cannot be represented by an \( S_i(A,b,c) \) or \( S_i(A,c) \) model; in fact, a more general form, called the improper **, or memoryless, state space model must be used:

\[
S_i(A,b,c,d) \quad \begin{bmatrix}
    \dot{x} \\
    y
\end{bmatrix} = A(t)x + b(t)u \\
\]

(12-8)

\[
y = c(t)x + d(t)u \quad \dagger
\]

(12-9)

* The character T denotes the transpose.

** The term improper is used [2] since when \( d(t) \neq 0 \) the transfer function \( G(s) \) corresponding to \( S_i(A,b,c,d) \) does not tend to zero as \( s \to \infty \).

\dagger The letter \( d \) is used here and henceforth as the memoryless driving vector.
The control signal, \( u \), can now act directly upon the output \( y \).

### 12.3. disturbances

Sooner or later in process control or dynamics, we encounter the s.i.s.o. system which cannot be adequately described by models of the type \( S(A,b,c,d) \). What is found to be lacking in eqs (12-8) and (12-9) is the possibility of introducing a comprehensive range of disturbances. It would be untrue to say that the state space models given so far are totally unable to cope with disturbances; as we saw in the last section, any disturbance causing an instantaneous, once-and-for-all jump in the state variables (at \( t = 0 \)) can be accommodated. However, not all disturbances are of this type. A possibility would be to allow disturbances to enter into the system through the control variable, \( u \), but then we would no longer be able to talk of \( u \) as the manipulated variable, since part of it would not be within our power to manipulate. Another possibility is to create new states (thus increasing the dimension of the state space) — “dummy” states which simulate the effect of the disturbances.

**Example 12.3.**

Transform the system whose states are described by

\[
\dot{x} = Ax + bu + w
\]

where \( w \) is a constant \( (0 \leq t \leq \infty) \) disturbance vector acting upon the system \( \text{dim} (w) \leq \text{dim} (x) \).

Define a set of dummy state variables, \( z \), according to \( \mathbf{z} = \mathbf{w} \) and note that \( \dot{z} = 0 \). Now augment the original states with the dummy states, giving

\[
\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u
\]

or

\[
\mathbf{z} = A \mathbf{x} + b \mathbf{u},
\]

where \( I \) is the identity or unit matrix, and \( \mathbf{z} \) is called the augmented state vector.

We can expect problems here, however, whenever our disturbance cannot be transformed into a dummy state model \( S_1(A^*) \), eq. (12-6).

To overcome all these difficulties it is sometimes necessary to resort to the use of the following “disturbance-state space” model:

\[
S_1(A,b,c,d,e,f) \quad \begin{cases} \dot{x} = A(t)x + b(t)u + e(t)w \\ y = c(t)x + d(t)u + f(t)w \end{cases} \tag{12-10}
\tag{12-11}
\]

This form of model is ideally suited to our own (i.e. process control and dynamics) needs. Consider the state space description of a heat-exchanger:

**Example 12.4.**

In Fig. 12.2 is sketched a constant temperature bath. We can control either or both of the fluid flows, \( F_w \) and \( F_k \). Neglecting any heat loss to the environment it is possible to write the energy balances on the two fluids as follows (see for example [4]):
Fig. 12.2.

\[
F_W c_p \theta_{wi} - F_W c_p \theta_w - hA(\theta_w - \theta_k) = c_p M_w \frac{d\theta_w}{dt}
\]

\[
F_k c_p \theta_{ki} - F_k c_p \theta_k - hA(\theta_k - \theta_w) = c_p M_k \frac{d\theta_k}{dt}
\]

Casting these equations in perturbation variables and linearising gives:

\[
a_1 \Delta F_w + a_2 \Delta \theta_{wi} + a_3 \Delta \theta_w + a_4 \Delta \theta_k = \frac{d\Delta \theta_w}{dt}
\]

\[
a_5 \Delta F_k + a_6 \Delta \theta_{ki} + a_7 \Delta \theta_k + a_8 \Delta \theta_w = \frac{d\Delta \theta_k}{dt}
\]

where we have assumed that the parameters \(a\) are time invariant. Regarding the two input temperatures as possible disturbances gives the model:

\[
\begin{pmatrix}
\Delta \theta_w \\
\Delta \theta_k
\end{pmatrix} = \begin{pmatrix}
a_3 & a_4 \\
a_8 & a_7
\end{pmatrix} \begin{pmatrix}
\Delta \theta_w \\
\Delta \theta_k
\end{pmatrix} + \begin{pmatrix}
a_1 & 0 \\
0 & a_5
\end{pmatrix} \begin{pmatrix}
\Delta F_w \\
\Delta F_k
\end{pmatrix} + \begin{pmatrix}
a_2 & 0 \\
0 & a_3
\end{pmatrix} \begin{pmatrix}
\Delta \theta_{wi} \\
\Delta \theta_{ki}
\end{pmatrix}
\]

which is in the form:

\[
\dot{x} = Ax + Bu + Ew
\]

\[x(0) = x_0\]

Note that this is a multi-input system.

Frequently we shall opt to let our disturbance variable \(w\) be a stochastic (Ch. 7, p. 70) quantity, having a zero mean value. In Chapter 15 the reason behind this move – the remarkably simple structure of the “optimum” controller – will become apparent. For disturbance with a deterministic character, a useful disturbance model – with which we can generate steps, ramps etc. etc. – is given by.
The use of this type of disturbance generator is described elsewhere [5].

Q. 59: Derive a state space model of a water purification plant consisting of a number of (what must be assumed leaking) reservoirs, connected in series.

12.4. relationship of the state space description with other model descriptions

12.4.1. differential operator form of model

We have already seen in Examples (12-1) and (12-2) how, given a simple system in the form of a differential equation (or in differential operator [1] form) it is only a question of a few simple substitutions before we arrive at the state space model. This is not true when we must handle systems where terms such as \( a_1 \), \( a_2 \) etc. occur. Such systems are frequently encountered — any process model having one or more zeroes, any non-minimum phase system etc. The transformations to be applied in these cases are illustrated below.

Example 12.5.

Derive a state space model of a real 3-term controller.

The dynamics of a commercial P-I-D controller can be described by the differential equation (cf. Ch. 9, p. 96):

\[
\frac{\ddot{c}}{a_1} + \frac{\dot{c}}{a_2} + c = \beta_1 \dot{q} + \beta_2 \ddot{q} + \beta_3 q. \tag{12-14}
\]

where \( q \) is the input signal and \( c \) is the controller output. We will not concern ourselves at present with the initial conditions.

Define the following state variables:

\[
x_1 = c - \gamma_1 q, \quad \gamma \text{ constant throughout};
\]

\[
x_2 = \frac{c}{a_1} - \gamma_1 \dot{q} - \gamma_2 q.
\]

It follows that

\[
x_2 = \dot{x}_1 - \gamma_2 q \tag{12-15}
\]

and on differentiating that

\[
\dot{x}_2 = \frac{c}{a_1} - \gamma_1 \ddot{q} - \gamma_2 \ddot{q}.
\]

Substituting from eq. (12-14) for \( \frac{c}{a_1} \) and by repeated use of the defined state variables in place of \( \frac{c}{a_1} \) and \( c \), we find eventually that

\[
\dot{x}_2 = \frac{a_3}{a_1} x_2 - \frac{a_3}{a_1} x_1 - \frac{a_2}{a_1} \gamma_1 + \left( \frac{\beta_2}{a_1} - \frac{a_2}{a_1} \gamma_1 - \gamma_2 \right) q + \left( \frac{\beta_3}{a_1} - \frac{a_2}{a_1} \gamma_2 - \gamma_2 \gamma_1 \right) q.
\]

On choosing \( \gamma_1 = \frac{\beta_1}{a_1} \) and \( \gamma_2 = \frac{\beta_2}{a_1} - \frac{a_2}{a_1} \beta_1 \) the above equation reduces to

\[
\dot{x}_2 = \frac{a_3}{a_1} x_2 - \frac{a_3}{a_1} x_1 - \frac{1}{a_1^2} \left( \frac{a_2}{a_1} \beta_2 + a_3 \beta_1 \right) q + \frac{a_2}{a_1} \beta_1 \tag{12-16}
\]

Equations (12-15) and (12-16) together form the P-I-D controller state space model.
where $\gamma_3$ is the bracketed term in eq. (12.16).

It is interesting to note that in the above example, a model of the theoretical P-I-D controller would have yielded higher order derivatives in $q$ than in $c$. This is in conflict with our previous (Ch. II, p. 115) statement that for physically meaningful systems, the number of zeroes must be less than the number of poles (following this definition is eq. (12-14) also unrealizable).

The answer to this paradox is that a realizable system must be so for all frequencies; if we restrict our range of interest to frequencies within a certain band, then it may well be possible to realise systems with more zeroes than poles.

12.4.2. transfer function model

It is an easy matter to derive the relationship between the transfer function and the (strictly proper) time invariant state space form of a s.i.s.o system. Laplace transforming eqs. (12-4) and (12-5) we have that

$$L(s) = (sI - A)x = bu + x_0$$

$$y = c^T x$$

Assuming that $x_0 = 0$ (e.g. perturbation variables) we find

$$y = c(sI - A)^{-1}b u$$

so that the transfer function $G(s)$ is given by

$$G(s) = c(sI - A)^{-1}b$$

Q. 60: Derive the relationship between input and output for an improper time invariant s.i.s.o system. Compute this quantity from the state model developed in Example 12-5 for the commercial P-I-D controller and check your answer with that previously stated (Ch. 9, p. 96).

It might be as well for the reader to check at this point that his/her knowledge of elementary matrix operations is up to scratch; nearly all other texts on control at this introductory level contain a section or appendix on matrices.

12.5. canonical forms

It is natural to ask at a certain point whether all $n^2$ elements of the state matrix are really necessary if we are concerned only with specifying the dynamic behaviour of $n$ simultaneous differential equations. Do "simpler" or computationally better conditioned forms of the state space models presented so far exist? It has already been noted that the state space forms given are not unique – any non-singular linear transformation matrix $P$ can be used to
construct a new set of similar equations. For example, let $x = P\tilde{x}$ define a new set of state variables, $\tilde{x}$. If the original system was described by eq. (12-3), eq. (12-8) and eq. (12-9), then substitution for the new state gives:

$$\begin{align*}
\dot{\tilde{x}} &= AP\tilde{x} + bu \\
y &= cP\tilde{x} + du \\
\tilde{x}(0^+) &= P^{-1}x_0
\end{align*}$$

Rearranging (12-21) in state space format:

$$\dot{\tilde{x}} = P^{-1}AP\tilde{x} + P^{-1}bu$$

Eqs. (12-22) to (12-24) are easily seen to be the required proper state space model:

$$\begin{align*}
\dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{b}u \\
y &= \tilde{c}\dot{\tilde{x}} + du \\
\tilde{x}(0^+) &= \tilde{x}_0
\end{align*}$$

It turns out that there are a number of unique canonical forms of the matrices $\tilde{A}$, $\tilde{b}$ and $\tilde{c}$. By canonical is meant that a number of unit vectors are present in the coefficient or state matrix, either as row vectors or column vectors. We shall deal with only two canonical forms here, although many others exist [7]. As we have said, these canonical forms can be most advantageously employed in real engineering applications; since they have fewer non-zero elements than the general case, the computational burden is almost always reduced.

12.5.1. Jordan canonical form

Let us suppose that the state matrix $A$ has no repeated roots. Under this assumption — usually the system is said to have distinct eigenvalues — it can be proved (see [8] for references) that a transformation matrix $P$ exists such that the new canonical system $S_I(\tilde{A}, \tilde{b}, \tilde{c}, \tilde{d})$ has the form of eqs. (12-25) to (12-27) where

$$\tilde{A} = \begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \lambda_n
\end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

and

$$\tilde{c} = [\rho_1 \rho_2 \rho_3 \ldots \rho_n]$$

Here the $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the $n$ distinct eigenvalues, i.e. solutions of the equation.
\[ \begin{vmatrix} \lambda I - A \end{vmatrix} = 0 \] (12-28)

and \( \rho_1, \rho_2, \rho_3, \ldots, \rho_n \) are constants called residues, for which formulas exist [3]. The transformation matrix \( P \), rather ironically, can only be given numerical values once the \( n \) eigenvalues are known, and anyway it is unlikely that an a priori knowledge will be available as to whether or not these eigenvalues are distinct. For multiple roots the structure of \( \tilde{A} \) and \( \tilde{b} \) changes slightly - \( \tilde{A} \) becoming composed of a number of so-called Jordan blocks

\[
\tilde{A}_{ij} = \begin{bmatrix}
\lambda_i & 1 & 0 \\
& \lambda_i & 1 \\
& & \ddots & \ddots \\
& & & \lambda_i & 1 \\
& & & & \lambda_i
\end{bmatrix}
\]

and efficient numerical techniques exist [3], [8] to transform any general system into the Jordan form. Should the system have complex eigenvalues, then the state vector can be augmented (as shown in [3]) to give the "real" Jordan form.

Not only is the Jordan form unique but it also expresses the time behaviour of a s.i.s.o system having distinct eigenvalues in at most \((3n + 1)\) non-zero coefficients. We shall see later that for stability studies or when we come to solve the state space equation this particular form of \( \tilde{A} \) is a naturally attractive one.

Q. 61: Prove that the eigenvalues of the original matrix \( A \) are the same as those of \( \tilde{A} \) [6].

12.5.2. the phase variable canonical form

If the system of interest can be expressed as a rational (Ch. 11, p. 115) transfer function*

\[ G(s) = \frac{\beta_0 s^n + \beta_1 s^{n-1} + \ldots + \beta_{n-1} s + \beta_n}{s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n} \] (12-29)

then the phase variable canonical form can be written down from inspection of eq. (12-29) as \( S_1(A^*, b^*, c^*, d) \), where

\[ \tilde{A} \]

\[ \tilde{b} \]

\[ \tilde{c} \]

\[ \tilde{d} \]

† In accordance with [10] we will denote the determinant of a matrix with bars and that of a scalar by "det" to avoid confusion in the latter case with the absolute value of a scalar.

* Kwakemaak [8], p. 83 gives necessary and sufficient conditions for a transformation to phase variable canonical form to be possible.
The system has been condensed into at most \((3n + 1)\) non-zero coefficients.

**Example 12.6**

In Example 12.5 we derived a state space model of a real 3-term controller. What is the phase variable canonical form of the model?

From eq. (12.14) or (9-11) one has that

\[
G(s) = \frac{a_1 s^3 + a_2 s^2 + a_3}{s^3 + a_2 s^2 + a_3}
\]

The phase variable canonical space model is therefore

\[
\begin{pmatrix}
\overset{z_1}{x_1} \\
\overset{z_2}{x_2} \\
\overset{z_3}{x_3}
\end{pmatrix} =
\begin{pmatrix}
0 & 1 & 0 \\
\frac{-a_2}{a_1} & \frac{a_2}{a_1} & \frac{-a_3}{a_1} \\
\frac{-a_3}{a_1} & \frac{a_3}{a_1} & \frac{-a_2}{a_1}
\end{pmatrix}
\begin{pmatrix}
\overset{\dot{z}_1}{x_1} \\
\overset{\dot{z}_2}{x_2} \\
\overset{\dot{z}_3}{x_3}
\end{pmatrix} +
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix} q
\]

\[
y = \begin{pmatrix}
\frac{-a_2}{a_1} & \frac{a_2}{a_1} & \frac{-a_3}{a_1} \\
\frac{-a_3}{a_1} & \frac{a_3}{a_1} & \frac{-a_2}{a_1}
\end{pmatrix}
\begin{pmatrix}
\overset{\dot{z}_1}{x_1} \\
\overset{\dot{z}_2}{x_2} \\
\overset{\dot{z}_3}{x_3}
\end{pmatrix} + \frac{\beta_1}{a_1} q
\]

The reader is advised to satisfy himself as to the similarity of this model and that previously obtained.

Note that the coefficients of \(A^*\) in the phase variable form can only be real numbers, unlike the corresponding elements of the \(\widehat{A}\) matrix of the Jordan form. The transformation from the phase variable form to the Jordan form and vice versa, i.e.

\[
S_1(A, \widehat{b}, \widehat{c}, d) \leftrightarrow S_1(A^*, b^*, c^*, d)
\]

and

\[
S_1(A^*, b^*, c^*, d) \leftrightarrow S_1(\widehat{A}, \widehat{b}, \widehat{c}, d)
\]

is interesting and often useful. It turns out [6] the \(P\) is the so-called Vandermonde matrix:
12.6. the dynamic response of the linear homogeneous state model

We now explore the dynamic response which is to be expected from various state space models. The aim will be to present as non-mathematical analysis as possible – since numerous texts exist [3,6,8,9] where the contrary is true – and we shall start with the simplest possible case, a system described by the homogeneous time invariant model already discussed, viz.

\[ \mathbf{P} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \ldots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \ldots & \lambda_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \ldots & \lambda_n^{n-1} \end{bmatrix} \quad (12-30) \]

where \( \lambda \) are the \( n \) distinct eigenvalues, and

\[ \mathbf{\tilde{A}} = \mathbf{P}^{-1} \mathbf{A}^* \mathbf{P} \quad (12-31) \]

Q. 62: Prove that the eigenvalues of \( \mathbf{A}^* \) are the same as those of the original state matrix \( \mathbf{A} \).

The question which is posed is the following: given the initial conditions of the state variables, eq. (12-3), what is the dynamic response of the system \( \mathbf{S}_1(A,c) \) or in other words what is the solution (if it exists) to eq. (12-7)? Two methods leading to the answer to this question are next presented.

A reasonable assumption is that the response of \( \mathbf{x} \) will be approximated by a Taylors series vector expansion:

\[ \mathbf{x} = \mathbf{E}_0 + \mathbf{E}_1 t + \mathbf{E}_2 t^2 + \ldots \quad (12-32) \]

where \( \mathbf{E}_k \) \( (k = 0,1,2,\ldots) \) are constant column vectors. At \( t = 0 \), \( \mathbf{x}(0^+) = \mathbf{x}_0 = \mathbf{E}_0 \). By repeatedly differentiating eq. (12-32) we find that

\[ \frac{d^k \mathbf{x}}{dt^k} = \mathbf{E}_k + 2\mathbf{E}_2 t + 3\mathbf{E}_3 t^2 + \ldots \]

\[ \therefore \frac{d^0 \mathbf{x}}{dt^0} = \mathbf{E}_0 = \mathbf{A} \mathbf{x}_0 \quad (12-33) \]

\[ \frac{d^2 \mathbf{x}}{dt^2} = 2\mathbf{E}_2 + 6\mathbf{E}_3 t + \ldots \]

\[ \therefore \frac{d^1 \mathbf{x}}{dt^1} = \mathbf{E}_1 = \mathbf{A} \frac{d^0 \mathbf{x}}{dt^0} = \mathbf{A}^2 \mathbf{x}_0 \quad (12-34) \]

etc. Using these relationships we can write eq. (12-32) as

\[ \mathbf{x} = (\mathbf{I} + \mathbf{A} t + \frac{\mathbf{A}^2}{2!} t^2 + \frac{\mathbf{A}^3}{3!} t^3 + \ldots) \mathbf{x}_0 = e^{\mathbf{A} t} \mathbf{x}_0 \quad (12-35) \]

and so \( y = \mathbf{c} e^{\mathbf{A} t} \mathbf{x}_0 \).

This algebraic vector equation describes the response of \( \mathbf{S}_1(A,c) \). An alternative
method of analysis proceeds as follows.

Laplace transformation of eq. (12-7) gives, in analogy to the scalar case:

$$s\bar{x}(s) - x_0 = A\bar{x}(s)$$  \hspace{1cm} (12-37)

$$\therefore \bar{x}(s) = (sI - A)^{-1}x_0$$  \hspace{1cm} (12-38)

Taking the inverse transform:

$$x = L^{-1}\{(sI - A)^{-1}\}x_0$$  \hspace{1cm} (12-39)

and so

$$y = cL^{-1}\{(sI - A)^{-1}\}x_0.$$  \hspace{1cm} (12-40)

Comparing equations (12-36) and (12-40) it is apparent that the structure of the solution is

$$y = c\Omega(t)x_0,$$

where $$\Omega(t) = e^{At} = L^{-1}\{(sI - A)^{-1}\}x_0.$$  

The quantity $$\Omega(t)$$ is called the matrisant or state transition matrix. Properties of $$\Omega(t)$$ are fully discussed in [3, 6, 10]. We now investigate how the response is calculated in practice.

**Example 12.7.**

Consider the state space model of a batch reactor (Example 12.2). What is its dynamic response?

From Example 12.2 it follows that

$$sI - A = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} -k_1 & 0 & 0 \\ k_1 & -k_2 & 0 \\ 0 & k_2 & 0 \end{bmatrix} = \begin{bmatrix} s + k_1 & 0 & 0 \\ -k_1 & s + k_2 & 0 \\ 0 & -k_2 & s \end{bmatrix}. $$

The inverse matrix, $$(sI - A)^{-1},$$ called the resolvent of $$A [8],$$ is found in the usual way:

$$(sI - A)^{-1} = \frac{\text{adj}(sI - A)}{|sI - A|} = \begin{bmatrix} s(s + k_2) & 0 & 0 \\ sk_1 & s(s + k_1) & 0 \\ k_1k_2 & k_2(s + k_1) & (s + k_1)(s + k_2) \end{bmatrix} = \begin{bmatrix} \frac{1}{s + k_1} & 0 & 0 \\ \frac{k_1}{(s + k_1)(s + k_2)} & \frac{1}{s + k_2} & 0 \\ \frac{k_2}{s(s + k_1)(s + k_2)} & \frac{k_2}{s(s + k_2)} & \frac{1}{s} \end{bmatrix}.$$  

From any table of Laplace transforms we can now find, term for term, the inverse transforms of the resolvent:
Comparing the work which was necessary to find the response with that by conventional techniques (Ch. 4, p. 53), the reader might well be doubtful of the practicality of the newly presented state space approach. The apparent clumsiness is removed, however, when the Jordan canonical form is used.

Q. 63: Put the model of Example 12.2 in Jordan form and rework the reactor response.

12.7. the dynamic response of the linear non-homogeneous state model

The next step is to consider the inclusion of a control signal, $u(t)$ and a disturbance $w$ in the model, i.e., $S_1(A,b,c,d,e,f)$, eqs. (12-3), (12-10) and (12-11). Laplace transforming eq. (12-10):

$$s\vec{x}(s) - \vec{x}_0 = A\vec{x}(s) + b\vec{u}(s) + e\vec{w}(s)$$

$$\therefore \vec{x}(s) = L \{e^{At}\vec{x}_0 + L \{e^{At}(b\vec{u}(s) + e\vec{w}(s))\}. \tag{12-41}$$

By convolution the second term can be transformed back into the time domain. We find that

$$\vec{x} = e^{At}\vec{x}_0 + \int_0^t e^{A(t-\tau)}\{b u + e w\}d\tau \tag{12-42}$$

The response of $S_1(A,b,c,d,e,f)$ is therefore

$$y = ce^{At}\vec{x}_0 + \int_0^t e^{A(t-\tau)}\{b u + e w\}d\tau + du + f w. \tag{12-43}$$

On comparing this equation with that of the homogeneous system (eq. 12-36) the influence of $u$ and $w$ upon the response is easy to see. Perhaps not so obvious is the fact that the scalar, disturbance free version of eq. (12-43) has already been handled (Ch. 5, p. 59).
Example 12.8.

Suppose that a system is not controlled, but nevertheless acted upon by a step disturbance at \( t = 0 \). What is the system response?

Our system becomes (Ch. 2, p. 19):

\[
\dot{x} = Ax + c U(t = 0) \quad (e \text{ is } (n \times 1) \text{ vector})
\]

and for simplicity we assume \( y = c \dot{x} \). From eq. (12.43) we have that

\[
y = c e^{At} x_0 + \int_0^t e^{A(t-\tau)} c U d\tau = c e^{At} x_0 + c e^{At} \int_0^t e^{-A\tau} d\tau \cdot e =
\]

\[= c e^{At} \{ x_0 - A^{-1} (e^{-At} - I)c \}. \]

We recommend the reader to try the following simple calculation.

Q. 64: Show that the response of the system

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{bmatrix} =
\begin{bmatrix}
0 & 1 \\
-2 & -3
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} +
\begin{bmatrix}
0 \\
1
\end{bmatrix} u
\]

to a unit step change in \( u \) at \( t = 0 \) is given by

\[
x_1 = \frac{1}{2} - e^{-t} + \frac{1}{2} e^{-2t}, \\
x_2 = e^{-t} - e^{-2t}.
\]

12.8. eigenvalues, time constants and stability

It is very important to understand the relationships between the eigenvalues of the system (the roots of eq. (12.28)) and its time constants and poles — terms which were frequently encountered in Part 1 of this book. The easiest way to do this is to begin with the transfer function of the system \( S_1 (A, b, c) \), which can be written (eq. (12.20)) as

\[
G(s) = c(sI - A)^{-1} b
\]

or equivalently

\[
G(s) = \frac{c \text{adj}(sI - A)b}{|sI - A|}. \tag{12.44}
\]

Now going back (Ch. 11, p. 115) to the splitting of a rational transfer function into the ratio of two polynomials in \( s \), we see that the denominator of eq. (12.44) occurs in the system characteristic equation, i.e.

\[
|sI - A| = 0,
\]

or in other words that the eigenvalues of the system are the same as the poles \( \dagger \) of the system. It follows also (see Q. 55) that for simple processes described by time constants, the rule of thumb

\[\dagger \text{ For an equivalent state space representation of the zeros of the system, see [12].} \]
"eigenvalue = \frac{1}{\text{time constant}}" \\

should be committed to memory.

Of course, since the words pole and eigenvalue are interchangeable, every-
that has already been said concerning the dependence of the stability of a 
system upon the poles of the system will be equally applicable to the eigen-
values of the system. However, just as during our earlier discussion (Ch. 11, p. 116) we were careful to distinguish between open- and closed-loop systems, so we must be equally cautious when dealing with system eigenvalues. It will 
be recalled, for example, that we proved (Ch. 11, p. 117) that the closed loop 
poles of a s.i.s.o system depend upon the open loop poles and zeros. Frequent-
ly new interpretations can be found for state space representations if one is 
prepared to manipulate the equations a little, as the following example shows.

Example 12.9.
The linear s.i.s.o system
\[
\dot{x} = Ax + bu, \quad x(0) = x_0 \\
y = cx
\]
is controlled via a proportional feedback controller having a setpoint, \( v \). Derive expressions 
for the closed loop transfer function, \( T(s) \) relating the output \( y \) to the setpoint \( v \) \[11\].
The closed loop system is
\[
\dot{x} = Ax + bu \\
y = cx \\
u = f\dot{x} + v, \quad f < 0.
\]
The most obvious approach to find \( T(s) \) is as follows:
\[
\dot{x} = Ax + bf\dot{x} + bv \\
\therefore \dot{x} = (sI - A - bf)^{-1}bv \\
\therefore T(s) = c(sI - A - bf)^{-1}b.
\]
Not so obvious is:
\[
y = c\dot{x} = c(sI - A)^{-1}bu.
\]
Letting \( z = f\dot{x} \) gives
\[
y = c(sI - A)^{-1}b(z + v) \\
z = f\dot{x} = f(sI - A)^{-1}b(z + v) = (1 - f(sI - A)^{-1}b) - f(sI - A)^{-1}bv \\
\therefore y = c(sI - A)^{-1}b \left( \frac{f(sI - A)^{-1}b}{1 - f(sI - A)^{-1}b} + 1 \right) v \\
\therefore T(s) = \frac{c(sI - A)^{-1}b}{1 - f(sI - A)^{-1}b}
\]
Notice how from the first expression for \( T(s) \) it follows that the closed loop poles (eigen-
values) are given by the solution of
\[
|sI - A - bf| = 0.
\]
12.9. references and bibliographical notes

A comparison of state space, transfer function and differential operator models of a process to be controlled can be found in


The difficulties in formulating a system's dynamic behaviour in the state space have led Rosenbrock to propose an alternative representation. See, for example,


The meaning of "state" is discussed in


Our heat-exchanger example in Section 12.3 is treated in detail in


The accommodation of external disturbances in state space models is covered in


[6]: OGATA, K. Modern Control Engineering, Prentice-Hall, 1970

gives several examples of the transformations from one model representation to another.

A recent review of canonical forms for linear multivariable systems is that of


Natural reading to follow on this short introduction to the state space approach would be Chapter 1 of the book by


and/or the mathematically orientated


Likewise the book by


is a good advanced text. Some manipulations of s.i.s.o state space forms are to be found in


A treatment of the state space equivalent representation of zeros is to be found in

We progress in this chapter to the treatment of m.i.m.o. systems with all their problems: non-square matrices, interaction and so forth, which have not up till now been encountered. In doing so, we try to compare two approaches; the state space representation (Section 13.2) and the alternative transfer function formulation (Section 13.3), which has been extensively developed for multivariable systems over the last decade by H. H. Rosenbrock and A. G. J. MacFarlane.

13.1. Introduction

We progress in this chapter to the treatment of m.i.m.o. systems with all their problems: non-square matrices, interaction and so forth, which have not up till now been encountered. In doing so, we try to compare two approaches; the state space representation (Section 13.2) and the alternative transfer function formulation (Section 13.3), which has been extensively developed for multivariable systems over the last decade by H. H. Rosenbrock and A. G. J. MacFarlane.

13.2. Multivariable state space representations

The effect of additional inputs and outputs of a system upon the state space representations given in the preceding chapter are only, of course, notational. That is to say, our system model retains its s.i.s.o. structure, but the input and output coefficient vectors become replaced by matrices. In shorthand form,

\[
\begin{align*}
\text{model} & \quad \text{equations} & \quad \text{s.i.s.o.} & \quad \text{m.i.m.o.} \\
\text{nonhomogeneous} & \quad (12-1) \to (12-3) & \quad S_1(A, b, c) & \quad S_2(A, B, C) \\
\text{homogeneous} & \quad (12-2), (12-3), (12-6) & \quad S_1(A, c) & \quad S_2(A, C) \\
\text{improper} & \quad (12-3), (12-8), (12-9) & \quad S_1(A, b, c, d) & \quad S_2(A, B, C, D) \\
\text{disturbance} & \quad (12-3), (12-10), (12-11) & \quad S_1(A, b, c, d, c, e, f) & \quad S_2(A, B, C, D, E, F) \\
\end{align*}
\]

where, if we assume that \( \dim(y) = r \), \( \dim(x) = m \) and \( \dim(w) = l \), then the above matrices are: \( A(n \times n) \); \( B(n \times r) \); \( C(m \times n) \); \( D(m \times r) \); \( E(n \times l) \) and \( F(m \times l) \).

Just as with s.i.s.o. systems, a number of very useful canonical forms exist; unfortunately in the m.i.m.o. case these forms are not unique. Suppose the model has the form

\[
\begin{align*}
\dot{x} & = Ax + Bu \\
y & = Cx \\
x(0) & = x_0
\end{align*}
\]

or \( S_1(A, B, C) \) for short. Then just as with the s.i.s.o. system, a \( (n \times n) \) transformation matrix \( P \) often exists such that

\[
S_1(A, B, C) \overset{P}{\Rightarrow} S_1(\hat{A}, \hat{B}, \hat{C})
\]
where $\tilde{A} = P^{-1}AP$, $\tilde{B} = P^{-1}B$, and $\tilde{C} = CP$. Six different m.i.m.o. canonical forms are at present known [1], two of which have particularly simple structures – the output identifiable canonical form, where the $B$-matrix is arbitrary while $\tilde{A}$ and $\tilde{C}$ have the forms:

$$\tilde{A} = \begin{bmatrix} 0 & 1 \\ A_1 & A_2 \end{bmatrix} \quad \text{and} \quad \tilde{C} = [I \mid 0]$$

and the input identifiable canonical form where $\tilde{C}$ is arbitrary but

$$\tilde{A} = \begin{bmatrix} 0 & A_1 \\ I & A_2 \end{bmatrix} \quad \text{and} \quad \tilde{B} = [1 \mid 0]$$

Methods for obtaining the appropriate transformation matrices are presented in [1].

Following the development in the previous chapter it is also interesting to extend the concept of a transfer function to the m.i.m.o. system. The term transfer matrix has been coined to relate the multiple output to the multiple input:

$$\tilde{y}(s) = G(s) \tilde{u}(s) \quad \text{(13-4)}$$

where $G$ is an $(m \times r)$ matrix-valued rational function having the form

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \ldots & G_{1r}(s) \\ G_{21}(s) & G_{22}(s) & \ldots & G_{2r}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{m1}(s) & G_{m2}(s) & \ldots & G_{mr}(s) \end{bmatrix}$$

the element $G_{ij}(s)$ being the transfer function relating the $i$-th output to the $j$-th input, provided all other inputs are zero. What is meant here can be seen if eq. (13-4) is written out in full for the $i$-th output:

$$\tilde{y}_i(s) = G_{i1}(s)\tilde{u}_1(s) + G_{i2}(s)\tilde{u}_2(s) + \ldots + G_{ij}(s)\tilde{u}_j(s) + \ldots + G_{ir}(s)\tilde{u}_r(s). \quad \text{(13-5)}$$

In the multivariable system there can, in general, be no question of inverting $G(s)$, since in general $G(s)$ is not square ($m \neq r$). Thus, contrary to the s.i.s.o. case where

$$\tilde{u}(s) = G^{-1}(s)\tilde{y}(s) \quad \text{(13-6)}$$

except in certain trivial cases the inverse transfer matrix does not exist, although each of its elements may well be invertible.

By substituting eqs. (13-1) and (13-2) into eq. (13-4) an expression for the transfer matrix is obtained in state space form:

$$G(s) = C(sI - A)^{-1}B. \quad \text{(13-7)}$$
Compare this equation with eq. (12-20) for the s.i.s.o. system. We shall frequently refer to the transfer matrix in subsequent sections; next, however, we consider what at first sight seems a completely different approach to modelling dynamic systems.

### 13.3. System matrix representation

The fact that difficulties can arise in transforming the model equations, derived from the basic physical and chemical laws, into the state space form has caused some people to look for representations for multivariable systems which alleviate these (and other) problems. One attractive method has been extensively described in the literature [2],[3], and is very suited to the demands of process control work; in what follows an outline of this approach will be given.

Starting point is the equations of conservation of mass, energy and momentum (Chapter 3) or other relations (all of which are assumed to have been linearised about the initial operating point) which together describe the multivariable plant or process of interest. Some of these differential or algebraic equations may be of the following general form:

\[
\begin{align*}
\frac{d^a \xi_1}{dt^a} + a_1 \frac{d^{a-1} \xi_1}{dt^{a-1}} + \cdots + a_{1,a-1} \frac{d \xi_1}{dt} + a_{1,a} \xi_1 + \frac{d^b \xi_2}{dt^b} + a_{2,1} \frac{d^{b-1} \xi_2}{dt^{b-1}} + \\
\hspace{1cm} + \cdots + a_{2,b-1} \frac{d \xi_2}{dt} + a_{2,b} \xi_2 + \cdots + \frac{d^n \xi_N}{dt^n} + a_{N,1} \frac{d^{n-1} \xi_N}{dt^{n-1}} + \\
\hspace{1cm} + \cdots + a_{N,n-1} \frac{d \xi_N}{dt} + a_{N,n} \xi_N \\
= \\
\frac{d^p u_1}{dt^p} + \beta_{1,1} \frac{d^{p-1} u_1}{dt^{p-1}} + \cdots + \beta_{1,p-1} \frac{du_1}{dt} + \beta_{1,p} u_1 + \frac{d^q u_2}{dt^q} + \beta_{2,1} \frac{d^{q-1} u_2}{dt^{q-1}} + \\
\hspace{1cm} + \cdots + \beta_{2,q-1} \frac{du_2}{dt} + \beta_{2,q} u_2 + \cdots + \frac{d^z u_R}{dt^z} + \beta_{R,1} \frac{d^{z-1} u_R}{dt^{z-1}} + \\
\hspace{1cm} + \cdots + \beta_{R,z-1} \frac{du_R}{dt} + \beta_{R,z} u_R.
\end{align*}
\]

Here, the \( \xi_1, \xi_2, \ldots, \xi_N \) are what we shall call system variables, and \( u_1, u_2, \ldots, u_R \) are the R manipulated inputs. For the sake of clarity the possibility of additional disturbance inputs has been neglected.

Other model equations may not have the structure of eq. (13-8). In that case, their form may well be given by:

\[
\begin{align*}
\gamma_1 \frac{d^a \xi_1}{dt^a} + \gamma_{11} \frac{d^{a-1} \xi_1}{dt^{a-1}} + \cdots + \gamma_{1,a-1} \frac{d \xi_1}{dt} + \gamma_{1,a} \xi_1 + \\
\hspace{1cm} + \frac{d^b \xi_2}{dt^b} + \gamma_{21} \frac{d^{b-1} \xi_2}{dt^{b-1}} + \cdots + \gamma_{2,b-1} \frac{d \xi_2}{dt} + \gamma_{2,b} \xi_2 + \\
\hspace{1cm} + \cdots + \gamma_{n-1,n} \frac{d \xi_n}{dt} + \gamma_{n,n} \xi_n \\
= \\
\gamma_{1,1} \frac{d^{p-1} u_1}{dt^{p-1}} + \gamma_{1,p} u_1 + \frac{d^{q-1} u_2}{dt^{q-1}} + \gamma_{2,1} \frac{d^{q-1} u_2}{dt^{q-1}} + \\
\hspace{1cm} + \cdots + \frac{d^z u_R}{dt^z} + \gamma_{R,1} \frac{d^{z-1} u_R}{dt^{z-1}} + \\
\hspace{1cm} + \cdots + \gamma_{R,z-1} \frac{du_R}{dt} + \gamma_{R,z} u_R.
\end{align*}
\]

149
In the above equation, \( y_j \) is just one of the \( y_1, y_2, \ldots, y_m \) process outputs — which may or may not be a measured variable (Ch. I, p. 1). The following detailed example of modelling a simple process should help the reader grasp the idea of using eqs. (13-8) and (13-9) and helps somewhat to justify the approach.

**Example 13.1.**

Consider a continuous industrial plant for producing partially whipped cream, shown in Fig. 3.1. It may not be assumed that the tank contents have a constant density.

![Diagram of a continuous industrial plant for producing partially whipped cream.](image)

In the above equation, \( y_j \) is just one of the \( y_1, y_2, \ldots, y_m \) process outputs — which may or may not be a measured variable (Ch. I, p. 1). The following detailed example of modelling a simple process should help the reader grasp the idea of using eqs. (13-8) and (13-9) and helps somewhat to justify the approach.

**Example 13.1.**

Consider a continuous industrial plant for producing partially whipped cream, shown in Fig. 3.1. It may not be assumed that the tank contents have a constant density.

![Diagram of a continuous industrial plant for producing partially whipped cream.](image)
(i) Overall mass balance.

\[ F_i - F_o = \frac{d}{dt}(\rho V), \]

where \( V \) is the volume of the contents of the tank.

Defining the normal operating point by the symbol “\( r \)” and let

\[ \Delta \rho \equiv \rho - \rho_r, \]
\[ \Delta V \equiv V - V_r, \]
\[ \Delta F_i \equiv F_i - F_{ir}, \]
\[ \Delta F_o \equiv F_o - F_{or}. \]

If the operating point is chosen to coincide with the (steady state) initial conditions, i.e.

\[ \rho_r \equiv \rho(0^-), \]
\[ V_r \equiv V(0^-), \]
\[ F_{ir} \equiv F_i(0^-), \]
\[ F_{or} \equiv F_o(0^-), \]

then the linearised overall mass balance becomes particularly simple, since

\[ \frac{d}{dt}(\rho V) = \frac{d}{dt}(\rho_r + \Delta \rho) (V_r + \Delta V) = V_r \frac{d}{dt}(\Delta \rho) + \rho_r \frac{d}{dt}(\Delta V) \]
\[ \therefore F_i + \Delta F_i - F_{or} - \Delta F_o = \Delta F_i - \Delta F_o = V_r \frac{d}{dt}(\Delta \rho) + \rho_r \frac{d}{dt}(\Delta V) \]

(13-10)

Here we have used the fact that

\[ F_{ir} - F_{or} = F_i(0^-) - F_o(0^-) = \frac{d}{dt}(\rho(0^-) V(0^-)) = 0. \]

(ii) Density change.

It is assumed that the density of the whipped cream in the tank is at any instant of time uniform throughout the tank and proportional to the speed of rotation of the stirrer or the power input through the stirrer blades. In any case, a first approximation would be to set

\[ \Delta \rho = k_1 \Delta \omega \]

(13-11)

(iii) Level control loop.

Supposing that the incoming fluid has a constant density, and that a pneumatic control valve with linear trim and constant pressure drop is used, we see from eq. (8-25) (Ch. 8, p. 91) that

\[ \Delta F_i = k_2 \Delta c, \]

where \( c \) is the pneumatic control signal in the range 3 - 15 psig. If, furthermore, the controller is a two term controller, then the equation describing the level control loop becomes:

\[ \Delta F_i = k_3 \Delta V + k_4 \int_0^1 \Delta V(\tau) d\tau \]
or in differential form:

\[
\frac{d\Delta F}{dt} = k_3 \frac{d\Delta V}{dt} + k_4 \Delta V
\]

Equations (13-10) to (13-12) together with appropriate initial conditions form a simple model for the continuous whipped cream process. To cast the model in the required form, define:

- **input variables:** \( u_1 \equiv \Delta \omega \)
- \( u_2 \equiv \Delta F \)
- **output variable:** \( y_1 \equiv \Delta F \)
- **system variables:** \( \xi_1 \equiv \Delta \rho \)
- \( \xi_2 \equiv \Delta V \).

Then rewrite eqs. (13-10) to (13-12) as:

\[
\begin{align*}
\dot{\xi}_1 &= \gamma_1 \xi_1 + \gamma_2 \xi_2 + u_2 \\
\dot{\xi}_2 &= a_{21} \xi_1 + a_{22} \xi_2 + \frac{du_2}{dt}
\end{align*}
\]

where \( \gamma_1 \equiv -V \); \( \gamma_2 \equiv -\rho \); \( \beta_1 \equiv k_1 \); \( a_{21} \equiv k_4 \) and \( a_{22} \equiv k_3 \).

It is easy to see now that eq. (13-10a) is in the form of eq. (13-9) and eqs. (13-11a), (13-12a) are similar to the generic eq. (13-8).

**Q. 65:** Write a state space representation from the equations (13-10) to (13-12).

The next step is to Laplace transform the system equations of the type eq. (13-8) and/or eq. (13-9). If the warnings concerning the use of perturbation variables have been heeded, then the initial conditions will be zero (Ch. 2, p. 22) and the transformed system equations have the following structure:

\[
\begin{align*}
\tilde{\mathbf{T}}(s)\tilde{\mathbf{x}}(s) &= \mathbf{U}(s)\tilde{\mathbf{u}}(s) \\
\tilde{\mathbf{y}}(s) &= \mathbf{V}(s)\tilde{\mathbf{x}}(s) + \mathbf{W}(s)\tilde{\mathbf{u}}(s)
\end{align*}
\]

Notice that eqs. (13-13) and (13-14) completely describe the whole process, whereas eqs. (13-8) and/or (13-9) are only representative of how two system equations are structured. In other words, we now have the vectors \( \tilde{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{\mathbf{y}} \), where \( \dim \{\tilde{\mathbf{x}}\} = N \); \( \dim \{\tilde{\mathbf{u}}\} = R \); and \( \dim \{\tilde{\mathbf{y}}\} = M \). The matrices \( \tilde{\mathbf{T}}(s), \mathbf{U}(s), \mathbf{V}(s) \) and \( \mathbf{W}(s) \) are **not** similar in nature to the transfer matrix \( \mathbf{G}(s) \) considered in the last section; their elements are each polynomials in \( s \), and with real coefficients \( a, \beta, \gamma \) or \( \varepsilon \). Hence their name: **real polynomial matrices**. Of course, if the dynamical process being described had distributed parameters (Ch. 2, p. 11) or for some other reason contained one or more pure time delays, then \( \tilde{\mathbf{T}}(s), \mathbf{U}(s), \mathbf{V}(s) \) and \( \mathbf{W}(s) \) might well only be approximated by rational polynomials.

Note finally that \( \tilde{\mathbf{T}}(s) \) must be a square \( (N \times N) \) non-singular matrix, otherwise eq. (13-13) is indeterminant. This condition can often be regarded as the
signal to seek additional model equations.

**Example 13.2.**

We continue with the previous example. The transformed system equations may be written:

\[
\begin{align*}
\tilde{y}_1(s) &= \gamma_1 \tilde{x}_1(s) + \gamma_2 \tilde{x}_2(s) + \tilde{u}_2(s) \\
\tilde{x}_1(s) &= \beta_1 \tilde{x}_1(s) \\
\tilde{x}_2(s) &= a_{22} s \tilde{x}_2(s) + a_{21} \tilde{x}_2(s) = s \tilde{u}_3(s)
\end{align*}
\]

or in vector notation:

\[
\begin{bmatrix}
1 & 0 \\
0 & a_{22}s + a_{21}
\end{bmatrix}
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2
\end{bmatrix}
= 
\begin{bmatrix}
\beta_1 & 0 \\
0 & s
\end{bmatrix}
\begin{bmatrix}
\tilde{u}_1 \\
\tilde{u}_2
\end{bmatrix}
\]

\[
\tilde{y}_1 = [\gamma_1 s \quad \gamma_2 s]
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2
\end{bmatrix}
+ [0 \quad 1]
\begin{bmatrix}
\tilde{u}_1 \\
\tilde{u}_2
\end{bmatrix}
\]

The reader is advised to work through the modelling of another dynamical process (for example, this time an open-loop system) to become familiar with this new Laplace domain representation.

It is a fairly simple step now to express eqs. (13-13) and (13-14) in the more concise form:

\[
\begin{bmatrix}
T(s) & U(s) \\
-\mathcal{V}(s) & W(s)
\end{bmatrix}
\begin{bmatrix}
\xi \\
-u
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
-\tilde{V}
\end{bmatrix}
\quad (13-15)
\]

which is analogous to the well-known matrix equation \( Ax = b \). The partitioned matrix above is called the system matrix \([2]\), which we shall denote henceforth by \( P(s) \). Quite a lot of chemical and physical processes, both open and closed loop, can be described by eq. (13-15). Consider, for example, a process which we assume can be described in the open loop situation by eq. (13-15) – let this process be denoted by \( G(T_1, U_1, V_1, W_1) \). When feedback is added to the process – in the form of a second dynamic system \( H(T_2, U_2, V_2, W_2) \) – and when either a non-constant setpoint or disturbance \( \tau \) acts upon the closed loop system (see Fig. 13.2) the following development shows how eq. (13-15) can still be used to describe the closed loop system.

The system equations can be written down in full as:

\[
G(T_1, U_1, V_1, W_1): \quad T_1(s)\xi = U_1(s)\tilde{u} \quad (13-16)
\]

\[
\tilde{V} = V_1(s)\tilde{\xi} + W_1(s)\tilde{u} \quad (13-17)
\]

\[
H(T_2, U_2, V_2, W_3): \quad T_2(s)\lambda = U_2(s)\tilde{V} \quad (13-18)
\]

\[
\tilde{C} = V_2(s)\tilde{\lambda} + W_2(s)\tilde{V} \quad (13-19)
\]

Setpoint/disturbance: \( \tilde{u} = \tau - \tilde{c} \quad (13-20) \)
Fig. 13.2.

After some manipulation, we find from eqs. (13-16), (13-17), (13-18) and (13-20):

\[
\begin{align*}
T_{1} \bar{x} - U_{1} \bar{v} + U_{1} \bar{w} &= 0 \\
T_{2} \bar{\lambda} &= U_{2} \bar{y} \\
-V_{1} \bar{\xi} - W_{1} \bar{v} + W_{1} \bar{c} &= -\bar{y},
\end{align*}
\]

whereupon by using eq. (13-19) to eliminate \( \bar{c} \):

\[
\begin{align*}
T_{1} \bar{\xi} - U_{1} \bar{r} + U_{1} V_{2} \bar{\lambda} &= -U_{1} W_{2} \bar{y} \\
T_{2} \bar{\lambda} &= U_{2} \bar{y} \\
-V_{1} \bar{\xi} - W_{1} \bar{r} + W_{1} V_{2} \bar{\lambda} &= -(1 + W_{1} W_{2}) \bar{y}.
\end{align*}
\]

In partitioned matrix form these equations look like

\[
\begin{bmatrix}
T_{1} & U_{1} V_{2} & U_{1} \\
0 & T_{2} & 0 \\
-V_{1} & W_{1} V_{2} & W_{1}
\end{bmatrix}
\begin{bmatrix}
\bar{x} \\
\bar{\lambda} \\
\bar{r}
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\]

or, by defining an augmented system variable vector \( \bar{\phi} = [\bar{x}, \bar{\lambda}^T]^T \), which does nothing more than dividing up the above partitioned form along the given dotted lines, we arrive at

\[
\begin{bmatrix}
\bar{T} & \bar{U} \\
-\bar{V} & \bar{W}
\end{bmatrix}
\begin{bmatrix}
\bar{\phi} \\
-\bar{r}
\end{bmatrix}
= \begin{bmatrix}
I & 0 & U_{1} W_{2} \\
0 & 0 & -U_{2} & 0
\end{bmatrix}
\begin{bmatrix}
0 \\
-\bar{y}
\end{bmatrix},
\]

where

\[
\begin{align*}
\bar{T} &= \begin{bmatrix}
T_{1} & U_{1} V_{2} \\
0 & T_{2}
\end{bmatrix}; \quad \bar{U} = \begin{bmatrix} U_{1} \end{bmatrix} \\
-\bar{V} &= \begin{bmatrix} -V_{1} & W_{1} V_{2} \end{bmatrix}; \quad \bar{W} = \begin{bmatrix} W_{1} \end{bmatrix}
\end{align*}
\]

\textbf{154}
Now using the fact [4] that the inverse of a partitioned matrix of the form
\[
\begin{bmatrix}
1 & \tilde{Q} \\
0 & \tilde{P}
\end{bmatrix}^{-1} =
\begin{bmatrix}
1 & -\tilde{Q}\tilde{P}^{-1} \\
0 & \tilde{P}^{-1}
\end{bmatrix}
\]
we see that so long as the matrix \([I + W_1 W_2]\) is nonsingular the closed-loop system matrix may be written as
\[
\begin{bmatrix}
\tilde{T} + \tilde{Q}\tilde{P}^{-1} \tilde{V} \\
\tilde{P}^{-1} \tilde{V}
\end{bmatrix}
\begin{bmatrix}
\tilde{P}^{-1} \\
-\tilde{V}
\end{bmatrix}
= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]
(13-21)
which has exactly the same form as eq. (13-15).

It is also interesting to compare the system matrix with both the multi-variable state space and transfer matrix representations. The latter is easily obtained from eqs. (13-13) and (13-14).

Remembering that \(T(s)\) is a non-singular matrix, the system variables, \(\xi\), can be eliminated from the transformed system equations, leaving
\[
\tilde{Y}(s) = [V(s)T^{-1}(s)U(s) + W(s)] \overline{U}(s)
\]
(13-22)
Comparing this with eq. (13-4) it follows that
\[
G(s) = V(s)T^{-1}(s)U(s) + W(s).
\]
(13-23)
In general, the relationship between \(G(s)\) and \(P(s)\), the system matrix, becomes rather messy [3]. However, for the special case of a dynamical system having as many outputs as inputs \((M = R)\), \(P(s)\) is then a partitioned square matrix, whose determinant can be broken down [2] into:
\[
|P(s)| = |T(s)||V(s)T^{-1}(s)U(s) + W(s)| = |T(s)||G(s)|
\]
(13-24)
Although other possibilities exist, the state space representation may be developed via the transfer matrix, \(G(s)\). Recalling eq. (13-7) and comparing with eq. (13-23) suggests that
\[
T(s) = (sI - A); \quad U(s) = B; \quad V(s) = C
\]
and hence that
\[
P(s) = \begin{bmatrix}
sI - A & B \\
-C & 0
\end{bmatrix}
\]
(13-25)
which has been termed the state-space system matrix [3].

13.4. closed-loop systems

In the last section little attention was paid to the role of disturbances affecting the process. It will be recalled, however, that a disturbance vector
w can easily be incorporated into the general, linear state space model, eqs. (12-3), (12-10) and (12-11) giving:

\[
\dot{x} = Ax + Bu + Ew \\
y = Cx + Du + Fw \\
x(0) = x_0
\]  

or \(S, (A,B,C,D,E,F)\) for short. An attractive pictorial representation of eqs. (13-26) and (13-27) is shown in Fig. 13.3a. The triangular symbol represents integration (see Chapter 14) and double lines depict the various vectorial information streams. Likewise, the squares are matrices (which are not, however, themselves necessarily square). Notice how \(Ax\) can be seen as internal feedback.

\[\text{Fig. 13.3a. The process } S, (A,B,C,D,E,F).\]

Simple manipulation of the above state space equations shows that an equivalent transfer matrix representation exists. Assuming that \(x_0 = 0\), we have from eq. (13-26):

\[
\bar{x} = (sI - A)^{-1} Bu + (sI - A)^{-1} Ew.
\]

Hence

\[
\bar{y} = \{C(sI - A)^{-1} B + D\} \bar{u} + \{C(sI - A)^{-1} E + F\} \bar{w} = G(s)\bar{u} + G_d(s)\bar{w} = (13-29)
\]

The two transfer matrices together define \(S,\); see Fig. 13.3b.
Q. 66: Derive a pictorial representation of a process described in terms of system variables, and acted upon not only by manipulated inputs but also disturbances.

Let us now return to the very beginning of this story (Ch. 1, p. 1); there the distinction was made between measured output variables, \( y \), and unmeasured output variables, \( v \). We have really never paid very much attention to this point up to now – the variables \( y \in \bar{y} \) being often referred to vaguely as “outputs”. In fact, not all the unmeasured output variables need be considered. It is sufficient, as a few moments reflection will prove, to account for only two types of outputs: the measured outputs, \( y \), and the controlled variables, \( z \), – being those variables which we wish to control. Conceptually we have the Venn diagram of Fig. 13.4, where the total space is \( \chi (\chi = y \cup v) \). In vector notation, the system \( S_1 (A,B,C,D,E,F) \) must receive the supplementary equation:

\[
\mathbf{z} = \mathbf{Lx}
\]  \hspace{1cm} (13-29)

where \( \mathbf{L} \) is a \((k \times n)\) dimension unitary matrix. In transfer matrix notation, from eqs. (13-26) and (13-29)

\[
\mathbf{z} = \mathbf{L}(\mathbf{sI} - \mathbf{A})^{-1} \mathbf{B} \mathbf{u} + \mathbf{L}(\mathbf{sI} - \mathbf{A})^{-1} \mathbf{E} \mathbf{w}
\]  \hspace{1cm} (13-30)

\[
= \mathbf{M}(\mathbf{s}) \mathbf{u} + \mathbf{M}_d(\mathbf{s}) \mathbf{w}.
\]  \hspace{1cm} (13-31)

Fig. 13.4. Conceptual model of unmeasured \( v \), measured \( y \) and controlled \( z \) variables.
By taking account of the measurement transmitters, the controllers and the final control elements it is now possible to “close the loop” for the general, multivariable system in a fashion analogous to the s.i.s.o. case (Ch. 8, p. 84). Denoting the transmitters’, controllers’ and control elements’ transfer matrices by $H(s)$, $K(s)$ and $J(s)$ respectively, the general closed loop system can be drawn in block diagram form as shown in Fig. 13.5.

Fig. 13.5.

Q. 67: In view of what was already said in Chapters 8 and 9 of Part 1, what structure would the state space model of this general closed loop system be likely to have? Draw a flow diagram similar to Fig. 13.3a.

Fig. 13.5 is by no means unique; alternative forms can often be beneficial in specific applications.

Q. 68: Rearrange Fig. 13.5 so as to yield a system with an input compensator, $T(s)$, where only the setpoint is used as input to the compensator.

Suppose now that the variables to be controlled are the same as the
Equations (13-34) and (13-36) are two alternative forms of \( n(s) \). Note that we have assumed that \((I + GJKH)\) is invertible — in fact it can be shown that if this is so (see (5) or eq. (13-56)) then \((I + HGJK)\) is also invertible.

Similarly, if no setpoint changes occur, then:

\[
\bar{y} = GJK\bar{y}(s) = GJK(I + HGJK)^{-1}\bar{y}(s)
\]

\[
\therefore r\Omega_y(s) = GJK(I + HGJK)^{-1} GJK
\]  

Alternately, however, if we begin with

\[
\bar{y} = \bar{r}(s) - HGJK\bar{e}(s)
\]

then

\[
\bar{y} = GJK\bar{e}(s) = GJK(I + HGJK)^{-1}\bar{r}(s)
\]

\[
\therefore r\Omega_y(s) = GJK(I + HGJK)^{-1}
\]  

Equations (13-34) and (13-36) are two alternative forms of \( r\Omega_y(s) \). Note that we have assumed that \((I + GJKH)\) is invertible — in fact it can be shown that if this is so (see [5] or eq. (13-56)) then \((I + HGJK)\) is also invertible.

Similarly, if no setpoint changes occur, then:

\[
\bar{y} = -GJKH\bar{y}(s) + G_d\bar{w}(s)
\]

\[
\therefore w\Omega_y(s) = (I + GJKH)^{-1}G_d
\]  

Q. 69: Write an alternative form for the closed loop transfer matrix, \( w\Omega_y(s) \).

Looking at eqs. (13-34) and (13-37) we see that \((I + GJKH)\) is an important quantity in the case of either setpoint or disturbance changes. In fact, it occurs so often in more advanced studies that it has acquired its own name — the return difference matrix. To understand the name, consider Figs. 13.6a and 13.6b. Here we have condensed the forward transmittance \( GJK \) into one transfer matrix \( G(s) \). When feedback is present탑,

\[
\bar{y} = G\bar{r}(s) - GH\bar{y}(s),
\]  

while without feedback:

\[
\bar{y} = G\bar{r}(s) - GH\bar{u}(s)
\]

\[
\therefore \bar{y} - \bar{y} = GH(\bar{u} - \bar{y})
\]

탑 This interpretation of the return difference matrix differs from others ([3],[5]) in not requiring a zero setpoint vector, \( \bar{r} \equiv 0 \).
It is seen that the return difference thus expresses the effect of feedback upon the difference between the input and the output signals of an open loop system. The matrix $G(s)H(s)$ is called the \emph{return ratio} (sometimes loop gain) matrix.

13.5. poles and zeros of multivariable systems

The concepts of poles and zeros of s.i.s.o. systems cannot be straightforwardly carried over to multivariable systems. This should be obvious since in dealing with a transfer matrix we are handling a real rational function matrix. For the easiest case of square transfer matrices (dim $(u) = \text{dim } (y)$) it turns out (for proof, see p. 40 of [5]) that for an open loop system having a transfer matrix $G(s)$:

$$|G(s)| = \frac{\psi(s)}{\varphi(s)},$$

i.e.

$$|G(s)| = \frac{k(s - z_1)(s - z_2) \ldots (s - z_q)}{(s - p_1)(s - p_2) \ldots (s - p_n)}$$

A number of remarks are in order. Firstly, we notice on comparing eqs. (13-42) and eq. (11-2) that the determinant of the m.i.m.o. system must be taken. Secondly, the polynomial $\psi(s)$, the roots of which are the \emph{zeros} of the matrix $G(s)H(s)$.
m.i.m.o. system, can be shown to be of degree \( n - m \) or less. Furthermore, the polynomial, \( \varphi(s) \), is the open loop characteristic polynomial of the system and is given by

\[
\varphi(s) = |sI - A| \tag{13-43}
\]

for the system \( S_1(A, B, C) \).

Q. 70: Using the fact that the determinant of a scalar is the same as the scalar, prove eq. (13-43).

For a more detailed survey of poles and zeros in m.i.m.o. systems with non-square transfer matrices and cancellation of poles or zeros see [6], while a useful technique for ascertaining the stability of multivariable systems is given in [7]; we consider further only the similarities and differences between square open and closed loop systems. Turning, therefore, to closed loop systems, it would be reasonable to assume, it seems, that the closed loop characteristic polynomial of a system such as shown in Fig. 13.6a would be the denominator of the determinant of the closed loop transfer matrix. That is, with exactly the same reasoning as in eq. (13-41):

\[
\left| \Omega_c(s) \right| = \frac{\psi_e(s)}{\varphi_c(s)} \tag{13-44}
\]

where we are interested primarily in the closed loop characteristic polynomial, \( \varphi_c(s) \). This rational function must be the same wherever we choose to take our closed loop transfer matrix; the stability properties of a closed loop system are, after all, invariant of where we look in the loop. In other words,

\[
\left| \Omega_c(s) \right| = \frac{\psi_e(s)}{\varphi_c(s)} \tag{13-45}
\]

and

\[
\left| \Omega_f(s) \right| = \frac{\psi_f(s)}{\varphi_c(s)} \tag{13-46}
\]

From Fig. 13.6a it follows that

\[
\Omega_y(s) = [1 + G(s)H(s)]^{-1} G(s) \quad \text{(cf. eq. (13-34))} \tag{13-47}
\]

\[
= G(s)[1 + H(s)G(s)]^{-1} \quad \text{(cf. eq. (13-36))} \tag{13-48}
\]

\[
\Omega_e(s) = [1 + H(s)G(s)]^{-1} \tag{13-49}
\]

\[
\Omega_f(s) = H(s)G(s)[1 + H(s)G(s)]^{-1} \tag{13-50}
\]

Before taking determinants of the above equations we note the following two elementary properties of determinants of square matrices. Firstly, if \( A \) and \( B \) are two square rational function matrices, then

\[
|AB| = |A||B| \tag{13-51}
\]
and less obviously
\[ |A^{-1}| = (|A|)^{-1}. \]  
(13-52)

This latter result can be proved as follows:
\[ |A^{-1}A| = |I| = 1 = |A^{-1}| |A|, \]
from (13-51).

Hence \( \frac{1}{|A|} = |A^{-1}| \), i.e. eq. (13-52).

With the help of these two properties we have then that
\[
\begin{align*}
|\Omega_y(s)| &= \frac{|G(s)|}{|I + G(s)H(s)|} = \frac{|G(s)|}{|I + H(s)G(s)|} \quad (13-53) \\
|\Omega_e(s)| &= \frac{1}{|I + H(s)G(s)|} \quad (13-54) \\
|\Omega_f(s)| &= \frac{|H(s)||G(s)|}{|I + H(s)G(s)|} \quad (13-55)
\end{align*}
\]

Notice how the return difference matrix appears already. From eq. (13-53) it is clear to see that
\[ |I + G(s)H(s)| = |I + H(s)G(s)| = \frac{\lambda(s)}{\Delta(s)}, \text{ say.} \]
\[ (13-56) \]

Furthermore, by writing the determinants of \( G(s) \) and \( H(s) \) according to eq. (13-41) as
\[
\begin{align*}
|G(s)| &= \frac{\psi_G(s)}{\varphi_G(s)} \\
|H(s)| &= \frac{\psi_H(s)}{\varphi_H(s)}
\end{align*}
\]
and substituting all these polynomials into eqs. (13.53) to (13-55) we find that
\[
\begin{align*}
|\Omega_y(s)| &= \frac{\psi_G(s)\Delta(s)}{\varphi_G(s)\lambda(s)} \quad (13-57) \\
|\Omega_e(s)| &= \frac{\Delta(s)}{\lambda(s)} \quad (13-58) \\
|\Omega_f(s)| &= \frac{\psi_G(s)\psi_H(s)\Delta(s)}{\varphi_G(s)\varphi_H(s)\lambda(s)} \quad (13-59)
\end{align*}
\]

For eqs. (13-44) to (13-46) to be compatible with the above equations requires that
\[ \Delta(s) = \varphi_G(s) \varphi_H(s), \quad (13-60) \]

i.e. the denominator of the determinant of the return difference matrix is the product of the open loop characteristic polynomials of \( G(s) \) and \( H(s) \). Furthermore

\[ \lambda(s) = \varphi_c(s), \quad (13-61) \]

that is, the numerator of the determinant of the return difference matrix is the **closed-loop characteristic polynomial of the system**. It follows from eqs. (13-56) and (13-60), (13-61) that

\[ \varphi_c(s) = \varphi_G(s) \varphi_H(s) [I + G(s)H(s)]. \quad (13-62) \]

To summarize the above development, the stability of the closed loop system is dependent upon, of course, the closed loop poles, which are found as the roots of the numerator polynomial of the determinant of the return difference matrix.

**Q. 71:** Check that eq. (13-61) predicts the correct closed loop characteristic polynomial for the single input single output case (eq. (11-4)).

Unlike the poles of the closed loop multivariable system, the **zeros** of the closed loop system depend upon which transfer matrix is chosen. For example, eqs. (13-57), (13-60) and (13-4) give

\[ \psi_y(s) = \psi_G(s) \varphi_H(s), \quad (13-63) \]

while

\[ \psi_e(s) = \varphi_G(s) \varphi_H(s) \quad (13-64) \]

and

\[ \psi_f(s) = \psi_G(s) \psi_H(s) \quad (13-65) \]

can be found by similar reasoning. It is therefore impossible to talk about the zeros of a closed loop system without mentioning at which place in the loop we are looking.

### 13.6. interaction and pole placement

The problem of interaction has already been introduced (Ch. 10, p. 106). What we will do here is to recast the problem in matrix form, decouple (in the most elementary way possible) the m.i.m.o. system into a number of non-interacting s.i.s.o. control loops, and then show how a controller could be designed, using a pole assignment technique, for each of the individual, non-interacting, loops. Our aim, it must be stressed, remains consistent with the title

\[ \dagger \text{ Which, to the best of our knowledge, is new. For a proof that eq. (13-61) still holds if } m \neq r, \text{ see [5].} \]
of this book; we seek, that is, to introduce and explain fundamental concepts, rather than give a cookery book of practically applicable design recipes.

Consider the system shown in Fig. 13.5 and assume that the effect of the disturbances can be neglected.* Furthermore, suppose that \( \text{dim}(\mathbf{x}) = \text{dim}(\mathbf{u}) = \text{dim}(\mathbf{r}) = \text{dim}(\mathbf{e}) \), and that the measured variables \( \mathbf{y} \) are also the variables to be controlled, \( \mathbf{z} \). For complete decoupling of the \( r \) setpoints from the \( r \) outputs — which implies that any change in one setpoint will affect only one output — the closed loop transfer matrix \( (r \times r) \) must be diagonal, i.e.

\[
\begin{bmatrix}
\mathbf{1}_{\Omega_1(s)} & 0 \\
0 & \mathbf{1}_{\Omega_2(s)} \\
\mathbf{0} & \mathbf{1}_{\Omega_r(s)}
\end{bmatrix}
\]

(13-66)

where

\[
\begin{align*}
\mathbf{r}_{\Omega}(s) &= [I + GJKH]^{-1}GJK \\
&= GJK[I + HGJK]^{-1}
\end{align*}
\]

(13-34)

(13-36)

From the above equations we find, after some simple manipulations, that

\[
GJK = (I - \Omega H)^{-1} \Omega
\]

(13-67)

(13-68)

It follows that, provided the feedback matrix \( H(s) \) is diagonal, the forward transmittance, \( G(s)J(s)K(s) \), must also be a diagonal rational function matrix. The reader is advised to convince himself of the truth of this statement before proceeding further. In terms of the decoupling controller,

\[
K(s) = [G(s)J(s) - \Omega(s)H(s)G(s)J(s)]^{-1} \Omega(s)
\]

(13-69)

from eq. (13-67), and assuming that \( |GJ| \neq 0 \), that is that \( G(s)J(s) \) is square and invertible.

---

* The assumption often leads to a control system of little practical applicability. More advanced techniques would allow for both variations in \( w \) and \( r \), the setpoints.
Example 13.3.

Here we rework the example 10.1 (Ch. 10, p. 107) using the multivariable techniques developed above. See Fig. 13.7.

The LP users' pressure, $P$, was related to the difference in high and low pressure steam flows. As an approximation for small variations around the normal operating point:

$$\Delta P = K_7 (\Delta F_{H1} - \Delta F_L)$$  \hspace{1cm} (13-70)

Also

$$\Delta \omega = K_5 \Delta F_{H1} + K_6 \Delta F_L,$$  \hspace{1cm} (13-71)

hence

$$G(s) = \begin{bmatrix} K_5 & K_6 \\ K_7 & -K_7 \end{bmatrix}$$

represents the process transfer matrix, defined according to: $\bar{y}(s) = G(s)u(s)$, with $\bar{y} = (\Delta \omega \quad \Delta P)^T$ and $u = (\Delta F_{H1} \quad \Delta F_L)^T$. From eqs. (10-2), (10-3) and (13-70) the precise transfer matrix for the control elements – the high and low pressure valves – can be computed. To keep the picture clear, we will neglect the effect of pressure on valve flow, in which case we obtain the much simpler transfer matrix:

$$J(s) = \begin{bmatrix} K_1 & 0 \\ 0 & K_3 \end{bmatrix},$$

where $\bar{u}(s) = J(s)\bar{y}(s)$ and $\bar{x}$ is $(\Delta x_{H1} \quad \Delta x_L)^T$. To decouple the system the simplest choice of closed loop transfer matrix is

$$\Omega_y(s) = \begin{bmatrix} a & 0 \\ 0 & \beta \end{bmatrix}.$$

Then, if $H(s)$ is taken as the identity matrix, eq. (13-69) gives

$$K(s) = \begin{bmatrix} (1 - a)K_1K_5 & (1 - a)K_5K_6 \\ (1 - \beta)K_1K_7 & -(1 - \beta)K_5K_7 \end{bmatrix}^{-1} \begin{bmatrix} a & 0 \\ 0 & \beta \end{bmatrix}.$$

The inverse is easily obtained as

$$\frac{1}{\Delta} \begin{bmatrix} (1 - \beta)K_5K_7 & -(1 - a)K_5K_6 \\ -(1 - \beta)K_1K_7 & (1 - a)K_1K_5 \end{bmatrix}$$

where $\Delta = -(1 - a)/(1 - \beta)K_5 + K_6/K_1K_3K_7$

and hence the controller gains are given by

$$K(s) = \frac{1}{K_5 + K_6} \begin{bmatrix} K_1(1 - a) & K_6\beta \\ a & K_5(1 - \beta) \\ a & K_5\beta + a \\ K_1(1 - a) & K_3K_7(1 - \beta) \end{bmatrix}.$$  \hspace{1cm} (13-72)

Comparing eq. (13-72) with eqs. (10-4) and (10-7) we see that there is agreement.

Looking closely at the above example it is obvious that it is possible not
only to decouple the system, but also to arrange for the controlled system to follow a prescribed pattern of behaviour. To achieve this, we would simply replace the constants $\alpha$ and $\beta$ by appropriately chosen transfer functions.

In terms of a design procedure for control systems, one approach [3] is, roughly speaking, based upon the above philosophy. By splitting the controller transfer matrix $K(s)$ into the product $K_1(s)K_2(s)$, where $K_2(s)$ is a diagonal matrix, it is in fact possible to choose $K_1(s)$ so that the system is decoupled, and the diagonal elements of $K_2(s)$ may be freely chosen in accordance with s.i.s.o techniques to control each of the non-interacting loops.

Another technique, suited particularly to the state space representation, is that of pole assignment or pole placement [5], [8], a brief outline of which follows. Suppose we consider the single input system $S_1(A,b)$. Under certain conditions (p. 83, [5]) it is possible, by means of the transformation $\tilde{X} = P^{-1}X$, to write down $S_1(A^*,b^*)$ the phase variable canonical form (Chapter 12, section 12.5.2), which it will be recalled was:

$$S_1(A^*,b^*): \frac{2}{\tilde{X}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \tilde{X} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

$$\tilde{X}(0) = P^{-1}x(0)$$

Here, the alphas are the coefficients of the characteristic equation of $S_1(A,b)$:

$$[sI - A] = s^n + a_1s^{n-1} + \cdots + a_{n-1}s + a_n.$$ (13-75)

Suppose now that $S_1(A,b)$ was controlled by applying a scalar feedback law of the form:

$$u = -K^T\tilde{X} + r.$$ (13-76)

Here, $r$ is the setpoint, and $K$ is a column vector $(n \times 1)$ having the elements $K_i$, $i = 1,2,\ldots,n$. The closed loop system is seen to be $S_2(A^* - b^*K^T,b^*)$, where a little thought shows that the new state matrix of $S_2$ is given by:

$$A^* - b^*K^T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n - K_1 & -a_{n-1} - K_2 & -a_{n-2} - K_3 & \cdots & -a_1 - K_n \end{bmatrix}$$

(13-77)

This has again the phase variable structure and it is not difficult to predict therefore that the closed loop system $S_2$ will have the characteristic equation:

$$[sI - A^* - b^*K^T] = s^n + (a_1 - K_n)s^{n-1} + \cdots +
+(a_{n-1} - K_2)s + a_n - K_1$$

(13-78)
It follows that the coefficients of eq. (13-78) may be chosen to have any real values, which in turn implies that the poles of $S_2$ can be arbitrarily assigned to any location in the complex plane. Complex poles would have, of course, to appear in pairs.

It can readily be proved [5] that if the single input system $S_1(A,b)$ can be written in the phase variable form $S_2(A^*,b^*)$, then it is always possible to construct a proportional feedback controller having the structure of eq. (13-76), which can place all the poles of the closed loop system $S_2(A^* - b^*K^T,b^*)$ at arbitrary locations.

Q. 72: Show that, given $S_1(A,b)$, where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & -3 & 0 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and where it is desired that the poles of the controlled system shall lie at $(0,-1$ and $-2)$ that the feedback controller to do this [8] has the gains $[-2, -1, -3]$.

Q. 73: Using the results of Example 13.3, assume a second order dynamic response for $\Delta w_{1}$ and $\Delta P$. Decouple the system and assign the closed loop poles to $(-2, -3i, -2 + 3i)$ and $(-6, -5)$ respectively.

### 13.7. Reconstructing the State

Often it is easier to develop control laws or algorithms in terms of state feedback; that is, it is assumed that all the states of the system are available to be measured and fed back. In practice this is rarely the case, and we have to try to reconstruct the state from the information available in the measured variables. In this section we will introduce several ways of doing this, beginning by noting what it is possible to reconstruct and what not. Throughout it is assumed that there are no stochastic measurement or process disturbances acting upon the system.

Suppose that we have the homogeneous system $S_1(A,c)$ which although it has only one measured output, $y$, (to keep the story simple) has $n$ states, $x$, and may be either a closed or open loop system. If a transformation of the type $\tilde{x} = Px$ can be found which allows $S_1(A,c)$ to be written as $S_1(\tilde{A},\tilde{c})$, where

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & 0 \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \quad \text{and} \quad \tilde{c} = [\tilde{c}_1 \mid 0]$$

are in partitioned form, then $S_1(\tilde{A},\tilde{c})$ is said to be in the reconstructibility canonical form [5], which, on writing as two subsystems

$$S_2(\tilde{A}_{11},\tilde{c}_1): \quad \begin{bmatrix} \tilde{x}_1 \\ \tilde{c}_1 \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ c_1 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Where the vector $\tilde{x}_1$ represents a reconstruction of the state $x_1$ and the matrices $\tilde{A}_{ij}$ of the observables may be shown [13] to be

$$\tilde{A}_{11} = \begin{bmatrix} \lambda_1 \end{bmatrix}, \quad \tilde{A}_{12} = \begin{bmatrix} \lambda_2 \end{bmatrix}, \quad \tilde{A}_{21} = \begin{bmatrix} \lambda_3 \end{bmatrix}, \quad \tilde{A}_{22} = \begin{bmatrix} \lambda_4 \end{bmatrix}$$

$$\tilde{c}_1 = \begin{bmatrix} \gamma_1 \end{bmatrix}, \quad \tilde{c}_2 = \begin{bmatrix} \gamma_2 \end{bmatrix}$$

where $\lambda_i$ and $\gamma_i$ are the eigenvalues and outputs of $S_1(A,c)$.
and

\[ S_3: \dot{\bar{x}}_2 = \bar{A}_{21}\bar{x}_1 + \bar{A}_{22}\bar{x}_2 \]

shows that only those states contained in \( \bar{x}_1 \) can ever be reconstructed from the measured output, \( y \). See Fig. 13.8. If all the states can be reconstructed from all the measured outputs, \( y \), then the system is said to be \textit{completely reconstructible}.

One of the most obvious ways of reconstructing the reconstructible states is to differentiate the measured output \((n - 1)\) times. For \( S_1(A,c) \) we would get

\begin{align*}
\dot{\bar{x}}_1 &= \bar{A}_{11}\bar{x}_1 \\
y &= \bar{c}_1\bar{x}_1
\end{align*}

\begin{align*}
\dot{\bar{x}}_2 &= \bar{A}_{21}\bar{x}_1 + \bar{A}_{22}\bar{x}_2 \\
y &= \bar{c}_2\bar{x}_2
\end{align*}

Fig. 13.8.
\[ y = c x \]
\[ \dot{y} = c A x \]
\[ \cdots \]
\[ \vdots \]
\[ y^{(n-1)} = c A^{n-1} x \]

repeated use having been made of \( \dot{x} = A x \).

A neater way of writing this is

\[
\begin{bmatrix}
1 \\
D \\
D^2 \\
\vdots \\
p^{n-1}
\end{bmatrix}
\begin{bmatrix}
y \\
c \\
c A \\
c A^2 \\
\vdots \\
c A^{n-1}
\end{bmatrix}
= \frac{d}{dt} \begin{bmatrix} x \end{bmatrix},
\]

or

\[ H y = Q \dot{x}, \tag{13-79} \]

where \( H \) is \((n \times 1)\) and the square \((n \times n)\) matrix \( Q \) is the reconstructibility matrix. This latter matrix is the same, for time invariant systems, as the observability matrix. Provided that \(|Q| \neq 0\) it follows from eq. \((13-79)\) that

\[ x = Q^{-1} H y. \tag{13-80} \]

This method of state reconstruction suffers from the practical disadvantage that any noise in the measurement, \( y \), seriously degrades the reconstruction.

Another approach is to build a state space model of the given system, drive the model with the actual inputs to the given system and use the model state as an approximation to the real state. The most serious drawback to such an approach is in accurately setting the initial conditions of the model state vector. Another possible disadvantage is that the model dynamics will be just as slow as those of the process.

A third technique, and that which offers benefits such as the smaller dimension of the reconstructor, above those previously described, is to apply to the system what has become known as a Luenberger observer \([9]\). For a multivariable system \( S_1 (A,B,C) \) with \( m \) outputs, this consists of the \((n - m)\)th order observation model:

\[ \dot{z} = A z + B_1 y + B_2 u \tag{13-81} \]

and the linear state estimator:

\[ \dot{x} = C z + D y, \tag{13-82} \]

where the vector \( \dot{x} \) represents a reconstruction of the state \( x \) and the matrices of the observer may be shown \([5]\) to be:
\[
\dot{\mathbf{x}} = (C^* - KC)\mathbf{A}\mathbf{x} + (C^* - KC)\mathbf{B}\mathbf{u} + \mathbf{B}\dot{\mathbf{x}} + \mathbf{D}\mathbf{u} = (C^* - KC)\mathbf{A}\mathbf{x} + (C^* - KC)\mathbf{B} + \mathbf{B}\dot{\mathbf{x}} + \mathbf{D}\mathbf{u} \\
\dot{\mathbf{z}} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}
\]

The \((n - m \times m)\) \(K\) matrix is an arbitrary gain matrix, allowing all the poles of the observer to be assigned favourable locations, and \(C^*, L_1, \text{ and } L_2\) are found from the requirement that

\[
\begin{bmatrix}
\mathbf{C} \\
\mathbf{C}^*
\end{bmatrix}
\]

![Diagram of a Luenberger observer in the control loop.](image)
is a square, non-singular matrix and

$$\left[ \frac{C}{C^*} \right]^{-1} = \left[ L_1 \mid L_2 \right].$$

Fig. 13.9 shows the Luenberge observer applied to a closed loop.

13.8. references and bibliographical notes

Canonical forms for linear multivariable systems are reviewed in the paper by


The material of section 13.3 has been largely abstracted from the books


The formula used in section 13.2 for the inverse of a partitioned matrix can be found in any good textbook on mathematics; for example, see p. 53 of


A proof of the property of determinants needed for the comparison of eqs. (13-34) and (13-36) can be found on p. 40 of the book by


Section 13.5 can serve as an introduction to the article by

[6]: MACFARLANE, A. G. J. and KARCANIAS, N. Int. J. Control, 24, 1, pp. 33-75 (1976),

where algebraic, geometric and complex-variable theories of poles and zeros are surveyed.

The stability of multivariable systems is covered in references [2] to [5]; however, the method of


looks more promising than most others.

Pole assignment techniques are introduced in


and developed in more detail in [5].

The original reference to multivariable system observers is

Simulation is a pretence of what is not, and dissimulation a concealment of what is.

Steele

SIMULATION AND STRUCTURAL ANALYSIS

14.1. introduction

It frequently happens that the analytical techniques presented so far prove inadequate when let loose upon real problems. Remembering that we had confined ourselves (hence the title of this book) to linear, lumped-parameter, deterministic — for the most part — systems, this comes as no great surprise. A possible way out of this impasse is to develop theoretical approaches to the solution of these more advanced problems; another possibility is to use a digital, analogue or hybrid computer to simulate the response of the system. If used intelligently, and in conjunction with the mathematical procedures described already, there is no reason why a large variety of complicated problems cannot be handled — very effectively — by this second method. Problems, for example, whereby a prediction of past or future behaviour of a real process, either open or closed-loop, is important, such as safety studies, plant design etc. etc.

It is important to stress, however, right from the outset, that simulation is not a general panacea for all difficulties. Often, the effort, time and money spent in programming, testing, simulating and understanding the results of a simulation are in excess of what would have to be expended to obtain (approximate) analytical solutions, where the insight into the actual problem is frequently far greater.

As with a normal computer program, the most important goals to aim for with a simulation are

(i) high, but not unnecessary, accuracy;
(ii) fast computation;
(iii) minimum use of computer soft- and hardware.

The third goal is perhaps not too obvious for the uninitiated. Briefly, we require that the computer program (or in the case of an analogue simulation, the patchwork) be as efficient as is compatible with an easy understanding of what is happening in any part of the program. At the same time, it should take up as little memory, peripheral devices etc. etc. — that is to say, hardware — as possible. Obviously, as far as digital computers go, these goals will be more easy to achieve with a purpose built scientific machine than with a computer designed to process administrative work.

In the following sections we shall deal briefly with first analogue and then digital simulations. Nothing will be said, however, about the important topics of discretization, Monte Carlo simulations and simulations of random
processes [2], hybrid simulations [3] or the possibility of using fluidic, pneumatic or mechanical analogues.

The second half of this chapter is devoted to an analysis of the mathematical structure and possible decomposition of large scale problems. This field of study is a relatively new one, so we shall confine ourselves to stationary systems where the results of research are clearest. In essence the problem of the order in which to solve one's model equations will be answered.

14.2. analogue simulations

Analogue computers can be used for simulating non-linear or linear systems. Their usefulness stems from the fact that the mathematical equations relating various electronic components are often identical to those equations of thermal, mechanical, hydraulic and other systems. Thus the change in a temperature of a stirred tank can be simulated by an analogous electric voltage. Just as variables have analogous voltages, parameters can be incorporated by the choice of resistances. To continue the example above, we might find a heat transfer coefficient of the wall of the tank represented by a potentiometer set to, say, one megohm.

Commercially available analogue computers come in two types, one using a range of voltages between -10 volts and +10 volts, the other between -100 volts and +100 volts. Account must be taken of the voltage range when scaling (described shortly) etc. but in other respects it is not too important for the understanding of the principles of analogue simulation. We shall remark upon some of the advantages and disadvantages of analogue simulation later, after we have shown what an analogue computer consists of and how it is used.

The most common components of the analogue computer and the accepted symbols used to represent these components are shown in Fig. 14.1. The table below gives the mathematical operation performed by each component.

<table>
<thead>
<tr>
<th>component</th>
<th>mathematical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>potentiometer</td>
<td>$e_0 = k e_1$, $0 \leq k &lt; 1$</td>
</tr>
<tr>
<td>summer</td>
<td>$e_0 = e_3 - 10e_2 - e_1$</td>
</tr>
<tr>
<td>integrator</td>
<td>$e_0 = -I_0 \int (e_1 - 10e_2) dt$</td>
</tr>
<tr>
<td>multiplier</td>
<td>$e_0 = \frac{e_1 e_2}{100}$</td>
</tr>
<tr>
<td>function generator</td>
<td>$e_0 = f(e_1)$</td>
</tr>
</tbody>
</table>

Components are connected to each other via wires going into a patchboard. By selecting appropriate connections the user has the freedom to incorporate as many integrators, multipliers etc. etc. as are available in the machine. Some of the features of the components are as follows.

Potentiometers are used to introduce parameters into a simulation. The value of the parameter $k$ introduced by a potentiometer lies in the range zero

* For a ±10 volt computer $e_0 = \frac{e_1 e_2}{10}$
to one, and may usually be set correct to four decimal places. Summers, as the name implies, add together (often up to six inputs are permitted) voltages. There is always a *sign inversion* associated with a summer (or integrator). Moreover a choice of input resistance is usually provided — a gain of one or ten times each input voltage may be incorporated. Integrators also usually have this feature; note here that one must usually feed a negative voltage into the IC (initial condition) hole in the patchboard if the initial output of the integrator is specified to be a positive voltage. Multipliers and function generators vary considerably from one manufacturer to the other and the potential user is advised to consult the manufacturer's handbook for wiring and other instructions. Finally, in this connection, we note that voltage sources of -100 volt and +100 volt (or -10, +10 volt) are provided in the patchboard, and that pen recorders or oscilloscopes are usually employed to record the response of the simulated system.

In order to illustrate the procedure to be employed in simulating a system on an analogue computer, we will use as an example a pneumatic control valve (Ch. 3, p. 35). Suppose we wish to investigate the dynamic response of such
a valve to a step change in air signal, where it is desired (at a later stage of the study) to incorporate the effect of valve stem travel hysteresis. Rather than attempt an analytical approach it is decided to go all out, from the start, for an analogue simulation. How should one go about it?

In terms of perturbation variables, the model of the valve (without hysteresis) is given by

\[ x + a_1 x + a_2 x = bu, \]  

(14-1)

where \( x \) is the displacement of the valve stem from the stationary position (measured in metres) and \( u \) is the deviation in the control air pressure (N/m\(^2\)).

Associated initial conditions are

\[ x(0) = 0 \]
\[ \dot{x}(0) = 0. \]  

(14-2)

From the structure of eq. (14-1) it is evident that the interconnections of analogue computer components (the patching diagram) shown in Fig. 14.2 will solve the structurally identical equation:

\[ \ddot{z} + a_1 \dot{z} + a_2 z = \beta f \]  

(14-3)

This equation is composed of what are called machine variables, \( z \) and \( f \).

The next step requires the renowned engineer’s feel for the problem. Clearly, it is impractical to allow a one to one correspondence between the model variables and machine variables. If, for example \( u \) (eq. (14-1)) was \( 10^4 \) N/m\(^2\) we could not allow \( f \) (eq. (14-3)) to be \( 10^4 \) volts: the maximum that \( f \) can attain is +100 volts (for a 100 v.-machine). Hence the idea of scaling the
variables presents itself; what is necessary is that the engineer calculates (or guesses) approximate maximum and minimum values (whichever is likely to have the greatest absolute value) for the variables and their derivatives appearing in eq. (14-1). For example, suppose that the step change in air pressure in which we are interested has a maximum value of approximately $2.5 \times 10^4$ N/m$^2$. With such a step, the maximum deviation of the valve stem is, we guess, 0.01 metres, and the speed of travel will be not greater than 5 m/s. The maximum deceleration experienced by the valve before coming to rest we guess to be 10 m/s$^2$. In terms, then, of equations we have:

$$\left| x_{\text{max}} \right| \approx 0.01 \text{ m}$$
$$\left| \dot{x}_{\text{max}} \right| \approx 5.0 \text{ m/s}$$
$$\left| \ddot{x}_{\text{max}} \right| \approx 10.0 \text{ m/s}^2$$
$$\left| u_{\text{max}} \right| \approx 25000.0 \text{ N/m}^2.$$

Magnitude scaling factors ($k_1, k_2, k_3$ and $k_4$) are introduced, to relate each term in the model equation to terms containing machine variables:

$$\ddot{z} = k_1 \ddot{x}$$
$$\dot{z} = k_2 \dot{x}$$
$$z = k_3 x$$
$$f = k_4 u$$

Assuming that it is appropriate for a machine variable never to exceed ±90 volts (allowing a 10% safety margin) we find that

$$k_1 = \frac{90}{10} = 9.0 \text{ volts/m/s}^2$$
$$k_2 = \frac{90}{5} = 18.0 \text{ volts/m/s}$$
$$k_3 = \frac{90}{0.01} = 9000 \text{ volts/m}$$
$$k_4 = \frac{90}{25000} = 0.0036 \text{ volts/N/m}^2.$$

Now multiplying eq. (14-1) throughout by $k_1$ we have:

$$k_1 \ddot{x} + a_1 k_1 \ddot{x} + a_2 k_1 \dot{x} + b k_1 (k_1 u)$$

which from eq. (14-4) and the values of the magnitude scaling factors calculated above reduces to:

$$\ddot{z} + 0.5a_1 \dot{z} + 10^{-3}a_2 z = 25000bf.$$  \hspace{1cm} (14-5)

Not forgetting the initial conditions, we have from eq. (14-2):
\[ z(0) = k_3 x(0) = 0 \]
\[ \dot{z}(0) = k_2 \dot{x}(0) = 0 \]

Eqs. (14-5) and (14-6) are the scaled machine equations, and are what is actually used to simulate eqs. (14-1) and (14-2), see Fig. 14.3.

It is obvious that in this simulation

\[ a_1 = 0.5 a_1 \]
\[ a_2 = 10^{-3} a_2 \]
\[ \beta = 2500b \]

from eqs. (14-3) and (14-5).

When actual computation is in progress it is almost inevitable that sometimes a machine variable will exceed ± 100 volts. The machine is, of course, protected against such an eventuality, while the user gets a visual/audio warning of the overload. To actually compute, the user, having patched the circuit of Fig. 14.3 and chosen appropriate recording apparatus, sets simultaneously all initial conditions on all integrators by means of the “IC” button or switch. The computation is started by pressing a “RUN” button, while a “HOLD” button allows the computation to be temporarily stopped at any instant of time. To return to the start of a run, there is usually a “RESET” mode, while the values of potentiometers are adjusted in the “POTSET” mode.

Q. 74. Draw a circuit to simulate the system
Just as it is possible to magnitude scale, one can also time scale a simulation. For example our pneumatic control valve simulation proceeded at the same pace as the response of the real valve (i.e. one second of computer time was equal to one second of real time). Now it may be desirable — in accordance with the goals of a good simulation — to either speed up or slow down the simulation. This can be accomplished by the introduction of a time scaling factor, \( \lambda \):

\[
\lambda = \frac{\text{time to be taken for simulation}}{\text{time actually taken by real process}} = \frac{\text{computer time}}{\text{real time}} = \frac{\tau}{t}
\]  

(14-7)

Clearly, if \( \lambda < 1 \) then the simulation proceeds faster than in real life. To show how time scaling works we introduce the following example.

**Example 14.1.**

The following model of a drum boiler-turbine has been recently proposed [4,5]:

\[
\frac{dp}{dt} = -0.0018u_2p^2 + 0.02u_1 - 4.4 \times 10^{-4}u_3
\]  

(14-8)

where \( p \) is the drum pressure in kg/cm\(^2\), \( u_1 \) is the fuel flow (kg/hr), \( u_2 \) is the control valve setting and \( u_3 \) is the feedwater flow (kg/hr). Suggest a feasible simulation scheme.

We first access the model in terms of an approximate transfer function; thus, if the initial equilibrium condition is taken as the reference point and perturbation variables are introduced then:

\[
\frac{d\Delta p}{dt} + 0.0018u_2\Delta p \approx 0.02\Delta u_1
\]  

(14-9)

if \( u_2 \) and \( u_3 \) are unchanging, while

\[
\frac{d\Delta p}{dt} + 0.0018u_2\Delta p \approx -0.0018p_r\Delta u_2
\]  

(14-10)

providing \( u_1 \) and \( u_3 \) remain constant, and

\[
\frac{d\Delta p}{dt} + 0.0018u_2\Delta p \approx -4.4 \times 10^{-4}u_3
\]  

(14-11)

if \( u_1 \) and \( u_2 \) are constant. Here, \( u_2r \) and \( p_r \) are the initial equilibrium values, which can be taken (see Figs. 16 and 17 of [4]) as being approximately 0.925 and 110 kg/cm\(^2\) respectively. It follows (p. 43, Ch. 4) that an approximate time constant for the model is 10 minutes. We now know that after 10 minutes the drum pressure will have reached about 63\% is its new equilibrium value, following a step change in \( u_1, u_2 \) or \( u_3 \).

This is not really acceptable from the point of view of analogue computation, so we decide to make the simulation ten times faster. It follows that \( \lambda \) (eq. (14-7)) is 0.1. Since

\[
\lambda t = r
\]

and

\[
\lambda = \frac{r}{t}
\]
\[
\frac{dp}{dt} = \frac{dp}{dr} \frac{dr}{dt}
\]

It follows from eq. (14-8) that

\[
\frac{dp}{dt} = -0.0018 u_2 p^2 + 0.02 u_1 - 4.4 \times 10^{-4} u_3
\]

\[
\frac{dp}{dt} = -0.018 u_2 p^2 + 0.2 u_1 - 4.4 \times 10^{-3} u_3
\]

(14-12)

which is the time scaled version of eq. (14-8), having a time constant of about 1 minute.

Q. 75. Magnitude scale eq. (14-12) and indicate an analogue circuit diagram to effect the final simulation.

In practice it is not usually necessary to go all through the operations with the time scale factor, \( \lambda \), just described. It is sufficient to

(i) multiply every integrator input with the factor \( \frac{1}{\lambda} \) and

(ii) to change every time dependent input function, \( f(t) \), to \( f(\frac{t}{\lambda}) = f'(\tau) \).

To conclude this section we mention some of the pros and cons of analogue simulations. One of the biggest advantages of analogue computation in general is that we work with continuous-in-time variables. Hence a simulation is an analogy, in that sense, to real life, with the advantage that it can be speeded up or slowed down at will. Non-linear and stochastic systems are also very amenable to the analogue computer approach. Furthermore integration can be handled with such ease, and no specialised knowledge of a computer language is required.

Disadvantages are the time necessary for scaling and patching; possible drift or overload of the operational amplifiers which make up the computer components (summers, integrators etc.); poor accuracy (usually four significant figures maximum); the great number of amplifiers, integrators and multipliers necessary to handle real problems; possibility of patching mistakes and finally the difficulty of simulating processes with dead times.

Many of these disadvantages can be removed by combining a microprocessor(s) with the analogue computer. In this way, for example, patching connections can be made electronically instead of by means of wires, or potentiometers automatically set to the desired values.

In common with all types of simulation it is worth while stressing, as a final point, the value of carefully checking initial and final equilibrium states to eliminate the more obvious sort of errors in setup, parameter values etc.

14.3. Digital computer simulations

The present widespread availability of digital computers has meant a trend in the direction of digital, rather than analogue, simulations in control and dynamics work. A considerable amount of literature concerning digital simulations, from case histories to consideration of computer hardware, has been published and reflects, in a way, the growing acceptance of digital simulation as a normal, day to day activity. We do not intend to go very deeply into the subject here - good books on the subject abound and the depth of treat-
ment required for a good understanding of some of the more difficult topics require many hundreds of pages. Thus we omit any discussion of the simulation of partial differential equations [1], finite-difference methods [6] or finite element methods. Likewise, it is assumed that the systems of interest are deterministic systems.

We begin with a short discussion of the types of numerical (as apart from human) errors which can occur and accrue during the running of any computer program. These errors can usually be traced back to one of three sources: round-off error, truncation error or accumulated error. All three sorts can play havoc with the accuracy of a simulation (this in spite of bland statements frequently encountered such as the ability of digital computers to handle any numbers within the range $10^{-300}$ to $10^{+300}$ — which says nothing, of course, about how accurately a given program will work).

Round-off error results from there being a difference between a desired number and the corresponding machine number sitting in the computer. The latter is determined by a parameter of the computer, the word-bit length.

Consider, for example, a 16-bit word computer; two bits are used for the signs of the characteristic (integer portion) and mantisse (decimal portion) of the number, 4 bits are saved for the characteristic itself and the rest (or 9-bits if a “check bit” is present) may be used to store the mantisse, see Fig. 14.4.

![Fig. 14.4. A 16-bit word.](image)

The result is an accuracy of at most two significant figures! To understand this, remember that a number is stored in a computer according to

$$n = m \times a^k = \text{binary} (m) \times a^{\text{binary} (k)}.$$  

For example, if a equals two, then we have that

$$76.34 = 0.7634 \times 100 =$$

$$= 0.7634 \times 128 =$$

$$= 0.5964 \times 2^7 =$$

$$= 1011101001100 \times 2^{111}.$$
would require a word with 13 bits for mantisse and three bits for characteristic if it was to be free of round-off errors. To circumvent round-off errors computer manufacturers often string words together for floating point (i.e., non-integer) number storage, either with hardware or software (the "double precision" option of many machines).

Truncation errors arise from the use of truncated series instead of exact mathematical functions. For example, if the computer used

\[ 1 - x + \frac{x^2}{2} - \frac{x^3}{6} \]

to calculate \( e^{-x} \), then a truncation error is made, which has an order of \( x^4 \) and greater. Truncation errors are usually listed in computer manufacturers' software guides.

Errors can also accumulate and grow to unacceptably high proportions through repeated iterations or calculations; a small inaccuracy in a parameter, for example, being magnified at each successive line in a program.

These, and other, reasons make it necessary for us to repeat our advice that special care should be taken to check initial and final equilibrium or end values against those predicted by the theory.

We turn our attention now to the actual business of digital simulation of lumped parameter systems. In doing so we bypass the simulation of algebraic (i.e., instantaneous, memoryless or stationary) systems which are treated in the following section of this chapter, and consider from the beginning dynamical systems.

It should be fairly obvious that analogue and digital computers differ in that the latter uses discrete machine variables. Another difference is that no hardware is available in the digital machine to perform the task of integration. The computer user must therefore supply — or call in from the computer library of subroutines — a special program to achieve the numerical integration. The usual set-up is shown pictorially in Fig. 14.5. Since the numerical integration program sits in a loop, the faster, more efficient, more stable and more accurate (within the required bounds) the better. Consequently a vast amount of effort has been devoted to developing integration programs, and the user has a wide choice. Some help as to how to make the best choice of integration routine for a specific simulation can be had by consulting [7], where a comparison of some six different algorithms has been made using twenty test problems.

In the special purpose simulation program packages such as CSMP, MIMIC, GSSP etc. more than one integration algorithm is often available.
Fig. 14.5.

We describe how one very popular integration algorithm in its most simple form works. This is the 4th order Runge-Kutta method which calculates four times for each differential equation per time step, \( \Delta \), a value of the derivative. It is called fourth order because the error made during one integration is proportional to the fourth power of the time step.

Suppose just one differential equation

\[
\dot{x} = f(x, t) \tag{14-13}
\]

together with a single initial condition

\[
x(t_0) = x_0 \tag{14-14}
\]

forms the system model. Remember that the function \( f(x, t) \) may be non-linear. Referring to Fig. (14-6) the following four ordered calculations take place in the Runge-Kutta algorithm per time step.
1. Knowing \( \dot{x}(t) \) — see Fig. (14.6) — an approximate value of \( x(t + \frac{\Delta}{2}) \) is calculated from:

\[
\tilde{x}(t + \frac{\Delta}{2}) = x(t) + \dot{x}(t) \frac{\Delta}{2}.
\]

This value \( \tilde{x}(t + \frac{\Delta}{2}) \) is passed to the user supplied model subroutine where \( \ddot{x}(t + \frac{\Delta}{2}) \) is calculated.

2. A second approximation of \( x(t + \frac{\Delta}{2}) \) is calculated from:

\[
x^*(t + \frac{\Delta}{2}) = x(t) + \frac{\ddot{x}(t + \frac{\Delta}{2}) \Delta}{2}.
\]

Again, this value \( x^*(t + \frac{\Delta}{2}) \) is passed to the model subroutine and \( \ddot{x}^*(t + \frac{\Delta}{2}) \) calculated.

3. The first attempt at calculating \( x \) at the end of the time interval \( \Delta \) is now made, according to

\[
\tilde{x}(t + \Delta) = x(t) + \dot{x}^*(t + \frac{\Delta}{2}) \Delta.
\]

In the model subroutine, the derivative \( \ddot{x}(t + \Delta) \) may be found from \( \dot{x}(t + \Delta) \).

4. The final, and best, value \( x(t + \Delta) \) can now be computed:

\[
x(t + \Delta) = x(t) + \left[ \frac{\ddot{x}(t) + 2\dot{x}(t + \frac{\Delta}{2}) + 2\ddot{x}^*(t + \frac{\Delta}{2}) + \dot{x}(t + \Delta)}{6} \right] \Delta.
\]
There is no great problem in extending the algorithm to handle a number of simultaneous first order differential equations — and hence any lumped parameter model. The procedure is started by supplying the initial conditions $x(0)$ and the start time $t_0$. A quick theoretical method of choosing an appropriate value for the time step $\Delta$ has still to be found. Mostly in practice a trial-and-error approach is used which aims at a $\Delta$ small enough to prevent unacceptable inaccuracies or instability and large enough to allow the computation to proceed at a reasonable pace. Where this practical approach breaks down is when the eigenvalues (or time constants) of the system model are of very differing magnitude. Such systems are said to be systems of stiff differential equations, and various techniques [8],[9] have been proposed to get round the problem.

Only one possibility is mentioned here, since it has received so little attention elsewhere in texts on digital simulation. Basically the idea is extremely simple: the independent variable is replaced in such a way that the eigenvalues of the new system become more bunched up together. For example, if the system model is

$$
\frac{dx_1}{dt} = f_1(x_1, x_2, \ldots, x_n, t) \\
\frac{dx_2}{dt} = f_2(x_1, x_2, \ldots, x_n, t) \\
\vdots \\
\frac{dx_n}{dt} = f_n(x_1, x_2, \ldots, x_n, t)
$$

then it follows that we can also write the system model as:

$$
\frac{dx_1}{dx_j} = \frac{dx_1}{dt} \frac{dt}{dx_j} = \frac{f_1(\ldots)}{f_j(\ldots)} \\
\frac{dx_2}{dx_j} = \frac{f_2(\ldots)}{f_j(\ldots)} \\
\vdots \\
\frac{dx_n}{dx_j} = \frac{f_n(\ldots)}{f_j(\ldots)}
$$

where the variable $x_j$ is chosen to be the new independent variable according to its ability to reduce the spread of the model eigenvalues.
It is sometimes of great advantage to work with a numerical integration algorithm having a *variable step length*, for example the Runge-Kutta-Merson technique. Here, a maximum or limit is set by the user upon the permissible integration error; if this limit is exceeded during the computation then the time step is halved and that integration step repeated. If this proves successful the new time step is doubled before proceeding further. Of course, this automatic selection of a near optimal step length must be paid for; there are now five instead of four calls upon the integration routine per time step, and also the calculations become more complicated.

We close this section with some remarks concerning digital computer simulation packages. These we split into two sorts: the “building-blocks” type (packages such as *CSMP, MIMIC, GSSP* etc.) and the programs written specially for integrating very large dynamic systems. By “building-blocks” we mean to convey the philosophy behind a special sort of simulation package whereby the user interconnects a number of standard (dynamical and/or algebraic) “blocks” — such as a first order lag “block”, a dead-time “block” etc. — to simulate his system. In other words, one performs an analogue-type simulation — without the time or magnitude scaling — upon the digital computer. Such computer simulation packages often considerably simplify the user programming that would otherwise be necessary, but (because of the very general — cater for everything — approach) are sometimes computationally inefficient.

The programs written for large dynamic system simulation or solution [10, 11] are on the other hand very fast and efficient, and sometimes can handle stiff systems with ease.

### 14.4. the structural analysis of systems

We turn our attention now to study the mathematical structure of systems. There are two reasons for such a study; the first, which has already been mentioned, is that we have the feeling that it may be profitable to solve or simulate equations in a prescribed order, rather than in the order which we have happened to write them down. Secondly we wish to eventually be able to answer a question which never appears in any of the books on control engineering: where can controllers best be placed? This question should be seen in the light of real processes — i.e. processes which are “large-scale” — for example chemical factories, oil refineries, distillation column trains, where it is by no means obvious which variables should be manipulated in order to obtain the maximum economic performance, disturbance free response or stable operation.

Only during the last five years has the investigation of the structure of problems received much serious consideration, although profitable industrial applications of such research are already apparent. True, in the 1950’s much energy was expended in the development of graphical techniques — such as block diagrams and signal flow graphs [12,13] — which show some structural features of dynamic systems to advantage. But only recently [14,15] have hierarchical ordering and subsystem structure been investigated; this extra knowledge having been put to use in the design, optimization and control...
of very large systems. In particular, this new branch of analysis has shown up "bottlenecks" in process trains, and predicted the consequences of shutdown or damage to one or more processes or plants in an integrated process network.

In what follows we shall limit ourselves to a discussion of systems which can be completely described by algebraic functions. Three comments are necessary here. Firstly, an extension of the techniques shortly to be presented to treating dynamic systems modelled by differential as well as algebraic functions appears to be possible [16] and leads, for linear systems, to a representation very similar to that of Rosenbrock (Chapter 13). Secondly, we note that an algebraic system description is applicable when the system responds instantaneously to input stimuli. Finally in this connection, it is commonplace that many of the algebraic functions will be of the implicit type — requiring for their solution an iterative trial-and-error procedure. Such a procedure (various types of which are compared in [17],[18]) must be capable of quick convergence, even from a wildly inaccurate initial guess.

14.4.1. The macroscale: assigning distinct representatives

Let us assume that the mathematical model of our system can be written as a number of linear and nonlinear, explicit and implicit algebraic functions (Ch. 2, p. 12):

\[ 0 = f_k(\xi, d), \quad k = 1, 2, 3, \ldots , n, \quad (14-15) \]

where the vector \( d \) contains disturbance variables which cannot be manipulated. The form of eq. (14-15) does not assume an a priori choice for the variable (which we shall henceforth call the distinct representative) which will be explicitly calculated using the function \( k \). The assignment of the distinct representatives, which are all members of \( \xi \) and not of \( d \), is made aside of any physical or intuitive "feel" for what should be chosen. For example, if one of the functions is the Fanning equation describing the flow of fluid through a pipe, say

\[ 0 = f_1(G, P_u, P_d) \quad (14-16) \]

and the upstream pressure can be considered as a disturbance variable, then there are two possible variables — the fluid flow rate \( G \) and the downstream pressure \( P_d \) — from which to choose the distinct representative. Both choices have been drawn in Fig. 14.7; intuitively, since the flow of a fluid comes about as a consequence of a difference in pressure, Fig. 14.7b appears most acceptable. Actually in, for example, the analysis of a plastic extruder model it can be shown that \( P_d \) is the best assignment as distinct representative.

Obviously, the simplest assignment (when there is no choice!) occurs when only one variable — besides disturbances — is associated with a function:

\[ \xi_j = f_i(d). \quad (14-17) \]
Another trivial case occurs when through the assignment of certain variables as distinct representatives no choice exists any more for the assignment of distinct representatives to other functions. That this happens is due to the fact that the same variable can never appear as the distinct (hence the name) representative of two or more functions.

Usually the basis of the assignment must be a sensitivity analysis; when a choice exists between two or more variables the least-sensitive must be chosen as the distinct representative of the function. The sensitivity analysis is performed by calculating absolute numerical values of the weighting functions, \( W_{ij} \), where \( i, j = 1, 2, 3, \ldots, \text{dim}(\xi) \):

\[
W_{i1} = \xi_1 \left( \frac{\partial f_i}{\partial \xi_1} \right) \\
W_{i2} = \xi_2 \left( \frac{\partial f_i}{\partial \xi_2} \right) \\
\text{etc.}
\]

If, for example, \(|W_{i1}| > |W_{i2}| > \ldots > |W_{ij}|\), then \( \xi_1 \) is the least sensitive variable of the function \( f_i \) and is chosen as the distinct representative of \( f_i \).

Q. 76: Derive the relationship between the weighting functions and the original algebraic function.

It will sometimes happen that the above sensitivity analysis indicates that the same variable should be chosen as the distinct representative for two or more functions, as the following example shows.

**Example 14.2.**

The model equations describing the stationary behaviour of a countercurrent heat exchanger (Fig. 14.8) where there is no phase change or loss of heat to the surroundings are:

\[
\text{Fig. 14.7, Possible assignments of the distinct representative.}
\]
where the variables $G_h, G_c, \theta_{c1}, \theta_{h1}$ we have assumed are set by other process units and enter the heat exchanger as disturbance variables. The following numerical data is assumed:

\[ c_{ph} = 1005.0 \quad (J/kg K) \]
\[ c_{pc} = 1880.0 \quad (J/kg K) \]
\[ G_c = 2.0 \text{ kg/s} \]
\[ G_h = 1.0 \text{ kg/s} \]
\[ \theta_{h1} = 310 K \]
\[ \theta_{c1} = 297 K \]
\[ h = 0.88 \quad (J/m^2 s K) \]
\[ A = 310 m^2 \]

It is now possible [19],[20] to estimate the magnitude of the remaining variables ($\xi$):

\[ \theta_{h2} = 307 K \]
\[ \theta_{c2} = 297.8 K \]
\[ H = 3015 J/s \]

and hence calculate the weighting factors. We find that:

\[ |w_{11}| = |\frac{\partial f_1}{\partial \theta_{h2}}| = 6.54 \times 10^7 \]
\[ |w_{12}| = |\theta_{c2} \frac{\partial f_1}{\partial \theta_{c2}}| = 5.06 \times 10^7 \]
\[ |w_{13}| = |H \frac{\partial f_1}{\partial H}| = 3.02 \times 10^3 \]
\[ |w_{21}| = |\theta_{h2} \frac{\partial f_2}{\partial \theta_{h2}}| = 3.09 \times 10^5 \]
\[ |w_{23}| = |H \frac{\partial f_2}{\partial H}| = 3.02 \times 10^3 \]
\[ |w_{32}| = |\theta_{c2} \frac{\partial f_3}{\partial \theta_{c2}}| = 1.12 \times 10^6 \]
\[ |w_{33}| = |H \frac{\partial f_3}{\partial H}| = 3.02 \times 10^3 \]

It is apparent from the above weighting factors that the variable \( \theta_{h2} \) is the least sensible variable of both the first and second function; it would be nice if it were possible to choose \( \theta_{h2} \) as the distinct representative of both. However this can never be so and we must seek the "second-best" choice as it were. By normalising (i.e. dividing by the weighting factor having the greatest magnitude) it is often possible to order the weighting factors per function. The following table gives the normalised weighting factors:

<table>
<thead>
<tr>
<th>( \theta_{h2} )</th>
<th>( \theta_{c2} )</th>
<th>( H )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>( 1 )</td>
<td>( 0.773 )</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>( 1 )</td>
<td>( - )</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>( - )</td>
<td>( 1 )</td>
</tr>
</tbody>
</table>

From this table it can be seen that it is better to choose \( \theta_{h2} \) as the distinct representative of \( f_1 \) and \( H \) as the distinct representative of \( f_2 \):

\[
\theta_{h2} = f'_1(\theta_{c2},H,d)
\]
\[
H = f'_2(\theta_{h2},d)
\]
\[
\theta_{c2} = f'_3(H,d).
\]

This is not the only feasible assignment, however, and at the present time there is no sure way of saying that it is any better than the assignment

\[
\theta_{c2} = f'_1(\theta_{h2},H,d)
\]
\[
\theta_{h2} = f'_2(H,d)
\]
\[
H = f'_3(\theta_{c2},d).
\]

With the making of an assignment of all distinct representatives one has the guarantee that for a given (wild) initial guess of the value of each representative, the system model as a whole will converge, using an appropriate iterative technique, to the correct stationary values of all \( \frac{d}{dt} \) in as few iteration steps as is possible.

14.4.2. partitioning into subsystems

We can now begin to introduce some structure into the stationary model.
The first step is the investigation of the presence possible subsystems. We define a subsystem here to mean a group of functions which communicate with other groups of functions by a feedforward flow of information only. Fig. 15.9 shows such a "splitting" or partitioning of a model into subsystems \( g_1, g_2, \ldots, g_8 \).

![Diagram of subsystems](image)

Fig. 14.9.

The reason for such a partitioning is that we can organise our analysis: in Fig. 14.9 for example, subsystem \( g_1 \) can be analysed completely independently of all other subsystems. So can subsystem \( g_2 \). Armed with this knowledge, one can attempt to analyse subsystem \( g_3 \), and so on.

The algorithm to achieve this partitioning into subsystems is best explained by a simple example. For complicated large scale problems such an algorithm would be written into a computer program [21].

**Example 14.3.**

In an approximate steady state model of an oil refinery steam distribution network (Fig. 14.10) fourteen algebraic functions are used. The functions, after assignment of
Fig. 14.10. Steam distribution network of an oil refinery.
the distinct representatives, can be written as follows:

\[ \xi_1 = f_1(d) \]
\[ \xi_2 = f_2(\xi_6, \xi_{12}, d) \]
\[ \xi_3 = f_3(\xi_1, \xi_{10}, d) \]
\[ \xi_4 = f_4(\xi_{13}, \xi_{14}, d) \]
\[ \xi_5 = f_5(\xi_{12}) \]
\[ \xi_6 = f_6(\xi_{11}, d) \]
\[ \xi_7 = f_7(\xi_1, \xi_{13}) \]
\[ \xi_8 = f_8(\xi_{10}, d) \]
\[ \xi_9 = f_9(\xi_2, \xi_6) \]
\[ \xi_{10} = f_{10}(\xi_{14}, d) \]
\[ \xi_{11} = f_{11}(\xi_6, \xi_{12}, d) \]
\[ \xi_{12} = f_{12}(\xi_5, d) \]
\[ \xi_{13} = f_{13}(\xi_7, d) \]
\[ \xi_{14} = f_{14}(\xi_3, \xi_8, d) \]

The first step in the partitioning of this model into subsystems is the registration (by means of a Boolean matrix) of the influence of various variables on each function. The creation of this matrix – called here the characteristic occurrence matrix (COM for short) – is achieved by placing a "one" against every variable having a direct or indirect influence upon a function.

<table>
<thead>
<tr>
<th>variables</th>
<th>( \xi_1 )</th>
<th>( \xi_2 )</th>
<th>( \xi_3 )</th>
<th>( \xi_4 )</th>
<th>( \xi_5 )</th>
<th>( \xi_6 )</th>
<th>( \xi_7 )</th>
<th>( \xi_8 )</th>
<th>( \xi_{10} )</th>
<th>( \xi_{11} )</th>
<th>( \xi_{12} )</th>
<th>( \xi_{13} )</th>
<th>( \xi_{14} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_2 )</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_3 )</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_4 )</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_5 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_6 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_7 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( f_8 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>( f_9 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{10} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{11} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{12} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{13} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{14} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The characteristic occurrence matrix (COM)

Notice how the distinct representatives are always to be found on the diagonal of the COM. The disturbances, \( d \), are assumed known and, just like the parameters and coefficients of the functions, have no further bearing upon the analysis. To see how row eight, say, of the COM is arrived at we use the notation

\[ f_k \]
\[ \xi_i \to \xi_i \]
to mean: the variable $\xi_j$ is influenced via function $f_k$ by variable $\xi_i$. Then we have that

Every variable (except elements of $d$) in the above scheme is recorded by placing a Boolean “one” in row eight of the COM.

The second step in the partitioning algorithm is the collecting together in groups of all functions having the same column entries in the COM (i.e. being influenced by the same variables). By ensuring that as many “ones” as possible lie either on or below the main diagonal it is possible to order the groups of functions into a matrix form which we shall call the quasi-triangular characteristic occurrence matrix (abbreviated to $\Delta$COM), which in this example looks as follows:

The quasi-triangular characteristic occurrence matrix (\(\Delta\)COM).

Using the $\Delta$COM it is an easy matter to draw the original model in terms of subsystems – see Fig. 14.11 overpage.

Of course, we have “cooked” Example 14.3, so that eight nicely sized subsystems result. In practice it is usually possible to partition a model into subsystems, although it frequently happens that one of these subsystems is almost as big as the original model.
14.4.3. disjointing the model

If we look rather closely at Fig. 14.11 it is apparent that two, completely independent, sets of subsystems - \( \{g_1, g_3, g_5, g_8\} \) and \( \{g_2, g_4, g_6, g_7\} \) - exist. In other words, the original model can be manipulated into two disjoints which we can analyse and manipulate completely independently from one another. Although for small problems this is a fairly obvious observation, in large-scale models it is convenient to mechanize the disjointing procedure. This may be done from the compact form of the DCOM. Continuing Example 14.3 we would have
for the compact ΔCOM, giving the two disjoints:

14.4.4. the hierarchical structure of a system

We conclude our analysis on the macro-scale of system structure by ordering the various subsystems of each disjoint into what we shall call levels. To do this we impose the rule that a subsystem must always receive at least one variable from a subsystem that is situated one level higher.

Thus, in Fig. 14.12b, the subsystem \( g_3 \) belongs to level \( n \) and not to level \( n + 1 \) if subsystem \( g_3 \) is no longer influenced by the variable set \( z_1 \).

In order to create a hierarchy of levels within each disjoint we need to isolate the minimum number of diagonal submatrices in each disjoint. For the two disjointed matrices originating from Example 14.3 we would find:

where the desired diagonal submatrices are indicated by broken lines. The extreme upper left diagonal submatrix is the upper or first hierarchical level, and all subsystems (and therefore the functions belonging to them) to be found in this submatrix belong also to the first level.

We are now in a position to present a picture of the macrostructure of the whole model – built up by partitioning, disjointing and hierarchical ordering. The so-called multi-level matrix, where Boolean “ones” are given only to those variables having a direct influence on the function in question, is useful for...
(14.12a) Possible hierarchical distribution.  
(14.12b) Impossible distribution.

Fig. 14.12. Illustration of hierarchical levels.

For this purpose. For Example 14.3 we have the following two multi-level matrices:

```
<table>
<thead>
<tr>
<th>subsystems</th>
<th>z1</th>
<th>z3</th>
<th>z5</th>
<th>z8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ξ1</td>
<td>ξ13</td>
<td>ξ3</td>
<td>ξ8</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ξ10</td>
<td>ξ14</td>
<td>ξ4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>levels</th>
<th>f1</th>
<th>f13</th>
<th>f7</th>
<th>f3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

| disjoint 1  | 1  | 1  | 1  | 1  |
```
From these two matrices we can construct Fig. 14.13 to show the structure of the model in a more usual form (see overpage).

14.4.5. the microscale

The analysis of the structure of systems on a microscale, i.e. the structure within an (irreducible) subsystem found by partitioning techniques, is still in its infancy, and it is not proper to include much about this topic here. The analysis is obviously concerned with what we might call loosely *internal feedback*; we know from our definition of a subsystem that if more than one function is present in a subsystem, then feedback must be present. Most often this feedback has not originated from the physical feedback links present in a closed loop system but is the result of mathematical, inherent or internal feedback of information between functions. It is possible to so order the functions that the number of feedback paths is a minimum; this ordering will not usually be compatible with the assignment of distinct representatives described earlier, however.

Finally to give some idea of the connection between the matrix representation used previously and internal feedback we show below subsystem $g_9$ again with the one feedback loop, $\xi_{14}$. Notice how the feedback to $f_{10}$ occurs through the Boolean “one” above the main diagonal.

<table>
<thead>
<tr>
<th>$\xi_{10}$</th>
<th>$\xi_3$</th>
<th>$\xi_8$</th>
<th>$\xi_{14}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{10}$</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$f_{3}$</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$f_{8}$</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$f_{14}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Q. 77. Make a sketch of the above ordering of functions. Reorder the functions to obtain two feedback loops.

14.5. references and bibliographical notes

The simulation of distributed parameter systems is treated in


A general technique for the digital simulation of multidimensional Gauss-Markov random

197
Fig. 14.13.
The boiler-turbine model used in Example 14.1 is taken from the articles by:


and


The stability of finite difference methods has received much attention. One (somewhat old) very readable article is that of:


A comparative study of numerical integration programs can be found in:


Systems of stiff differential equations pertaining to chemical reactor systems are treated in a paper by:


while more generally applicable approaches are to be found in:


A very useful set of digital computer programs may be obtained from the authors of the book:


A typical, specialised simulation program is described in the paper by:


The development of block diagrams and the relation with structural analysis is considered in:


while the same treatment for signal flow diagrams appears in:


Some of the first ideas on structural analysis can be found in the books edited by:


and


For an extension to the method of structural analysis for dynamical systems see:


Iterative procedures for the solution of implicit algebraic functions are described, beginning on p. 123, in the book by:


while a discussion of convergence appears in
The solution of the implicit equations of a heat exchanger are considered in


and


The method of structural analysis presented in this chapter is based on, and inspired by,
the papers of

[21] KEVORKIAN, A. K. and SNOEK, J. Decomposition as a tool for solving large scale
The possible is not necessarily desirable and the desirable is perhaps not yet possible.

DYNAMIC OPTIMIZATION

15.1. Introduction

Whenever we wish to transfer a system or a process from a given state to another state we incur certain costs. The term "cost" must be seen in a broad sense; for example, if the transference must occur quickly, then the factor time will occur in the costs. If one prizes the thrifty use of control action, then the controller energy is part of the costs. In general we can say that the costs are a function of the state trajectory, the control action and the time. Using the techniques of dynamic optimization we try to minimize the costs incurred in taking the system from one state to another, by making a suitable choice of control strategy. A precise definition of the dynamic optimization problem is, given a system

\[ \dot{x}(t) = f(x(t), u(t), t) \]  

and a cost functional*

\[ J = \int_{t_0}^{t_1} f_0(x(t), u(t), t) \, dt \]  

we require the system to be transferred from \( x(t_0) \) to \( x(t_1) \) such that the cost functional \( J \) is minimal and \( u(t) \in U \) for \( t_0 \leq t \leq t_1 \). Some brief words of explanation follow.

\( x(t_0) \) and \( x(t_1) \) are the initial and final states respectively. The control signal resulting from the dynamic optimization, indicated henceforth by \( u_0(t) \),

* It may be useful to recall the following two mathematical definitions.

A function \( f \) is a rule of correspondence that assigns to each element \( q \) in a certain set \( D \) a unique element in a set \( R \).

A functional \( J \) is a rule of correspondence that assigns to each function \( f \) in a certain class \( \Omega \) a unique real number. For example in

\[ J = \int_{t_0}^{t_f} x(t) \, dt \]

the real number assigned by the functional \( J \) is the area under the curve \( x(t) \). Loosely speaking a functional is a "function of a function".
will be called the optimal control signal. Associated with \( u^0(t) \) will be an optimal state trajectory, \( x^0(t) \). The vector space of all allowable control signals has been denoted by \( U \).

There are a number of ways of tackling the dynamic optimization problem. First and most obvious is application of the classical calculus of variations, which results in Euler-Lagrange equations; for multidimensional, controlled systems these must in general be augmented with Lagrangian multipliers. Unfortunately, to apply these techniques it is necessary to permit unbounded control signals, \( u(t) \). More recent developments in dynamic optimization include Pontryagin’s Maximum Principle and dynamic programming. These two methods were introduced in the fifties by Pontryagin and Bellman, and can cope with bounded (or restricted) control signals.

If any of these techniques of dynamic optimization are applied to linear processes with a quadratic performance index—that is, the function \( f_o \) of eq. (15-2) has a quadratic form—then a linear optimal control law results, which may be expressed in terms of a matrix-Riccati equation. This optimal feedback law requires a complete knowledge of all the states of the process—these are not, however, usually all available for measurement—and so it is necessary to reconstruct the unavailable system states using an observer. Observers have, it may be recalled, already been discussed (Chapter 13); there is, unfortunately, often a complication that arises since measurements can be corrupted with noise. In that case the Kalman filter is appropriate; it produces the “best” estimate of the unmeasurable states from measurements contaminated by noise. Should all the variables in the system be stationary (Appendix B, Section B.5) it is possible to apply a Wiener filter to achieve optimal estimates of the unaccessible states.

In the following sections we shall delve deeper into the subjects touched upon here with emphasis upon the most simple case of a linear process and a quadratic performance index. The purpose is to give some indication of the scope and method of problem solving used in one branch of modern control engineering. Although it is not possible to present mathematically watertight proofs of many of the results we will try to present the results meaningfully and concentrate on the relationships between results.

### 15.2. variational calculus

#### 15.2.1. optimization without constraints

The most elementary problem of dynamic optimization can be stated as follows: given the cost functional

\[
J = \int_{t_0}^{t_f} f_o(x, \dot{x}, t) \, dt
\]

(15-3)

we wish to know the trajectory of \( x(t) \) —the optimal trajectory \( x^0(t) \)—which minimizes \( J \). Values for \( x(t_0) \) and \( x(t_f) \) are assumed known. Since the states, \( x(t) \) are not constrained to be part of any dynamical system, we talk of optim-
Using variational calculus we can easily prove [3] that any solution of the above problem must also necessarily satisfy the so-called Euler-Lagrange equations, of which there are \( n \):

\[
\frac{\partial f_0}{\partial x_i} - \frac{d}{dt}\frac{\partial f_0}{\partial \dot{x}_i} = 0, \quad i = 1, 2, \ldots, n
\]  \( (15-4) \)

Remember that the states, \( x(t) \), are completely unrestricted in their behaviour.

To translate this classical problem so that it can be applied to constrained systems, i.e. with \( y \) entering eq. (15-3) and accounting for the dynamics of the system, eq. (15-1), appears at first sight to be fairly easy; indeed many authors state that it is possible. One simply solves eq. (15-1) for \( y \), substitutes the result in a cost functional \( J(x, y, t) \) such as eq. (15-2) which now is of the form given above, eq. (15-3).

However, it is possible to show (we have given a proof for this conjecture in Appendix C since none can be found in the standard textbooks on the subject) that for the special — but very important — case of a linear process with a quadratic performance index the Euler-Lagrange equations cannot be used to predict the optimal control policy for a general, multidimensional, controlled process.

Q. 78: Try to calculate the optimal control policy for the linear system given by

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u
\end{align*}
\]

subject to \( J = \int_{t_0}^{t_f} (x_1^2 + a^2 u^2) dt \).

We look, therefore, to a modification of the basic procedure which would allow meaningful problems to be tackled. The extra sophistication necessary is the addition of so-called Lagrangian multipliers; with these we can tackle a far broader class of problems, as now shown.

### 15.2.2. optimization with constraints

Consider now the same cost functional, eq. (15-3), as before but this time subject to \( k \) constraints, expressed by

\[
\phi_k (x, \dot{x}, t) = 0, \quad k = 1, 2, \ldots, q, \quad q \leq n.
\]  \( (15-5) \)

The problem formulation remains the same: minimize \( J(x, \dot{x}, t) \) by means of a suitable choice of \( \dot{x}(t) \), the optimal trajectory, \( \ddot{x}(t) \). In order to solve this new problem by variational calculus techniques, we define an expanded cost functional:

* These are necessary, but not sufficient conditions. It remains, namely, to show that the extremum \( \tilde{J}(x^0, \dot{x}^0, t) \) so found is in fact a global minimum. These conditions are given in many books [1], [10].
\[ J^* = \int_0^t F(t) dt, \quad (15-6) \]

where \( F = f_0(x, \dot{x}, t) + \sum_{k=1}^q \lambda_k(t) \phi_k(x, \dot{x}, t). \quad (15-7) \]

The new variables, \( \lambda_k, \ k = 1, 2, \ldots, q \) are the Lagrangian multipliers. Now according to (15-5) \( \phi_k(x, \dot{x}, t) = 0 \) for all \( k \), so that \( J^* = J \). This implies in turn that minimizing \( J^* \) is equivalent to minimizing \( J \), with the constraints appearing in the cost functional \( J^* \). This all means that we have reformulated the problem as the minimization of \( J^* \) without constraints, and so the results of Section 15.2.1 apply. Thus

\[ \frac{\partial F}{\partial x_i} - \frac{d}{dt} \frac{\partial F}{\partial \dot{x}_i} = 0, \quad i = 1, 2, \ldots, n, \quad (15-8) \]

where the trajectories \( x(t) \) may now be a set of constrained variables.

When we try to apply the above technique to our dynamic optimization problem stated in the introduction, we have more success than previously. Writing eq. (15-1) as

\[ \dot{x}_k = f_k(x, u, t), \quad k = 1, 2, \ldots, n \quad (15-9) \]

we define a set of functions \( \phi_k \) by

\[ \phi_k(x, \dot{x}, u, t) = \dot{x}_k - f_k(x, u, t), \quad k = 1, 2, \ldots, n. \quad (15-10) \]

Note that we have used all the available constraint functions, eq. (15-5), to express the system dynamics. Further constraints in the form of restrictions etc. on \( x(t) \) or \( u(t) \) can only be handled if they can somehow be written into eqs. (15-3) or (15-9). The only other point to note is that we now have, on first sight, more unknown variables — \( n \) states and \( r \) controls — than equations. However there are now two vector Euler-Lagrange equations:

\[
\begin{align*}
\frac{\partial F}{\partial x_i} - \frac{d}{dt} \frac{\partial F}{\partial \dot{x}_i} &= 0, \quad i = 1, 2, \ldots, n \\
\frac{\partial F}{\partial u_j} - \frac{d}{dt} \frac{\partial F}{\partial \dot{u}_j} &= 0, \quad j = 1, 2, \ldots, r
\end{align*}
\]

\[ (15-11) \]

where \( F = f_0(x, u, t) + \sum_{k=1}^n \lambda_k \phi_k. \quad (15-12) \]

Example 15.1.

Derive the Euler-Lagrange equations for the linear system used in Q. 78, with the same quadratic performance index.

Defining two functions by eq. (15-10),

\[ \phi_1 = \dot{x}_1 - x_2 \\
\phi_2 = \dot{x}_2 - u. \]
Hence from eq. (15-12)

\[ F = f_0 + \lambda_1 \phi_1 + \lambda_2 \phi_2 = x_1^2 + a^2 u^2 + \lambda_1 (\ddot{x}_1 - x_2) + \lambda_2 (\ddot{x}_2 - u). \]

The Euler-Lagrange equations are, therefore, from eq. (15-11):

\[ \begin{align*}
2x_1 - \ddot{x}_1 &= 0 \\
-\lambda_1 - \ddot{\lambda}_1 &= 0 \\
2a^2 u - \lambda_2 &= 0.
\end{align*} \]

The above three equations, plus those of the system, are sufficient to provide a solution for the five unknowns, \( x_1, x_2, u, \lambda_1 \) and \( \lambda_2 \).

Until now we have not indicated how it is possible in practice to achieve the optimum trajectories which result from the simultaneous solution of eqs. (15-10) and (15-11). By eliminating the Lagrangian multipliers and substituting the given boundary conditions (i.e. \( x(t_f) = 0 \)) we are left with an optimal feedforward control law

\[ u^0 = g(x_0, t). \quad (15-13) \]

For the special case of a linear process and a quadratic performance index it is possible to manipulate the optimal feedforward control law into a linear optimal feedback control law:

\[ u^0 = k x. \quad (15-14) \]

Q. 79: Prove that eq. (15-14) results from the minimization of a quadratic performance index subject to the constraint of a linear dynamical system.

Q. 80: Suppose that the purpose of the control of the system in Q. 78 is to drive the system from any arbitrary initial state \( \mathbf{x}(t_0) \) to the state \( \mathbf{x}(t_f) = 0 \), where \( t_f \to \infty \). Prove that the optimal feedback control law is as shown in Fig. 15.1.

Fig. 15.1.
Before closing this discussion of constrained optimization by classical techniques we make the following comments. Firstly, restrictions in state or control signal trajectories — other than those imposed by the dynamics of the system itself — are difficult to incorporate. Secondly, although some authors disagree [3], the principle of feedback does not follow from the analysis automatically; the mathematics says that feedforward control is just as good. Thirdly, in general we arrive sooner or later at the two-point boundary problem [1] when we try to solve our set of differential equations, which may be nonlinear. This two-point boundary problem presents a major mathematical difficulty. Finally, we note that the classical approach does not lead automatically to an optimal control law — this we must seek ourselves.

15.3. Pontryagin's maximum principle

An approach which is closely related to that just described has been developed by L. S. Pontryagin in the Fifties. We shall show how this approach can handle even restricted (limited) control signals, although no proof of his Maximum Principle is attempted — the interested reader should consult, for example, ref. [1].

The basic problem remains the same as stated in the introduction: given a system and cost functional

\[ \mathcal{J} = f(x,u,t) \]  \hspace{1cm} (15-1)

and

\[ J = \int_{t_0}^{t_1} f_0(x,u,t) \, dt \]  \hspace{1cm} (15-2)

we require the minimization of \( J \) by means of the appropriate choice of \( u(t) \) — the optimal control \( u^0(t) \). Following Pontryagin, we define a so-called Hamiltonian function \( H \):

\[ H = -f_0(x,u,t) + \sum_{k=1}^{n} p_k f_k(x,u,t) \]  \hspace{1cm} (15-15)

Pontryagin's Maximum Principle states that the cost functional \( f_0(x,u,t) \) is minimized by the control signals \( u(t) \) which maximize the Hamiltonian, eq. (15-15), for all \( t_0 \leq t \leq t_1 \). He was further able to show that the necessary conditions for this to occur can be expressed in the form of two sets of equations (Pontryagin's equations):

\[ \dot{p}_i = -\frac{\partial H}{\partial x_i}, \quad i = 1,2, \ldots, n \]  \hspace{1cm} (15-16)

and

\[ \dot{x}_j = \frac{\partial H}{\partial p_j}, \quad j = 1,2, \ldots, n \]  \hspace{1cm} (15-17)

with the associated boundary conditions, for which two possibilities exist [3]:

\[ \text{at } t = t_0, \quad p_i = \text{given} \]

\[ \text{at } t = t_1, \quad x_i = \text{given} \]
(a) Fixed end-point. \(x(t_0)\) and \(x(t_1)\) are given but boundary conditions for \(p\) are not known, or
(b) Free end-point. \(x(t_0)\) given and \(p(t_1) = 0\).

If the control \(u(t)\) is unrestricted then the maximization of \(H\) amounts to the solution of the equations

\[
\frac{\partial H}{\partial u_i} = 0, \quad i = 1, 2, \ldots, r, \tag{15-18}
\]

in which case Pontryagin's Maximum Principle can be derived from the Euler-Lagrange equations.

**Example 15.2.**

We rework Example 15.1 using Pontryagin's Maximum Principle. The Hamiltonian function, from eq. (15-15), is

\[H = -x_1^2 - a^2 u^2 + p_1 x_2 + p_2 u.\]

Since the control is unrestricted, \(H\) is a maximum when

\[
\frac{\partial H}{\partial u} = -2a^2 u + p_2 = 0 \quad \text{for all time},
\]

where the differential equations describing \(p_1\) and \(p_2\) are (eq. (15-16))

\[
\begin{align*}
\overset{\circ}{p}_1 & = 2x_1 \\
\overset{\circ}{p}_2 & = -p_1.
\end{align*}
\]

Together with the system equations, we now have five equations and five unknowns. Actually, the equations are identical to those of Example 15.1 if \(p_1\) and \(p_2\) are replaced by \(\lambda_1\) and \(\lambda_2\), and the solution proceeds further as before.

Suppose, for the sake of illustration, that the control is restricted — let

\[|u| \leq u_{\text{max}}.\]

In that case eq. (15-18) no longer holds. The function \(H\) is now maximal for

\[
\begin{align*}
u & = \frac{p_2}{2a^2} \quad \text{if } |p_2| < 2a^2 u_{\text{max}} \\
v & = \pm u_{\text{max}} \quad \text{if } |p_2| > 2a^2 u_{\text{max}}.
\end{align*}
\]

Again, the optimal control law can be calculated in the usual fashion, although this time, of course, a nonlinear controller will result.

Let us consider the relationship between the classical calculus of variations approach and that of Pontryagin a little further. Using eqs. (15-10) and (15-12) in the Euler-Lagrange equations, eq. (15-11), gives:

\[
\begin{align*}
\overset{\circ}{\lambda}_i & = \frac{\partial}{\partial x_i} \left[ f_0 - \sum_{k=1}^{n} \lambda_k f_k \right], \quad i = 1, 2, \ldots, n \\
\frac{\partial F}{\partial u_j} & = \frac{\partial}{\partial u_j} \left[ f_0 - \sum_{k=1}^{n} \lambda_k f_k \right] = 0, \quad j = 1, 2, \ldots, r.
\end{align*} \tag{15-19}
\]
If we define a Hamiltonian function by

\[ H = -f_0 - \sum_{k=1}^{n} \lambda_k f_k \]  

then eq. (15-19) can be interpreted as

\[
\frac{\partial H}{\partial x_i} = -\lambda_i, \quad i = 1, 2, \ldots, n
\]

\[
\frac{\partial H}{\partial u_j} = 0, \quad j = 1, 2, \ldots, r
\]

while from eq. (15-20) follows that

\[
\frac{\partial H}{\partial \lambda_k} = x_k, \quad k = 1, 2, \ldots, n.
\]

We see from this analysis that just as eq. (15-11) is a necessary condition for optimality, so is eq. (15-21). Indeed, so long as \( u(t) \) is not restricted, the maximum principle and eq. (15-16) are equivalent to eq. (15-11); furthermore eq. (15-17) is the same as eq. (15-22). We close this section with an example of a process having a restricted control signal.

**Example 15.3.**

Given a process described by

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u,
\end{align*}
\]

i.e. the same as in Examples 15.1 and 15.2. This time, however, the control is subjected to the constraint:

\[ |u| \leq u_m. \]

The problem is also slightly different; namely we wish to bring the process from an arbitrary initial state \( x(t_0) \) to a final state \( x(t_f) = 0 \) in the minimum time. Such a minimum-time problem, as it is called, is easily formulated in the framework of the theory presented so far by requiring that the cost functional

\[ J = \int_{t_0}^{t_f} 1 \, dt = \Delta t \]

is minimal.

The Hamiltonian function is obviously

\[ H = -1 + p_1 x_2 + p_2 u \]

with the \( u(t) \) which maximizes \( H \) at all times given by:

\[ u = u_m \text{sgn} (p_2). \]

From eq. (15-16),

\[ \text{sgn} (p_2) = \begin{cases} 1 & \text{if } p_2 > 0 \\ 0 & \text{if } p_2 = 0 \\ -1 & \text{if } p_2 < 0 \end{cases} \]
\[
\begin{align*}
p_1 &= 0 \\
p_2 &= -p_1
\end{align*}
\]

must hold, and on being integrated these give
\[
\begin{align*}
p_1 &= a \\
p_2 &= at + b.
\end{align*}
\]

Thus the system equations become
\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u_m \text{sgn } (at + b),
\end{align*}
\]

where the arbitrary constants \(a\) and \(b\) must be so chosen that \(x(t_f) = 0\) when \(x(t_0)\) is given. Rather than pursue this further here, we examine briefly the optimum control signal:
\[
u = u_m \text{sgn } (at + b).
\]

The function \((at + b)\) crosses the time axis at most once for the period \(t_0 \leq t \leq t_f\). The optimal control signal is therefore \(u^0(t) = \pm u_m\) with 0 or 1 switches. In general, for an \(n\)-th order system \((n - 1)\) switchings are necessary to bring the system to the final state in minimal time.

A control which takes only the maximum values is called a bang-bang control. In general minimum-time problems where the control signal is limited lead to bang-bang optimal controllers.

### 15.4. Dynamic programming

Another new, and very important, approach to the dynamic optimization problem, is that suggested by R. Bellman [9]. His dynamic programming can also handle both restricted states and control signals; in its continuous form dynamic programming is equivalent to the maximum principle. We consider here, however, only discrete dynamic programming, which serves as an introduction to discrete systems (which are treated in the next chapter) and is useful in that, for our purposes, it is easily applicable to processes containing dead times.

Dynamic programming is a mathematical theory of multi-stage decision processes, and is therefore particularly suited to optimization problems which are naturally formulated into a number of discrete steps. The theory is applied in the form of an algorithm, most suited for execution in a digital computer, rather than as a formula or equation, and hence its name.

Consider a continuous process having \(n\) states, denoted by the vector \(x(t)\). If the value of each state is only available at discrete times \(t_0, t_0 + \Delta, t_0 + 2\Delta, \ldots, t_0 + N\Delta, t_0 + (N + 1)\Delta, \ldots\), then we could think of the process as being discretized into stages as shown in Fig. 15.2, overpage.

Here, we have shown the usual shorthand notation for both states, \(x\), and controls \(u\); that is to say that \(x(t_0 + k\Delta)\), for example, is shown as \(x(k)\). Note in Fig. 15.2 that \(x(1)\) is the initial state and \(x(N + 1)\) is the final state. Obviously \(N\) decisions are needed to transfer \(x\) from the initial to the final state. We
shall call the collection of decisions, one for each stage, which can achieve such a transfer, the control strategy. In analogy to the continuous case, we can now formulate the dynamic optimization problem of the introduction into a dynamic programming problem: given the discrete process

\[
\begin{align*}
x(k+1) &= \Gamma(x(k), u(k), k) \\
J &= \sum_{k=1}^{N} g_k(x(k), u(k))
\end{align*}
\]

where \( \Gamma \) represents some mathematical operator, and a cost functional

\[
J = \sum_{k=1}^{N} g_k(x(k), u(k))
\]

(where \( x(k) \) and \( u(k) \) take on discrete values) then we require the minimization of \( J \) by means of the choice of the optimum control strategy \( u^*(k), k = 1, 2, \ldots, N \), where \( u(k) \in U \) for \( k = 1, 2, \ldots, N \). Theoretically it is possible to calculate every possible control strategy and then to choose the optimal decision or control strategy as the one which minimizes \( J \); however, for any real process this would quickly lead to a vast amount of computation. Bellman's Principle of Optimality luckily comes to our aid, allowing us to solve the problem stage by stage. This states that:

An optimal control strategy has the property that, whatever the initial state and the initial decision, the remaining decisions must form an optimal control strategy with respect to the state resulting from the first decision.

It follows that the best place to start the calculations is at \( x(N+1) \), i.e. the final state. We calculate the series \( u(N+1-k), \ldots, u(N) \) in order to arrive at \( x(N+1) \) from \( x(N+1-k) \) with the minimum of “cost”. Beginning with \( k = 0 \) the values of \( u(k) \) giving the minimum cost are stored up; the fact that not all possible \( u(k) \) need to be considered leads to a considerable reduction in computation.

We can mathematically formulate what we have tried to say above in words as follows. Suppose \( V_k \{x(N+1-k)\} \) represents the “costs” incurred over \( k \) stages from the state \( x(N+1-k) \) to the final state \( x(N+1) \) by using some control strategy. Furthermore let \( V_k^0 \) be the minimum of \( V_k \); then the general recursive formula or algorithm of dynamic programming can be written as:

\[
V_k^0 \{x(N+1-k)\} = \min_{u(N+1-k)} \left[ g_{N+1-k} \{x(N+1-k), u(N+1-k)\} + V_{k-1}^0 \{x(N+2-k)\} \right]
\]
where \( g_k(\cdot) \) is the performance index of eq. (15-24) and where from eq. (15-23) it follows that

\[
x(N + 2 - k) = \Gamma \{x(N + 1 - k), u(N + 1 - k), N + 1 - k \}
\]  

(15-26)

These two formulas, starting with \( x(N - 1) \), allow the optimal control strategy to be calculated.

**Example 15.4.**

Fig. 15.3.

In Fig. 15.3 the costs associated with travelling by various routes from A to D are given. We wish to find the cheapest travel route using dynamic programming.

It is convenient to solve the problem by constructing a table as is shown in Fig. 15.4.

<table>
<thead>
<tr>
<th>Stage 3</th>
<th>Stage 2</th>
<th>Stage 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_3 )</td>
<td>( u(1) )</td>
<td>( V_2 )</td>
</tr>
<tr>
<td>18</td>
<td>A</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>B2</td>
<td>13</td>
</tr>
</tbody>
</table>

Fig. 15.4.

In constructing the table we start with \( x(4) = D \) at the extreme right of the page; in the appropriate column we note all possible routes from \( x(3) \) with corresponding costs. Now we go one stage “backwards” and note, in the middle columns, all possible routes from \( x(2) \) to \( x(3) \), together with the total costs corresponding with reaching the end of the journey, D. For example from B1 to D via C1 costs 13, whereas via C2 it would cost 15. Clearly, if it turns out that we should travel via B1, we should choose further to progress via C1 and not C2. It is clear to see which decisions give the least total costs per stage – these are underlined in the table, and correspond to \( V_1^0, V_2^0 \) and \( V_3^0 \). The other possible control strategies – in accordance with the Principle of Optimality – we can forget. Clearly the result of the table is that the route A - B2 - C1 - D is the optimal route, with the cost of the journey a minimum of 16 units.

In this simple example just given it would be feasible to calculate every possible route combination in order to find the optimal strategy; thus the example should be taken as an illustration of the method which could be employed in far more complicated situations.

Often, and especially in process control work, we are given a problem...
formulated in terms of continuous variables. These can, of course, be discretized and an exact solution sought by means of the discrete form of dynamic programming; this would be a particularly attractive approach for systems containing pure time delays. The method is illustrated in the next example.

Example 15.5.
Given a process
\[ \dot{x} = x + u, \]
with \( x(0) = x_0 \) and \( x(t_1) = x_1 \), find the control strategy which minimizes the cost functional
\[ J = \int_0^{t_1} (x^2 + a^2u^2)dt. \]

We divide the time interval from 0 to \( t_1 \) in \( N \) equal steps \( h \), such that \( Nh = t_1 \). Using the notation
\[ x(t)_{|t=kh} = x(k) \]
we may write a discrete model of the process as:
\[ \frac{x(k+1) - x(k)}{h} = -x(k) + u(k) \]
or
\[ x(k+1) = (1-h)x(k) + hu(k) \]
and the cost functional as
\[ J = \sum_{k=1}^{N} \{x^2(k) + a^2u^2(k)\}h. \]

Initial and final states are \( x(0) \) and \( x(N) \). For \( x(k) \) and \( u(k) \) we must now take a number of discrete values at every point \( k \), and proceed with the help of either a table, as in the previous example, or the recursive formula, to find the optimal control strategy.

Some of the advantages of dynamic programming which spring to mind are:
- constraints in state or control trajectories may be accounted for;
- very suitable for applications using a digital computer;
- an absolute (global) minimum rather than local extrema is found;
- since we usually work backwards from the final state, there is a kind of inbuilt feedback in the algorithm;
- from the tabular form of data storage it is easy to see the influence and source of any deviations from the desired strategy;
- points from graphs of experimental tests can be incorporated just as easily as analytical relationships.

One serious drawback of dynamic programming is the demand made on computer memory and calculation time as the problems increase in size and as more accuracy is required.
15.5. Linear optimal control systems

15.5.1. Optimal feedback laws

It is not surprising that if we apply any of the techniques described in the last sections, viz. calculus of variations, Pontryagin’s Maximum Principle or Bellman’s Dynamic Programming, to a linear process and a quadratic performance index, then we get exactly the same answer every time. What is surprising is that this, and only this, combination of linear process minimizing a quadratic performance index results in a linear optimal control law. We shall investigate this attractive result in some depth here; the problem may be stated as follows.

Given a linear system \( S_1(A, B, C) \) and a quadratic performance index:

\[
J = \int_{t_0}^{t_1} \left[ x^T Q x + u^T R u \right] dt
\]  
(15-27)

where \( Q \) is a real, symmetric and positive semi-definite matrix and \( R \) is a real, symmetric and positive definite matrix. Note that the quadratic performance index is not only mathematically convenient but has a certain physical significance. For example, it gives a measure of the energy dissipated in controlling a process, and it can be shown to minimize – in a certain sense – errors propagating from the non-linearity of the real (as opposed to the model) process.

We are required to control the system from some initial disturbance \( x(t_0) \neq 0 \) back (eventually) to \( x = 0 \), such that \( J \) is minimal. Note that \( x(t_1) \) does not have to be zero. The control strategy which ensures \( J \) is minimal we shall call the optimal control law, \( u^0(t) \).

To solve the stated problem we could apply the Maximum Principle. Using eq. (15-15) the Hamiltonian function is

\[
H = -x^T Q x - u^T R u + p^T (Ax + Bu).
\]  
(15-28)

The control \( u(t) \) is unconstrained so \( H \) has a maximum when

\[
\frac{\partial H}{\partial u} = -2Ru + B^T p = 0
\]  
(15-29)

and from eq. (15-16):

\[
p = 2Qx - A^T p
\]  
(15-30)

must also hold. It is possible to show from eqs. (15-28) to (15-30) that the optimal control law may be written as:

\[
u^0(t) = -Kx(t),
\]  
(15-31)

where

\[\dagger\] It is just as possible to write the result in the form of a feedforward control system, \( u^0(t) = K^*x_0 \). Feedback is not a result of the mathematics.

213
and $P$ is to be found from the matrix equation

$$p = A^T P + PA + Q - PBR^{-1}B^T P$$

(15-33)

with

$P(t_1) = 0$.  

(15-34)

Eq. (15-33) has the form of a so-called *matrix Riccati equation*. Under the given conditions this equation always has one unique solution, $P$, which is a symmetric and semi-positive definite matrix [7]. The linear optimal feedback law, eq. (15-31), deserves some further comment. It can be seen that application of eq. (15-31) to a real system would result in a controller with all setpoints maintaining constant values (remember that our model variables are always perturbation variables). Such a linear optimal control system is called a *regulator* [7]. The proportional control matrix $K$ of eq. (15-31) is in general composed of a number of time varying gain elements; if, however, we may assume that the matrices $A, B, C, Q$ and $R$ are time-invariant and furthermore that $t_1 + \infty$, our $K$ matrix becomes a constant gain matrix.

The conditions under which the matrix Riccati equation then has a unique solution are twofold; namely the system $S_1(A,B,C)$ must be controllable and it must be observable. To be precise, the controllability matrix

$$[B \mid AB \mid A^2 B \mid \ldots \mid A^{n-1} B]$$

and the observability matrix

$$[C^T \mid A^T C^T \mid (A^T)^2 C^T \mid \ldots \mid (A^T)^{n-1} C^T]$$

must both be of rank $n$ [7]. It may then be proved that $P$ is a constant matrix, being the unique semi-positive definite symmetric solution of the *algebraic matrix Riccati equation*:

$$A^T P + PA + Q - PBR^{-1}B^T P = 0.$$ 

(15-35)

Furthermore, it can also be shown that under the above conditions of controllability and observability the closed loop system of process plus optimal linear feedback law is asymptotically stable; this means that by employing eq. (15-31) we have not only ensured optimality but also stability.

**Example 15.6.**

Consider once again the system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$J = \int_0^\infty \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} x + u(1)u^2 dt.$$

The optimal controller is calculated as follows. From eq. (15-35)
\[
\begin{bmatrix}
0 & 0 \\
1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
P_{11} & P_{12} \\
P_{12} & P_{22} \\
\end{bmatrix}
\begin{bmatrix}
P_{11} & P_{12} \\
P_{12} & P_{22} \\
\end{bmatrix}
\begin{bmatrix}
0 & 1 \\
0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & 0 \\
\end{bmatrix}
- 
\begin{bmatrix}
P_{11} & P_{12} \\
P_{12} & P_{22} \\
\end{bmatrix}
\begin{bmatrix}
0 \\
1 \\
\end{bmatrix}
\begin{bmatrix}
P_{11} & P_{12} \\
P_{12} & P_{22} \\
\end{bmatrix}
= 0.
\]

Multiplying this expression out we find that
\[
1 - P_{12}^2 = 0 \\
P_{11} - P_{12}P_{22} = 0 \\
2P_{12} - P_{22}^2 = 0.
\]

Solving these three equations we find that
\[
P = \begin{bmatrix}
P_{11} & P_{12} \\
P_{12} & P_{22} \\
\end{bmatrix} = \begin{bmatrix}
\sqrt{2} & 1 \\
1 & \sqrt{2} \\
\end{bmatrix},
\]
so that the optimal controller gain matrix becomes (eq. (15-32)):
\[
K = [1][0 1] \begin{bmatrix}
\sqrt{2} & 1 \\
1 & \sqrt{2} \\
\end{bmatrix} = [1 \sqrt{2}]
\]
and the optimum controller therefore
\[
u = -x_1 - \sqrt{2}x_2
\]

An important result for the practical application of the optimal controller is the following. Suppose that the process is most accurately described by the inclusion of a stochastic disturbance source in the process model:
\[
\begin{align*}
\dot{x} &= A x + B u + w \\
y &= C x
\end{align*}
\]
(15-36)
The vector \(w\) represents uncorrelated white noise sources (Chapter 7) and consequently has a zero mean value. Cost criterion \(J\) is now a stochastic variable, and clearly we must take the expected value \(E\) of the performance index (see Appendices B and D)
\[
J = E \left[ \int_{t_0}^{t_1} \{x^T Q x + u^T R u\} \, dt \right].
\]
(15-37)
Now it so happens that if eq. (15-37) is optimized subject to eq. (15-36), then the result is eqs. (15-31) to (15-34), i.e. the optimal linear feedback law is exactly the same for a system with or without a white noise process disturbance source. This means that it is unimportant for the optimal controller if a deviation in the form of an initial condition (Chapter 12) \(x(t_0) \neq 0\) must be corrected or if the continuing effect of zero-mean process disturbances, \(w\), must be minimized. What is more, if \(w\) is also Gaussian (Chapter 7), then eq. (15-31) is not only the optimal linear feedback law, but also the optimal feedback law.
15.5.2. state estimation

We have already encountered the problem of reconstructing unobservable states from \( y \) in Chapter 13, where we dealt with the deterministic situation. The more practical situation is now treated, when both process and measurements are contaminated with noise; should this be the case, then the separation theorem [7] shows how to develop the optimal controller. The separation theorem states that

*given a linear system, quadratic performance index and process and measurement disturbances in the form of uncorrelated white noise, then the optimal linear control is a combination of the optimal linear feedback law — as given in eqs. (15-31) to (15-34) above — preceded by an optimal state estimator.*

An optimal state estimator provides us with an optimal approximation, \( \hat{x}(t) \), of the state \( x(t) \) from the measurements \( y(t) \).

![Stochastic process disturbances](diagram)

Fig. 15.5. Stochastic multivariable control.

In Fig. 15.5 the separation theorem has been applied to a stochastic multivariable system; the state estimator is the subject of what follows in this section.

216
It turns out that for the stochastic system just as for the deterministic counterpart, the ideal state estimator is the observer (Section 13.7, Chapter 13).

Fig. 15.6 shows the structure of the observer; the observer equation is

\[ \dot{\hat{x}} = A\hat{x} + Bu + G(y - C\hat{x}) \]  

and the estimation error can be given as

\[ e = x - \hat{x}. \]  

A measure of the quality of the estimates, \( \hat{x} \), are the numbers \( E\{e_i^2(t)\} \), \( i = 1, 2, \ldots, n \), i.e. the variance of \( e \). It so happens that each \( E\{e_i^2(t)\} \) is a minimum for one and the same value of the matrix \( G \) in eq. (15-38); clearly this value is the optimal setting for \( G \). For the configuration of Fig. 15.6 with the variance matrices of \( w, n \) and \( e \) denoted by (see Appendix D) \( W, V \) and \( U \) the optimal \( G \) is given by (assuming \( |V| \neq 0 \))

\[ G = UC^TV^{-1}, \]  

where \( U \) can be found from
Note that $U$ is a symmetric, positive definite matrix. Eq. (15-41) is again a so-called matrix Riccati equation, and it will always have a unique solution.

The optimal observer just described is known throughout the world as the Kalman filter, after its discoverer. This filter is, indeed, the optimal state estimator referred to in the separation theorem. It can be shown [7] that the Kalman filter is the best linear solution to the estimation problem; if $w$ and $n$ are Gaussian then it is also the optimal solution.

Obviously eqs. (15-33) and (15-41) are extremely similar. Indeed, the optimal estimator problem is called the dual of the optimal controller problem.

In general since the matrices $A$, $B$, $C$, $W$ and $V$ are time varying, it is not surprising that the feedback matrix $G$ is also, in general, time dependent. For practical purposes, however, we often assume that the system is time-invariant and look for a solution as $t \to \infty$. Under the conditions of controllability and observability (Section 15.5.1) $U$ is constant, and is the unique semi-positive definite solution of the algebraic matrix Riccati equation:

\[
AU + UA^T + W - UC^T V^{-1} CU = 0. \tag{15-42}
\]

Example 15.7.

A process is described by the model

\[
\begin{align*}
\dot{x} &= -x + u + w \\
y &= x + n.
\end{align*}
\]

$w$ and $n$ are uncorrelated Gaussian white noise sources. We wish to stability control the system so that

\[
J = E \left[ \int_0^\infty (x^2 + u^2) \, dt \right] = E \{x^2 + u^2\}
\]

is minimized, whatever the initial conditions $x(0)$ and in spite of the random process disturbance, $w$.

In this example we see that $A = -1$, $B = C = Q = R = 1$ and $W = \sigma_w$, $V = \sigma_n$. The optimal feedback control law can be found from eq. (15-35); we have that

\[
-2P + 1 - p^2 = 0, \quad P \text{ scalar}
\]

\[
\therefore P = -1 \pm \sqrt{2}.
\]

We take the positive root and find from eqs. (15-31) and (15-32) that

\[
u^0 = -\sqrt{2} - 1 \, x. \tag{15-43}
\]

The Kalman filter is calculated as follows. From eq. (15-42):

\[
-2U + \frac{\sigma_w^2}{\sigma_n^2} - \frac{U^2}{\sigma_n^2} = 0, \quad U \text{ scalar}
\]

\[
\therefore U = \sigma_n \left[ -1 \pm \sqrt{1 + \frac{\sigma_w^2}{\sigma_n^2}} \right].
\]

Taking once more the positive root we find, with the help of eq. (15-40) that
\[ G = -1 + \sqrt{1 + \frac{\sigma_w}{\sigma_n}} \]

and hence the Kalman filter becomes (eq. 15-38):

\[ \dot{\hat{x}} = -\hat{x} + u + G(y - \hat{x}) = -(1 + G)\dot{x} + Gy + u \]

\[ = -\sqrt{1 + \frac{\sigma_w}{\sigma_n}} \dot{x} + \left( -1 + \sqrt{1 + \frac{\sigma_w}{\sigma_n}} \right)y + u, \]

i.e. the Kalman filter in this case is a first order filter with a time constant \(1/\sqrt{1 + \sigma_w/\sigma_n}\).

The complete solution to the problem is thus an optimal controller whereby the state is estimated according to (15-44), the optimal control signal being constructed from the estimate according to (15-43), i.e.

\[ u^0 = -(\sqrt{2} - 1)\dot{x}. \]

For the sake of completeness we record that the optimal observer problem can also be solved \([7]\) when \(w\) and \(n\) are non-white, correlated and when \([V] = 0\). We must also mention the existence of the so-called discrete Kalman filter; more suitable, in some respects, than the continuous filter developed here when we are using a computer to control the process. This must usually be the case, since in general the matrix Riccati equations are solved by numerical means.

It is also possible to obtain an optimal state estimator from the theory of Wiener \([4]\). Such a Wiener filter gives exactly the same optimal estimate as the Kalman filter when used for stationary (Appendix B) signals. Van der Grinten \([6]\) has worked out a number of optimal Wiener filters, which are commendably presented in the form of a table of transfer functions. However, there are disadvantages:

- only applicable in practice for stationary signals
- infinite observation time necessary
- not immediately applicable to multivariable systems
- not suited to digital computation
- practical realization of the theoretical Wiener filter can be very complicated

and these disadvantages combine to make the Kalman filter the obvious choice in most applications.

15.6. references and bibliographical notes

In decreasing order of complexity and number of pages, the books by


and finally


make a good introduction to the subject. See also the references of Appendix C. Several chapters of the book by

also make interesting reading in the context of this chapter. The books by

VAN DER GRINTEN, P. M. E. M. and LENOIR, J. M. H. Statistische procesbeheer-
sing, Het Spectrum, Utrecht, 1970

and

VAN DER GRINTEN, P. M. E. M. Regeltechniek en automatisering in de proces-
industrie, Het Spectrum, Utrecht, 1970

both treat the subject admirable from the process control point of view, but unfortunately
are only available in Dutch.

As a standard text on linear systems control it is possible to recommend without any
hesitation

KWAKERNAAK, H. and SIVAN, R. Linear optimal control systems. John Wiley,
1972.

We have made no mention here of the Lyapunov approach to optimal control. For
this, see


The best modern book on dynamic programming is, for our purposes, by

BOUDAREL, R., DELMAS, J. and GUICHET, P. Dynamic programming and its
application to optimal control, Academic Press, 1971,

while Kalman filters are neatly discussed in

... if a computer could be programmed to play good chess, it could also be programmed to perform other intellectually difficult tasks...

David Levy

PROCESS CONTROL BY COMPUTER

16.1. introduction

Digital computers can be used in various ways to control a process. For our purposes there are four main possibilities:

- collection and processing measurements (data-logging) for use later for scientific or administrative purposes,
- to replace the conventional analogue computer in the control loop (so-called direct digital control, DDC),
- to optimize the economic profit of a process by suitable calculation and adjustment of the setpoints (supervisory control),
- to optimize the profit of a whole or part of a company, by coordinating the production of various processes and plants (coordination, see Chapter 14).

In this chapter we will consider only DDC, the principle of which is shown in Fig. 16.1.

Fig. 16.1. Conventional control and DDC.
Just as we had, in Fig. 8.1 (p. 84, Chapter 8), divided the conventional control loop into its main components, so we can show the general DDC control loop as in Fig. 16.2. The digital computer processes only digital information; thus any measurements from our process (which will usually be analogue signals) must first be discretized in amplitude (quantized) by means of an analogue-digital converter (ADC). Since ADC's are fairly expensive, it is usual to use just one ADC for many analogue measurement signals - these being serially presented to the ADC by means of a multiplexer. Thus the multiplexer discretizes the given analogue signal in the time - an important process which is called sampling.

Periodically the results of calculations performed in the digital computer will be sent to the digital-analogue converter (DAC) for dispatch via another multiplexer to a so-called hold element. The purpose of this element is to reconstruct the sampled signal, i.e. to make a continuous-in-the-time signal from the sampled data.

Due to the quantization in the ADC there is obviously an error introduced into the control loop which will be equal to the difference between the real value of the signal and the quantified value; this error is called quantization noise. The sampling effects chiefly the dynamic behaviour of the control loop through a deterioration in the controllability of the process. In general this deterioration will be more important from a control point of view than the loss of accuracy inherent in quantization. In Fig. 16.3 we have summarized the most important functions of the elements which were given in Fig. 16.2. The quantization has been neglected in Fig. 16.3, and we have symbolized the process of sampling by the use of a switch (sampler).

A system where sampling occurs is frequently referred to as a sampled data system. In the following section the sampling and reconstruction of signals will be discussed. The z-transformation is then introduced to facilitate the calculation of sampled data systems. Section 16.4 deals with computer software, while the final section introduces the state-space description of sampled data systems.

16.2. sampling and reconstruction of signals

16.2.1. sampling and samplers

On inspection of Fig. 16.3 it is clear that we can make a distinction between the discrete part (digital processing) and the continuous part (reconstruction, correction, process and measurements) of a sampled data system. This is shown in Fig. 16.4. Clearly, a sampled signal can be applied to either a continuous or a discrete element, and it turns out that the effect of the intermittent signal is not the same in both cases.

The sampler itself can be modelled as a switch S (Fig. 16.5) which at a certain time is closed and lets a signal x(t) through for a short time τ before opening. The sampler S thus transforms the signal x(t) to the sampled signal x*(t). We shall assume henceforth that S closes periodically at the times 0, T_b, 2T_b, ... etc., and T_b will be thus the sample time. The sampling
frequency, \( f_b \), is the reciprocal of the sample time (thus in Hz or cycles/sec), while the frequency expressed in radians per second, the angular frequency, we will denote by \( \omega_b = 2\pi f_b \).

16.2.2. sampling and continuous elements

The continuous element is described, we assume, by means of a differential equation. If \( \tau \ll T_b \), then the sample duration \( \tau \) is no longer important although the energy content is. We may thus regard the sampled signal \( x^*(t) \) as a series of impulses [4]:

\[
x^*(t) = x(t) \sum_{n=0}^{\infty} \delta(t - nT_b).
\]
Fig. 16.4. The discrete and continuous parts of a sampled data control loop.

Fig. 16.5. The sampling process.
It can be shown [7] that the infinite series of eq. (16-1) can be written as

$$\sum_{n=0}^{\infty} \delta(t - nT_b) = \sum_{n=-\infty}^{\infty} \frac{1}{T_b} e^{jn\omega_b t}$$

and so

$$x^*(t) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} x(t)e^{jn\omega_b t}.$$  

By Laplace transforming this expression and substituting $s = j\omega$ it is possible to arrive at the following expression for $x^*(t)$ in the frequency domain:

$$X^*(j\omega) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_b).$$

The frequency spectra of the continuous and sampled signals can be compared using eq. (16-4) – see Fig. 16.7. We see, for example, that sampling has had the effect of creating an infinite number of complementary frequency bands, each $\pi\omega_b$ radians/second apart, and each identical to the spectrum of $x(t)$. If the continuous signal $x(t)$ contains frequencies higher than $\omega_b/2$ then overlapping will occur in the spectrum of $x^*(t)$. It is then no longer possible to reconstruct the original signal $x(t)$ from the sampled signal $x^*(t)$. This fact is stated in Shannon's theorem: when there are no frequencies higher than $\omega_c$ in a signal, then the signal can be completely characterised by measuring the signal at intervals of $T_c/2$.

We can also write eq. (16-1) in the equivalent form:

$$x^*(t) = \sum_{n=0}^{\infty} x(nT_b) \delta(t - nT_b).$$

If this is now Laplace transformed then the following expression results, to which we will refer when we consider the $z$-transformation in Section 16.3:

$$x^*(s) = \sum_{n=0}^{\infty} x(nT_b) e^{-nT_b s}.$$  

### 16.2.3. Sampling and discrete elements

The discrete element of Fig. 16.8. is assumed to be described by a difference equation. When $\tau \ll T_b$ then the pulse width can be neglected in comparison with its height. The signal $x^*(t)$ can thus be thought of as a series of
In practice the discrete element will need a certain time to perform the calculations necessary before it can produce an output. Since this time has been included in the pulse width, which is then neglected, it is possible to model the output of the discrete element by a number series

\[ y^*(t) = \{y(nT_b)\}. \tag{16-8} \]

**16.2.4. reconstruction**

The reconstruction of a continuous signal from a set of sampled data is achieved by using a continuous element called a hold-element; we know from Section 16.2.1 that \( x^*(t) \) is therefore best described as a series of impulses. Furthermore, in order to reconstruct \( x(t) \) for \( kT_b \leq t < (k + 1)T_b \) we develop \( x(t) \) using a Taylor series expansion (p. 17, Chapter 2):
\[ x(t) = x(kT_b) + \frac{kT_b}{1!} \dot{x}(kT_b) + \frac{(kT_b)^2}{2!} \ddot{x}(kT_b) + \ldots \quad (16-9) \]

Obviously, the more derivative terms we include, for \( t = kT_b \), the more accurate our reconstruction of \( x(t) \) for \( kT_b \leq t < (k+1)T_b \) will be. We therefore say that the reconstruction of \( x(t) \) is by means of an \( n^{\text{th}} \) order hold element when we take into account terms of eq. \((16-9)\) up to and including the \( n \)-th derivative.

Here it is only possible to mention the most common hold circuit, the zero-order hold. This element can be described by

\[ x_h(t) = x(kT_b), \quad kT_b \leq t < (k+1)T_b \quad (16-10) \]

and its operation is shown in Fig. 16.9. We see that the output of the zero-order hold, \( H(s) \), remains constant between sampling periods. The transfer function \( H(s) \) may be found as the response to a unit impulse at \( t = 0 \) (p. 59, Chapter 5) — the response is

\[ x_h(t) = U(t) - U(t - T_b), \quad (16-11) \]

where \( U(t) \) is the unit step function. From eq. \((16-11)\) it follows that

\[ H(s) = L\{x_h(t)\} = \frac{1}{s} e^{-sT_b}. \quad (16-12) \]

16.3. the z-transformation

Just as the Laplace transformation is of such use for continuous systems, so the z-transformation makes a simple description of sampled data systems possible.

16.3.1. impulse trains and continuous elements

It may be recalled that eq. \((16-6)\) gave the Laplace transform, \( \tilde{x}^*(s) \), of a train of pulses. Defining a variable, \( z \), as

\[ z = e^{sT_b} \quad (16-13) \]

we have the so-called z-transform of \( \tilde{x}^*(s) \):

\[ \tilde{x}(z) = Z\{\tilde{x}^*(s)\} = \sum_{n=0}^{\infty} x(nT_b)z^{-n}. \quad (16-14) \]

Sometimes the following notation is used:

\[ \tilde{x}(z) = Z\{x^*(t)\}. \quad (16-15) \]

Comparing these last two equations it is obvious that we may proceed to the z-domain directly from the time domain, or via the s-domain. It is important to remember that the z-transformation applies to sampled signals, even although this is not always stated; in other words \( Z\{x(t)\} \) is identical to \( Z\{\tilde{x}^*(s)\} \).
Example 16.1.

What is the z-transform of the pulse train resulting from sampling the unit step function?

\[
x(z) = \sum_{n=0}^{\infty} x(nT_b)z^{-n} = \sum_{n=0}^{\infty} z^{-n} = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}.
\]

By making the substitution \( z = e^{sT_b} \) the complex s-plane (p. 117, Chapter 11) is mapped into a complex z-plane:

\[
z = e^{sT_b} = e^{sT_b + j2\pi n}, \text{ n integer}
\]

\[
e^{sT_b + jn\omega_b T_b} = e^{(s+jn\omega_b)T_b}.
\]

(16-17)

Substituting \( s = j\omega \) in this result gives

\[
z = e^{j(\omega + n\omega_b)T_b}
\]

(16-18)

so that

\[
|z| = 1 \quad \text{and} \quad \arg(z) = (\omega + n\omega_b)T_b
\]

(16-19)

and thus the imaginary axis of the s-plane is mapped into the z-plane as a unitary radius circle about the origin (see Fig. 16-10). The half of the s-plane to the left of its imaginary axis appears inside this circle, while the right-half plane falls outside the circle.

![Diagram](image)

Fig. 16.9. Zero-order hold element.

Suppose now that a sampled signal is applied to a continuous element, Fig. 16.11.

In terms of Laplace transforms, \( \bar{y}(s) = G(s)\bar{x}^*(s) \); usually it proves impossible to find a rational expression in s for \( \bar{y}(s) \). If, however, we supply a fictitious
When we are concerned with discrete elements the calculations are perform-
ed on series of numbers instead of impulses. However, the z-transformation is still applicable [5], so long as the transformation of a series of numbers \( \{x(kT_b)\} \) is defined by

\[
\tilde{x}(z) = z \{x(kT_b)\} = \sum_{k=0}^{\infty} x(kT_b)z^{-k}.
\]  

(16-22)

**Example 16.2.**

What is the z-transform of the number series \( \{g(nT_b)\} = \left(\frac{1}{2}\right)^n \)?

\[
Z\{g(nT_b)\} = \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^n z^{-n} = \frac{l}{l - \frac{1}{2}z^{-1}} = \frac{z}{z - \frac{1}{2}}
\]  

Suppose that the number series \( \{x(kT_b)\} \) is the input of a discrete element – Fig. 16.8. This discrete element is described – we assume – by a difference equation which relates input to output. We would expect that if the difference equation was transformed into the z-domain there would exist a transfer function \( G(z) \) relating the z-transforms of input and output, i.e.

\[
\tilde{y}(z) = G(z)\tilde{x}(z).
\]  

(16-23)

Fig. 16.12. z-domain representation of a number series applied to a discrete element.

In the following section, 16.4, the method of obtaining \( G(z) \) will be discussed.

It can be concluded from the above that the z-transformation is applicable to systems where continuous and/or discrete elements are present. With discrete elements the z-transformation gives the complete behaviour of that element; with continuous elements it gives only the behaviour at the instant of sampling, no information is retained concerning the behaviour between sampling times. Furthermore, it is apparent from Fig. 16.10 that a sampled data system \( G(z) \) is stable when all the poles of \( G(z) \) lie within the unit circle in the z-plane.

**Example 16.3.**

Is the system \( G(z) = \frac{z}{z - e^{-atT_b}} \) stable?

The system \( G(z) \) clearly has one pole at \( z = e^{-atT_b} \). Stability is ensured when \( |z| < 1 \), and hence the condition for stability is that the constant a be greater than zero (i.e. positive).

Note that \( G(z) \) is the z-transform of \( e^{-at} \); this function clearly only remains bounded when \( a > 0 \).

Just as with Laplace transforms, use is made of an inverse z-transformation to go from the z-domain back into the time domain. For this purpose a table of z-transforms is

**Table 16.4.**

<table>
<thead>
<tr>
<th>( f(t) )</th>
<th>( \hat{f}(z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e^{at} )</td>
<td>( \frac{1}{z - e^{-atT_b}} )</td>
</tr>
<tr>
<td>( \sin \left(\frac{\omega}{2}\right) )</td>
<td>( \frac{\omega}{z^2 + \frac{\omega^2}{4}} )</td>
</tr>
</tbody>
</table>

The process of obtaining the z-transform of a discrete element is shown in Fig. 16.14.

In terms of the use of inverse z-transforms, the following example can be formulated.

**Example 16.4.**

Find the time-domain function \( x(t) \) if \( \hat{x}(z) = \frac{z}{z - e^{-atT_b}} \).

Or
of z-transforms [4] may be consulted, a part of which is shown in Table 16.1.

Table 16.1. Some important z-transforms.

<table>
<thead>
<tr>
<th>f(t)</th>
<th>F(s) or F(z)</th>
<th>f(z) or ( f(z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta(t) )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>U(t)</td>
<td>( \frac{1}{s} )</td>
<td>( \frac{z}{z-1} )</td>
</tr>
<tr>
<td>( t )</td>
<td>( \frac{1}{s^2} )</td>
<td>( \frac{T_b z}{(z-1)^2} )</td>
</tr>
<tr>
<td>( e^{-at} )</td>
<td>( \frac{1}{s+a} )</td>
<td>( \frac{z}{z-e^{-aT_b}} )</td>
</tr>
<tr>
<td>( \sin(\omega t) )</td>
<td>( \frac{\omega}{s^2 + \omega^2} )</td>
<td>( \frac{z \sin(\omega T_b)}{z^2 - 2z \cos(\omega T_b) + 1} )</td>
</tr>
<tr>
<td>( \frac{t}{T_b} )</td>
<td>( \frac{1}{s - \frac{1}{T_b} \ln \alpha} )</td>
<td>( \frac{z}{z - \alpha} )</td>
</tr>
</tbody>
</table>

The most important properties of the z-transformation are its linearity and time-translation:

- **Linearity:** \( Z\{af_1(t) + bf_2(t)\} = at(z) + bT(z) \)  
- **Translation in Time:** \( Z\{f(t - mT_b)\} = z^{-m} \tilde{f}(z) \)  

Although in Table 16.1 and eqs. (16-24) to (16-26) the notation \( f(t) \) is used, the same expressions hold for a number series \( \{f(nT_b)\} \). If the required z-transform \( \tilde{f}(z) \) does not appear in a table of transforms then its inverse may be found by a partial fraction expansion in the form of \( \tilde{f}(z)/z \). The following example illustrates this technique.

**Example 16.4.**

Find the inverse transform of

\[
\tilde{f}(z) = \frac{1}{2} \frac{z}{(z-1)(z - \frac{1}{2})}
\]

In terms of partial fractions,

\[
\tilde{f}(z) = \frac{1}{2} \frac{1}{(z-1)(z - \frac{1}{2})} = \frac{1}{z - 1} - \frac{1}{z - \frac{1}{2}}
\]

or

\[
\tilde{f}(z) = \frac{z}{z - 1} - \frac{z}{z - \frac{1}{2}}
\]
From Table 16.1 it follows that

\[ f(nT_b) = 1 - \left(\frac{1}{2}\right)^n, \quad n = 0, 1, \ldots \]

### 16.4. control algorithms

![Fig. 16.13. A DDC control loop.](image)

The above figure shows the general DDC control loop arrangement, with D the digital controller, H(s) the hold element and G(s) the process.

In the case of a multivariable process G(s) will represent the transfer function relating one output to one input of the process – this having been split into a number of non-interacting s.i.s.o. systems (Chapter 13). The digital controller D then replaces the conventional P, PI or PID controller in the s.i.s.o. loop.

We have included in the figure a sampler before the setpoint; this is to indicate that in the computer the series \( \{r(nT_b)\} \) is compared with the series \( \{e(nT_b)\} \). The purpose of this section is to develop a digital controller, D, so that the desired process behaviour is obtained. This controller, D, is in fact a **control algorithm** that is executed in the computer by means of a special program. Its dynamic behaviour may be modelled by a linear **difference equation** with constant coefficients; if the input to D is \( \{e(kT_b)\} \) and the output \( \{y(kT_b)\} \), then:

\[

b_0 y(kT_b) + b_1 y(kT_b - T_b) + \ldots + b_m (kT_b - mT_b) = \\
= a_0 e(kT_b) + a_1 e(kT_b - T_b) + \ldots + a_n e(kT_b - nT_b). \quad (16-27)

\]

Notice the similarity between this equation and a linear **differential equation** with constant coefficients. The continuous time \( t \) in the differential equation is equivalent to the discrete intervals of time \( kT_b \), while the \( q \)-th derivative of \( y(t) \) appears in eq. (16-27) as \( y(kT_b) \) translated by an amount \( qT_b \) in time; that is, \( \frac{d^q y}{dt^q} \) is equivalent to \( y(kT_b - qT_b) \). In analogy with differential equation theory, we speak of an \( m \)-th order difference equation if the greatest time translation present in the output is \( mT_b \).
Just as a linear differential equation may be solved using Laplace transforms, so can difference equations be solved by applying the z-transformation. As we have seen in the previous section, it is also possible to obtain transfer functions in the z-domain. Applying, then, the z-transformation to eq. (16-27), and making use of eqs. (16-24) and (16-25) we find that

\[ b_0 \tilde{y}(z) + b_1 z^{-1} \tilde{y}(z) + \ldots + b_m z^{-m} \tilde{y}(z) = \] \[ = a_0 \tilde{e}(z) + a_1 z^{-1} \tilde{e}(z) + \ldots + a_n z^{-n} \tilde{e}(z) \] \[ (16-28) \]

or

\[ (b_0 + b_1 z^{-1} + \ldots + b_m z^{-m}) \tilde{y}(z) = (a_0 + a_1 z^{-1} + \ldots + a_n z^{-n}) \tilde{e}(z) \] \[ (16-29) \]

From this last expression it is obvious that the transfer function, \( D(z) \), of the digital controller is given by

\[ D(z) = \frac{\tilde{y}(z)}{\tilde{e}(z)} = \frac{a_0 + a_1 z^{-1} + \ldots + a_n z^{-n}}{b_0 + b_1 z^{-1} + \ldots + b_m z^{-m}} \] \[ (16-30) \]

Returning now to Fig. 16.13 we find that the behaviour of the hold and process elements can be described by

\[ c(s) = H(s)G(s)y^*(s) \]

\[ c^*(s) = \{H(s)G(s)*\} * Y^*(s) \] \[ (16-31) \]

\[ (16-32) \]

Using the notation

\[ HG(z) = Z\{H(s)G(s)\} \]

we can write the z-transformation of eq. (16-32) as

\[ \tilde{c}(z) = HG(z)\tilde{y}(z) \] \[ (16-33) \]

The transfer function between \( \tilde{r}(z) \) and \( \tilde{c}(z) \) is now easily calculated. Since

\[ \tilde{c}(z) = \tilde{r}(z) - \tilde{e}(z) \] \[ (16-34) \]

and

\[ \tilde{y}(z) = D(z)\tilde{e}(z) \] \[ (16-35) \]

\[ \tilde{c}(z) = \frac{HG(z)D(z)}{1 + HG(z)D(z)} \] \[ (16-36) \]

The various methods for continuous system controller design can also usually be applied to develop suitable digital controllers for particular applications. For example, the pole trajectory method [6] or the techniques for pole assignment (Chapter 13) can be applied to the z-plane; optimal digital controllers are also feasible (Chapter 15).

An alternative approach is prompted by the observation that most process
control is achieved in practice by employing P, PI or PID controllers; it seems reasonable to expect that a control algorithm based upon the digital equivalent of an analogue controller would meet with success. By setting various program parameters to their “best” values the desired response of the process can then be tuned, similarly to the setting of proportional, integral or derivative action in conventional controllers. To illustrate what we mean, let us consider the digital equivalent of the PID analogue controller (p. 25, Chapter 9):

\[
y(t) = k[e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \dot{e}(t)].
\]

(16-37)

The integral may be approximated by a summation, and the derivative by a finite difference approximation. Using the notation \( x(nT_b) = x_n \) eq. (16-37) may be written as

\[
y_n = k[e_n + \frac{T_b}{\tau_i} \sum_{k=0}^{n} e_k + \tau_d (e_n - e_{n-1})].
\]

(16-38)

Substituting \( k_p = k; k_i = \frac{kT_b}{\tau_i} \) and \( k_d = \frac{k\tau_d}{T_b} \) gives

\[
y_n = k_p e_n + k_i \sum_{k=0}^{n} e_k + k_d (e_n - e_{n-1}).
\]

(16-39)

Moreover, in view of this last expression,

\[
y_{n-1} = k_p e_{n-1} + k_i \sum_{k=0}^{n} e_k + k_d (e_{n-1} - e_{n-2})
\]

(16-40)

must also hold. Subtracting eq. (16-40) from eq. (16-39) results in the so-called position algorithm:

\[
y_n = y_{n-1} + k_p (e_n - e_{n-1}) + k_i e_n + k_d (e_n - 2e_{n-1} + e_{n-2})
\]

\[= y_{n-1} + (k_p + k_i + k_d) e_n - (k_p + 2k_d) e_{n-1} + k_d e_{n-2}.\]

(16-41)

In practice it is usual to employ a velocity algorithm instead of a position algorithm; the velocity algorithm gives a measure for the difference between the old and the new position of the control (or correction) organ. For our digital PID controller, the velocity algorithm would be

\[
v_n = (k_p + k_i + k_d) e_n - (k_p + 2k_d) e_{n-1} + k_d e_{n-2}.
\]

(16-42)

From eq. (16-41) we can write the transfer function of the position algorithm as

\[
\tilde{Y}(z) = \frac{(k_p + k_i + k_d)z^2 - (k_p + 2k_d)z + k_d}{z(z - 1)} = D(z),
\]

(16-43)
while for the velocity algorithm we have, from eq. (16-42)
\[
\frac{y(z)}{e(z)} = \frac{(k_p + k_i + k_d)z^2 - (k_p + 2k_d)z + k_d}{z^2} = D(z).
\] 
(16-44)

**Example 16.5.**

Using eq. (16-44) derive the velocity algorithm for a digital PI controller.

With \( k_d = 0 \) in eq. (16-44)
\[
D(z) = \frac{(k_p + k_i)z^2 - k_p}{z^2} = (k_p + k_i) - k_p z^{-1}.
\]

The velocity algorithm is thus
\[
v_n = (k_p + k_i)e_n - k_p e_{n-1}.
\]

Since in general it is found the differential action of the digital PID controller is susceptible to noise — the quantification noise of the ADC for example — it is usual to use from four to six samples in place of the two required theoretically to mechanize eq. (16-42). Note also that various rules of thumb (for example, that of Ziegler Nichols, p. 102, Chapter 9) used frequently to optimize analogue controller performance can similarly be applied — after suitable modification — to set digital controllers.

The sampling time, \( T_b \), determines to a large extent the dynamic character of the control loop. A good choice of this parameter is therefore of vital importance. Suppose, for example, we have the simplest case of a process having one predominant time constant \( \tau \) and a dead time \( T_d \). The process transfer function is therefore
\[
G(s) = \frac{a e^{-sT_d}}{sT + 1}.
\]
(16-45)

In terms of its frequency characteristics (p. 65, Chapter 6) the process magnitude ratio from eq. (16-45) is
\[
|G(j\omega)| = \frac{a}{\sqrt{1 + \omega^2 \tau^2}}
\]
(16-46)
while the phase angle is given by
\[
\arg [G(j\omega)] = -\omega T_d - \arctan (\omega \tau).
\]
(16-47)

Suppose now that for some given frequency range \( \omega > \omega_{c1} \) the amplitude attenuation exceeds, by some factor \( a \), the amplitude attenuation at \( \omega = 0 \). The relationship between \( \omega_{c1} \) and \( a \) is
\[
|G(j\omega)| = \frac{a}{\sqrt{1 + \omega_{c1}^2 \tau^2}} \approx \frac{a}{\omega_{c1} \tau} = \frac{a}{a^*}.
\]
(16-48)

Hence
\[
\omega_{c1} = \frac{a}{\tau}
\]
(16-49)

If the factor \( a \) is large enough then any frequencies higher than \( \omega_{c1} \) (i.e.
\( \omega > \omega_{c_1} \) will have no noticeable effect on the dynamic behaviour of the process. Hence the sampling frequency must be high enough to retain all information present in the band of frequencies between d.c and \( \omega_{c_1} \) (i.e. \( 0 \leq \omega \leq \omega_{c_1} \)). According to Shannon’s Theory (Section 16.2.2)

\[
\omega_b > 2\omega_{c_1} = \frac{2a}{T}
\]

from eq. (16-49). Obviously, since \( \omega_b = \frac{2\pi}{T_b} \) it follows that

\[
T_b < \frac{\pi T}{a}. \quad (16-51)
\]

The closed loop stability must now be examined. Since the phase lag introduced by the process (eq. (16-47) must always be smaller than \( \pi \) radians (180°) — otherwise the purpose of negative feedback is defeated (p. 4, Chapter 1) — and since \( \arctan (\omega T) \) is always smaller than \( \frac{\pi}{4} \) radians, the maximum available frequency \( \omega_{c_2} \) for stability is, from eq. (16-47)

\[
\omega_{c_2} = \frac{3\pi}{4T_d}. \quad (16-52)
\]

Applying once more Shannon’s Theory, we find another condition for the sampling frequency is

\[
\omega_b > 2\omega_{c_2} = \frac{3\pi}{4T_d} \quad (16-53)
\]

or

\[
T_b < \frac{4T_d}{3}. \quad (16-54)
\]

Using eqs. (16-51) and (16-54) an intelligent choice can now be made for \( T_b \).

16.5. the state space description of sampled data systems

Once again it is important to distinguish between sampled signals applied to continuous or to discrete elements; we will find that while the latter situation can be completely described by the discrete state space representation, the sampled continuous element is described only at the times of sampling.

16.5.1. sampled signals applied to discrete elements

A discrete, linear element may be modelled by the vector difference equations:

\[
x(k + 1) = A x(k) + b u(k) \\
y(k) = C x(k).
\]

By writing eq. (16-55) down for \( x(0), x(1), \ldots, x(N) \) one after the other, and substituting for all the intermediate, \( x(1), x(2), \ldots, x(N - 1) \) values we find eventually that
\[ x(N) = A^N x(0) + \sum_{i=0}^{N-1} A^{N-1-i} B u(i) \]  
(16-56)

and

\[ y(N) = C x(N) = C A^N x(0) + \sum_{i=0}^{N-1} C A^{N-1-i} B u(i). \]  
(16-57)

### 16.5.2. continuous elements preceded by a zero-order hold circuit

The continuous element may be described, as we saw in Chapter 13, by

\[ \dot{x} = Ax + Bu \]  
(16-58)

which has the solution

\[ x(t) = e^{At} x(0) + \int_0^t e^{A(t-\tau)} B u(\tau) d\tau. \]  
(16-59)

Between time zero and \( t = T_b \), \( u(t) = u(0) \) and for \( T_b < t \leq 2T_b \), \( u(t) = u(T_b) \). Hence for the instants \( t = T_b, 2T_b \) at which samples are taken we have from eq. (16-59)

\[ x(T_b) = e^{A T_b} x(0) + \int_0^{T_b} e^{A(T_b-\tau)} B d\tau u(0) \]  
(16-60)

\[ x(2T_b) = e^{A T_b} x(T_b) + \int_0^{T_b} e^{A(2T_b-\tau)} B d\tau u(T_b). \]  
(16-61)

Calling \( \tau - T_b = \tau' \) it follows that

\[ x(2T_b) = e^{A T_b} x(T_b) + \int_0^{T_b} e^{A(T_b-\tau')} B d\tau' u(T_b). \]  
(16-62)

Comparing this last equation with eq. (16-60) it can be seen that in general the following model holds

\[ x(kT_b + T_b) = A^* x(kT_b) + B^* u(kT_b), \]  
(16-63)

where

\[ A^* = e^{A T_b} \]

and

\[ B^* = \int_0^{T_b} e^{A(T_b-\tau)} B d\tau. \]  
(16-64)

The general expression, eq. (16-63), clearly is completely analogous to eq. (16-55) for discrete elements. Thus the state space representation of the sampling
instants of a continuous element preceded by a zero-order hold element is exactly equivalent to the state space description of a discrete element.

We conclude this chapter with two examples.

Example 16.6.

Given the continuous process

\[
\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u
\]

preceded by a zero-order hold element, what is a state space model which represents the system at the sampling times?

Since (Chapter 12) \( e^{AT} = L^{-1} \left( (sI - A)^{-1} \right) \) then

\[
e^{AT} = L^{-1} \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} \\ 0 & \frac{1}{s} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.
\]

Thus \( A^* = e^{AT}b = \begin{bmatrix} 1 & T_b \\ 0 & 1 \end{bmatrix} \) and also from eq. (16-64) we find that

\[
B^* = \int_0^T \begin{bmatrix} 1 & T_b \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} d_T = \begin{bmatrix} \frac{1}{2}T_b \\ T_b \end{bmatrix}.
\]

The state space model follows on substituting this \( A^* \) and \( B^* \) into eq. (16-63).

In a similar fashion it is possible to express continuous processes which include pure time delays and from part of a sampled data system in a state space model.

What is the state space description, at the sample times, of a process having the transfer function:

\[
G(s) = \frac{-sT_d}{s + a}
\]

and preceded by a zero-order hold element?

We can write the transfer function of the process in the equivalent state-space form

\[
\dot{x}(t) = ax(t) + u(t - T_d).
\]

Using eq. (16-64) it follows that

\[
A^* = e^{-aT_b} \quad \text{and} \quad B^* = \int_0^{T_b} e^{-a(T_b - \tau)} = \frac{1}{a} (I - e^{-aT_b}).
\]

The state space description is therefore

\[
x(kT_b + T_p) = e^{-aT_b} x(kT_b) + \frac{1}{a} (I - e^{-aT_b}) u(kT_b - T_d).
\]

16.6. references and bibliographical notes

For readers understanding Dutch an excellent book is

Those readers who cannot should consult the equivalent book in English by


The book by


has frequently been recommended elsewhere in this book. It also contains a chapter on sampled data systems.

Two good reference books concerning the topics of this chapter are


and the more recent


Extra mathematical help is provided in the excellent books by


and

Such things as bee most vaporous
do most dispose us to sleepe.
Anon.

THE DYNAMICS AND CONTROL OF DISTILLATION COLUMNS

17.1. introduction

This, and the following chapter differ from their predecessors in so far as the study of distillation or chemical reaction may be said to be a specialization. We remain, however, committed to presenting only elementary aspects of these complicated but very important topics. As far as distillation dynamics and control is concerned, whole books [76], [77], [83] have been published which will guide the reader deeper into the subject; the existing literature is — qua quantity at any rate — formidable and we have been forced to abandon our previous habit of providing detailed bibliographical notes, in favour of a short guide (Section 17.9) to the literature published since 1970. The frequently appearing reviews, state-of-the-art and surveys ([52],[53],[63]) are also useful further reading.

It has been necessary to assume that the reader has some chemical engineering background and is therefore at home with the important physical and technological aspects of the process of distillation and distillation columns. Should this be lacking then there are again many textbooks ([67],[74],[76] or [77]) to explain the intricacies of McCabe-Thiele diagrams, plate construction etc. etc.; for a short refresher consult (pp. 187-192,[74]).

The author of this last-mentioned reference has also crystallised the central problem in the control of distillation processes: “In reality, the primary design problem is that of determining the structure of the control system rather than ... the best adjustment of control system parameters. This stands in strong contrast to the main approaches to problems in control theory.” It seems quite possible that the analysis of system structure (Chapter 14) will finally provide an answer to this problem of control system structure; see also Section 17.8.

From the very beginning we must be careful to distinguish between binary and multicomponent distillation; most theoretical studies are based upon a binary distillation, and agreement between theory and practice is often good [63]. Multicomponent distillation, however, although more important and frequently encountered in industry, has received relatively little attention. For studies concerning column control systems, the usual approximation to a binary distillation is not only necessary for the analysis but often produces good results.

A typical continuous distillation tower and associated peripherals are sketched in Fig. 17.1.

Actually, the tower shown is plate-column as apart from a packed tower, the latter being filled with a “packing” (special rings, for example) and also sometimes being used to distill mixtures.

The reboiler (tray or plate 0) uses frequently steam for vaporization and can
be of either the thermosyphon or drum type. Note that if the column feed is pre-heated using the surplus heat from the bottoms, there is a chance of positive feedback causing instabilities. The feed can, of course, be supplied to the column at various temperatures, although the most usual is the "bubble point". In theory, the place of the feedplate (where the feed enters the column) can also be altered; some control schemes using this possibility have been proposed. Our simple sketch shows only two product streams (tops and bottoms), but a further complication is the presence, in many industrial columns, of sidestreams. The condenser (tray \( N + 1 \)) is usually a large tubular heat exchanger, cooled by water (sometimes air), and is often very sensitive to ambient temperature fluctuations. The accumulator, although not so desirable from some points of view of control [70],[74], allows the reflux ratio to be altered. This is necessary in order to permit the composition of the top product (distillate) to be controlled (the greater the reflux ratio, the higher the distillate purity).

There is an enormous assortment of models in the literature. The choice of a particular type of model is dependent upon the application; often control studies require only a very simple stationary (instantaneous) model while a
nonlinear, multicomponent distributed parameter model might prove necessary for meaningful dynamic optimization of tray performance. We shall limit ourselves here to simple models; the reader is referred to Section 17.9 for state space and discrete time representations (under “modern control theory” and “computer control” respectively). Notwithstanding, four different (differential-difference, distributed parameter, first- and second-order lumped parameter, and stationary) types of model may be found in Sections 17.2 to 17.5.

Some assumptions are always made if a dynamic distillation column model is to be at all useable. Howard [63] gives three of the most usual:

i) No vapour accumulation (“hold up”) between trays;

ii) The compositions of the liquid leaving the tray, and that liquid on the tray, are the same;

iii) An instantaneous (algebraic) relationship exists between the concentration of a component in the vapour phase and in the liquid leaving every tray.

Even with these three assumptions, a complete dynamic model of column, accumulator and condenser can be impressively large without further assumptions: energy and mass balances give [65]:

\[(3C + 7)(N - 1) + 8C + 21\]

variables (C is the number of components and N is the number of trays in the column) and

\[(3C + 5)(N - 1) + 7C + 11\]

independent equations. The only tractable method of analysis then is computer simulation, although often much time must be spent devising fast and efficient simulation techniques.

There has also been a great deal published concerning the control of distillation columns, although it would be wrong to give the impression that it is very much more at present than an art — certainly outside academic circles, anyway. Usually the purpose of control is to remove undesirable disturbances in the feed flow or composition; we shall consider this “classical control” in Section 17.6. Classical control methods, which can be split into material balance control or composition [70], control of top and bottom products, use principally feedback. More complicated control schemes are also in abundance; feed-forward control [39] for example, or parallel, cascade, non-interacting, ratio and override control schemes [65]. Special nonlinear controllers have also been developed [75] to compensate the inherent nonlinearities in the distillation process.

If a process control computer is available, then far more sophisticated control algorithms are feasible. Even when only an off-line facility exists, there is a possibility of running a modal analysis [19],[74] or decoupling program to reduce the control problem to that of a number of s.i.s.o. non-interacting loops [27] — recall Chapter 13. Although advanced methods of optimal control [22],[24],[28] have been widely tested on small-scale and pilot plants, there is little to suggest that on-line computers are used in industry for more than DDC or
supervisory (Chapter 16) control. Relevant literature is presented in Section 17.9. It is interesting to note that some of the latest research into distillation column control is concerned with periodic control (Ch. 1, p. 6) [64], where one tries, using an oscillating mode of operation, to achieve higher profit margins than can be achieved using stationary, conventional operation. The presence of a computer also opens up possibilities for applying modern identification and parameter estimation techniques to real distillation columns [60], [61]; adaptive control [33], [37] is a logical further step. Unfortunately we cannot delve into these and other ideas here; our discussion will not extend beyond the treatment of feedforward control in Section 17.7.

In concluding this introduction the reader is warned of the possibility of instabilities [44] and non-minimum phase response [15], [80] occurring in some distillation units.

17.2 model of an ideal binary distillation plate

In order to model the unit shown in Fig. 17.1 we begin here by considering plate $n$ of the distillation column. Only mass and component balances are used; more complicated models making use of energy balances are readily to be found in the literature [63].

17.2.1 overall mass balance round plate $n$

Fig. 17.2. The $n$-th plate of a binary distillation column having only liquid feed.

Referring to Fig. 17.2, we make the following assumptions:

(i) the molar vapour accumulation per tray may be neglected. This is often reasonable since although the volume the vapour occupies is large, the number of moles present is small;
the molar heat of vapourization of both components is equal. This means that one mole of condensing vapour will liberate enough heat to allow a mole of fluid to evaporate. So long as no exceptional heat losses occur in the column, then it follows that

\[ V_n = V_{n+1} = V_{n-1} = V \]  

(17-1)

The above approximation removes the need for energy or heat balance equations in our tray model.

The overall mass balance for tray \( n \) can now be written down:

\[ G_n + L_{n+1} - L_n = \frac{dM_n}{dt}. \]  

(17-2)

17.2.2. component balance for tray \( n \)

A component balance for the lighter of the two components yields:

\[ G_n x_n + V_n \cdot y_{n-1} + L_{n+1} \cdot x_{n+1} - V_n \cdot y_n - L_n \cdot x_n = \frac{d}{dt}(M_n \cdot x_n) \]  

(17-3)

where it has been assumed that the liquid feed is at its bubble point.

17.2.3. equilibrium relationship

If the relative volatility, \( \alpha \), can be taken as constant and if the trays are 100% efficient (i.e. liquid and vapour instantaneously in equilibrium on each tray), then

\[ y_n = \frac{\alpha x_n}{1 + (\alpha - 1) x_n} \]  

(17-4)

\[ = f_1(x_n), \text{ see Fig. 17-3.} \]

Fig. 17.3. The relative volatility relationship, eq. (17-4).
With volatility defined as

\[ \rho_i = \frac{p_i}{x_i} \]

where \( p_i \) is the partial pressure of component \( i \), and with relative volatility given by

\[ a_{ij} = \frac{\rho_i}{\rho_j} \]

prove that eq. (17-4) holds for a binary distillation [74].

We note here that for real (as opposed to theoretical) trays the Murphree tray efficiency equation [67],[74] must be applied.

### 17.2.4. Hydraulic relationship

The relationship between the accumulated liquid, \( M_n \), and the molar flow of liquid leaving plate \( n \) for plate \( n-1 \) can be described by the Francis weir formula [68] — hence the name, hydraulic relationship:

\[ h_n = K_n \frac{M_n}{L_n} \]

or

\[ M_n = f_2(L_n). \]

Q. 81: Has the role the liquid density plays been forgotten in eq. (17-5)?

Equations (17-1) to (17-5) express the dynamic behaviour of an ideal binary tray. For a tray without external feed the model is:

\[ \frac{dx_n}{dt} = V \left[ f(x_{n-1}) - f(x_n) \right] + L_{n+1} \left( x_{n+1} - x_n \right) \]  
\[ f_2(L_n) \]

\[ \frac{df_2(L_n)}{dt} = L_{n+1} - L_n. \]  

Unefortunately, in spite of all our simplifying assumptions, this model is not amenable to elementary techniques of linear analysis; moreover we have two equations for every tray in a column, which would result in a formidable model dimension in most practical cases. One possible further simplification can be obtained at the expense of the liquid holdup, \( M_n \). If this latter quantity is assumed constant on every tray, then the model reduces to one equation per tray:

\[ M \frac{dx_n}{dt} = V f_1(x_{n-1}) - V f_1(x_n) + L(x_{n+1} - x_n) \]

where \( L \) is now constant both above and below the feedplate (in the “rectifying” and in the “stripping” sections). The nonlinearities in eq. (17-8) may, of course, be removed by linearisation about an operating point.
17.3. extension to a column model

When extending the ideal plate model described in the previous section to predict the behaviour of a complete binary distillation unit, we must include the dynamics of the condenser, reflux accumulator, reflux lines and reboiler. A frequently used model for the top product periferals is the following:

\[ M_{N+1} \frac{dx_D}{dt} = V y_N - (L_{N+1} + D) x_D \quad (17-9) \]

\[ V = L_{N+1} + D. \quad (17-10) \]

Here \( L_{N+1} \) is the external reflux and \( D \) the top product molar flowrate. \( M_{N+1} \), the reflux accumulation, is frequently of much greater magnitude than that which was encountered on a tray. The model described by eqs. (17-9) and (17-10) must be used with discretion; if vapour and/or reflux line dead times are appreciable (Fig. 17.4) they can play an important role in stability studies.

![Diagram of column top](image)

Fig. 17.4. Model of column top.

The reboiler can also be taken to be an extra tray. If the bottom product is \( B \) (moles/s) we have:

\[ M_0 \frac{dx_B}{dt} = L_1 x_1 - B x_B - V y_0 \quad (17-11) \]

\[ V = L_1 - B. \quad (17-12) \]

An equilibrium relationship such as eq. (17-4) may be used to describe the interaction between \( x_B \) and \( y_0 \). Again, \( M_0 \) will usually be very much greater than \( M_n \).

Our complete binary distillation column can now be built up from eqs. (17-9) to (17-12) plus \((N - 1)\) equation (17-8)'s and a feedplate model.
Q. 83: Set up a 5 plate column model, which receives its fluid feed on plate number 2.

17.4. distributed parameter models

A completely different approach is to describe the distillation column by means of a (relatively) few partial differential equations. Such a model has been used to advantage for column stability [73] and control [24] studies.

Using the usual Taylor series expansion it is possible to transform the discrete (in the spatial coordinate) variable \( x_{n+1} \) into a continuous form \( x(z + k) \), where \( k \) is the distance between tray \( n \) and tray \( n + 1 \):

\[
x_{n+1} \approx x(z) + k \frac{dx}{dz} + \frac{k^2}{2} \frac{d^2x}{dz^2}.
\]

Similarly,

\[
f_1(x_{n-1}) \approx f_1(x(z)) - k \frac{df_1}{dz} + \frac{k^2}{2} \frac{d^2f_1}{dz^2}.
\]

We can now write the simple tray model, eq. (17-8), as [72]:

\[
\frac{M}{k} \frac{d\xi}{dt} = \frac{d}{dz} \left( L \xi - Vf_1(\xi) \right) + \frac{k^2}{2} \frac{d^2\xi}{dz^2}.
\]

Gould [74] discusses some of the interesting properties of this distributed parameter tray model.

Q. 84: What are the boundary conditions of eq. (17-15)?

Brosilow et al. [82] have quantified a model, similar to that developed here, into a small number of ordinary differential equations. With a substantial saving in the number of equations that would otherwise (if, say eq. (17-8) were chosen as the plate model) be necessary, they claim a very reasonable prediction of both top and bottom product dynamic performance is possible.

17.5. models for control purposes

If the purpose of our effort is to develop simple column control schemes or to adjust controller settings, then the simple “input-output” column model (Fig. 17.5) is to be preferred above the relatively complicated models described up till now [39].

Moreover, it is often possible to arrive at a sound control scheme using only stationary models; dynamic compensation (e.g. PID controllers, low pass filters, etc.) can be added later, and tuned on line.

A stationary model of a column can be found from the overall molar balances:

\[
G = D + B \quad (17-16)
\]

\[
x_G \cdot G = x_D \cdot D + x_B \cdot B \quad (17-17)
\]

and from balances around both rectification and stripping sections of the column:

247
Some research has been carried out into extending these stationary equations to fit plant test data. For example, from eqs. (17-16) and (17-17), since \( x_B \approx 0 \), we have that

\[
x_D = \frac{x_G}{D} \tag{17-22}
\]

Now if the dynamics of the transmittance between \( x_D \) and \( G \) are adequately described by a second-order plus dead-time element, it follows [56] that (at any rate for small deviations from the stationary, equilibrium state):

\[
\frac{x_D(s)}{G(s)} = \frac{k_c e^{-sT}}{\tau_1 s^2 + \tau_2 s + 1} . \tag{17-23}
\]

17.6. the basic principles of column control

We begin by making an inventory of all the variables which could be important in determining what control scheme should be used to regulate a binary distillation column. The two most important external sources of disturbance are likely to be:

(I) the feed composition, \( x_G \);

(II) the feedrate, \( G \).
Both are usually determined by the performance of, or conditions prevailing in, preceding units in the distillation train and often cannot be stabilized prior to entry into the column. Nearly always controlled — although it seems with little interaction upon other variables — are

(III) the column pressure;

(IV) feed enthalpy.

(III) influences the equilibrium existing between liquid and vapour (eq. (17-4)) and can usually be controlled through the adjustment of the flow of water to the condenser (Fig. 17.6). Of course, it must be assumed that the water temperature (and also the ambient temperature) remains constant; otherwise the condenser heat balance will be upset with a possible unfavourable effect upon control. Apart from this hazard, it is fairly easy to show that an increase in the condenser cooling water flowrate will lead to a lower column pressure. In practice, however, this relationship is in general a very slow one.

![Control of column pressure diagram](image)

The feed enthalpy (IV) can be manipulated to some extent if a feed preheater is present. It is necessary to repeat here our warning about the use of surplus bottom product heat [74]; this can lead to instabilities and in any event makes the job of holding the enthalpy of the feed constant more difficult.

There remains to be controlled the

(V) liquid flowrate, \( L_r \), through the rectifying section of the column and

(VI) the vapour flow, \( V \), through the stripping section,

if the effect of the disturbances mentioned earlier is to be minimized. Since there is considerable interaction between (V) and (VI) it is here that the heart of the (binary) distillation column control problem exists. That is also why Fig. 17.5 — where \( V \) has been replaced by the reflux — is a useful elementary
model for control purposes.

The most obvious solution of this control problem is — from Fig. 17.5 — feedback of the two product compositions. However, that would involve the use of a chromatograph (with a large sampling interval); often we settle with taking an indirect measurement of the composition. Thus, the temperature of the liquid on the top tray can be used to control the external reflux — see Fig. 17.7. If, for example, the purity of the top product declines, then the boiling point will increase, raising the temperature of the liquid on the top tray. This temperature can be brought down by allowing extra reflux (from the reflux accumulator — see introduction, Section 17.1) into the column.

![Diagram of distillate composition control loop.](image)

**Fig. 17.7.** Distillate composition control loop.

The vapour flow from the reboiler (VI) is usually controlled by manipulating the steam supply to the reboiler — Fig. 17.8. There is clearly no question here of trying to compensate for changes in bottom product purity.

![Diagram of reboiler vapour rate control scheme.](image)

**Fig. 17.8.** Reboiler vapour rate control scheme.

Finally we draw the reader’s attention to the fact that large capacities in
the condenser or reboiler recirculation loops are not desirable from the point of view of control. Buckley [70] gives several alternatives to the use of a large reflux accumulation tank and a reasoned motivation for using a thermosiphon type reboiler.

17.7. feedforward control

The logical next step in the improvement of the basic column control scheme just outlined is obviously the provision of some sort of feedforward action; this, together with the feedback control necessary for "fine" control (Ch. 10, p. 105), to compensate for any fluctuations in feed rate or composition.

Nisenfeld and Miyasaki [39],[40] present some very interesting examples of how feedforward control has successfully been applied to actual distillation units. We follow here their reasoning and note, in passing, that it gives a good picture of the practically-orientated approach.

It appears that for feedforward control of a column, a prediction based upon a very simple stationary model is often entirely adequate. This model, combined with the necessary dynamic compensation, results in practical lead-or-lag-networks (Ch. 9, p. 100). We begin by combining the mass balance given by eq. (17-16) with the overall component balance eq. (17-17):

\[ D = \frac{x_G - x_B}{x_D - x_B} \cdot G. \]  

(17-24)

The only other necessary equation is in the form of an empirical relationship [79]:

\[ \frac{V}{G} = k \ln S, \]  

(17-25)

where S is the separation factor, defined by the equation:

\[ S = \frac{x_D - 1}{1 - x_D} \cdot \frac{1 - x_B}{x_B}, \]  

(17-26)

and k is a (constant) parameter of the type of column, number of trays etc. Nisenfeld and Miyasaki show how eq. (17-24) can be used for a continuous prediction of top product production, D. Then, if it is possible to hold the ratio \( \frac{V}{G} \) constant, \( x_B \) will be specified to a certain value, depending on \( x_D \), since the separation factor is constant (eq. (17-25)). Furthermore again from eq. (17-24), we see that in fact the ratio \( \frac{D}{G} \) need only be altered if the feed composition \( x_B \) changes.

The application of the simple, but logical, model outlined above to a column depends upon particular circumstances. The previously named authors present several relevant examples.

17.8. analysis of the structure of a distillation column model

Considerable insight into how distillation column models and control schemes
work can be gained by applying the techniques presented in Chapter 14 to a simple case. We take a three tray column (middle tray is feedtray) with conventional condenser and reboiler; larger models can be treated by the same method of analysis, but the amount of data processing makes the use of a computer a "must". In the stationary state there are twelve [74] equations necessary (we assume that all the vapour condenses to liquid in the condenser) to describe the behaviour of the column in the simplest possible terms.

\[
\begin{align*}
    f_1 &= -V y_0 + L_s x_1 - B x_B = 0 \\
    f_2 &= V - L_s + B = 0 \\
    f_3 &= V(y_0 - y_1) + L_s (x_2 - x_1) = 0 \\
    f_4 &= V(y_1 - y_2) + L_s x_3 - L_s x_2 + x_G G = 0 \\
    f_5 &= V(y_2 - y_3) + L_s (x_D - x_3) = 0 \\
    f_6 &= V y_3 - (L_r + D) x_D = 0 \\
    f_7 &= V - L_r - D = 0 \\
    f_8 &= G - D - B = 0 \\
    f_9 &= y_0 - \frac{a x_B}{1 + (a - 1) x_B} = 0 \\
    f_{10} &= y_1 - \frac{a x_1}{1 + (a - 1) x_1} = 0 \\
    f_{11} &= y_2 - \frac{a x_2}{1 + (a - 1) x_2} = 0 \\
    f_{12} &= y_3 - \frac{a x_3}{1 + (a - 1) x_3} = 0
\end{align*}
\]

The following values are assumed:

\[
\begin{align*}
    a &= 2.0 & G &= 100 \text{ moles/s} \\
    x_1 &= 0.2 & D &= 37.5 \text{ } \\
    x_2 &= 0.5 & L_r &= 30 \text{ } \\
    x_3 &= 0.9 & \\
    y_0 &= 0.25 & x_G &= 0.6 \\
    y_1 &= 0.33 & x_D &= 0.947 \\
    y_2 &= 0.66 & x_B &= 0.144 \\
    y_3 &= 0.947 &
\end{align*}
\]

and
\[ B = 62.5 \text{ moles/s} \]
\[ V = 67.5 \text{ moles/s} \]
\[ L_s = 130 \text{ moles/s}. \]

Assuming that there are no more than twelve independent equations then there is clearly a deficit of information, since we have 16 variables in our model. The first problem, then, is to decide which variables must be regarded as disturbance variables. In our case this is not difficult; \( x_G \) and \( G \) are set by another upstream unit and we can do little but accept them as variables which cannot be influenced or manipulated within the reference frame of our model. These two variables play no further part in our analysis and can be thought of henceforth as being merely plant parameters.

The assignment of the distinct representatives (see Section 14.4.1) follows as the next step in the analysis. For this, the twelve functions are linearised about the equilibrium state given; the weighting factors are computed and normalised, those variables having a unity weighting factor being the least sensitive variables. Beginning with those functions having the least number of variables present and working through to the functions of the most variables, an assignment of distinct representatives is made based upon the least — or the nearly least — sensitive variables. The results of the assignment are given in tables I and II below.

### Table I

<table>
<thead>
<tr>
<th>( V )</th>
<th>( L_s )</th>
<th>( B )</th>
<th>( L_r )</th>
<th>( D )</th>
<th>( y_0 )</th>
<th>( x_1 )</th>
<th>( x_B )</th>
<th>( y_1 )</th>
<th>( x_2 )</th>
<th>( y_2 )</th>
<th>( x_3 )</th>
<th>( x_3 )</th>
<th>( x_D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>.649</td>
<td>1.0</td>
<td>.346</td>
<td>.481</td>
<td>.649</td>
<td>1.0</td>
<td>.346</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_2 )</td>
<td>.519</td>
<td>1.0</td>
<td>.415</td>
<td>.022</td>
<td>.519</td>
<td>1.0</td>
<td>.415</td>
<td>.698</td>
<td>.423</td>
<td>1.0</td>
<td>.445</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_3 )</td>
<td>.083</td>
<td>.6</td>
<td>.445</td>
<td>.556</td>
<td>.083</td>
<td>.6</td>
<td>.445</td>
<td>.556</td>
<td>.698</td>
<td>.423</td>
<td>1.0</td>
<td>.445</td>
<td></td>
</tr>
<tr>
<td>( f_4 )</td>
<td>.346</td>
<td>1.0</td>
<td>.444</td>
<td>.556</td>
<td>.346</td>
<td>1.0</td>
<td>.444</td>
<td>.556</td>
<td>.698</td>
<td>.423</td>
<td>1.0</td>
<td>.445</td>
<td></td>
</tr>
<tr>
<td>( f_5 )</td>
<td>.296</td>
<td>1.0</td>
<td>.444</td>
<td>.556</td>
<td>.296</td>
<td>1.0</td>
<td>.444</td>
<td>.556</td>
<td>.698</td>
<td>.423</td>
<td>1.0</td>
<td>.445</td>
<td></td>
</tr>
<tr>
<td>( f_6 )</td>
<td>1.0</td>
<td>.4</td>
<td>.163</td>
<td>.23</td>
<td>1.0</td>
<td>.4</td>
<td>.163</td>
<td>.23</td>
<td>1.0</td>
<td>.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_7 )</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_8 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_9 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{10} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{11} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{12} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The rings indicate the assignment of a distinct representative.
Table II.
\[ y_0 = f_1'(V, L_s, B, x_1, x_B) \]
\[ L_s = f_2'(V, B) \]
\[ y_1 = f_3'(V, L_s, y_0, x_1, x_2) \]
\[ x_3 = f_4'(V, L_s, L_r, y_1, x_2, y_3) \]
\[ y_2 = f_5'(V, L_r, x_3, y_3, x_D) \]
\[ x_D = f_6'(V, L_r, D, y_3) \]
\[ V = f_7'(L_r, D) \]
\[ B = f_8'(D) \]
\[ x_B = f_9'(y_0) \]
\[ x_1 = f_{10}'(y_1) \]
\[ x_2 = f_{11}'(y_2) \]
\[ y_3 = f_{12}'(x_3) \]

The reader will find, if he checks the calculations, that \( x_3 \) is an equally good choice as \( L_r \) for the distinct representative of function \( f_4 \). In general it should be noted that the assignment often is not unique; as such, our example here is typical.

Note that as a consequence of there being too few functions in our model, the variables \( L_r \) and \( D \) remain unassigned. The deficit in functions must be made up by us in the form of two controller functions, \( f_{13} \) and \( f_{14} \) – hence \( L_r \) and \( D \) would be a good choice (if no other considerations are present) for the external feedback. Ideally, of course, for such a role \( L_r \) and \( D \) should also be the most sensitive variables in the functions where they occur in our model. Still, we have at any rate provided some reasoned guidance as to the problem of “which variables do I control” and “how many control loops should there be”.

Leaving aside, for the moment, variables \( L_r \) and \( D \), we can construct the “open loop” multi-level matrix, shown below.

![Open loop multi-level matrix](image-url)
We see that there are four subsystems arranged in three hierarchies (Fig. 17.9). The distinct representatives appear along the main diagonal; any entry above the main diagonal signals the presence of an internal feedback variable. We have tried, by arranging the functions in different orders, to minimize the number of variables internally fed back – which is four in our example. Again, it must be noted that the choice of four variables is not unique. Note also that the choice of $f_{i,1}$ as the first (upper left) entry in the multi-level matrix is unique – it follows from our definition (Chapter 14) of a subsystem.

It may be concluded from a study of the open loop (corresponding to operation of the column with $L_r$ and $D$ held constant all the time) that all the variables will suffer as an affect of deviations in feedrate or composition, some more than others. For example a change in feedrate will cause $B$ to deviate – but the effect would be only about half as great as if $D$ were free and $B$ held constant, since $B$ (see Table I) is the least sensitive variable of $f_8$. We can also trace the passage of a disturbance through the model: suppose, again, that the feedrate $G$ changes. Through $f_8$ $B$ changes and so does $y_o$ (through $f_{i,1}$). But the result is then that $x_B$ changes ($f_{i,1}$) – contrary to the desired column performance.

If we take account of external feedback, the multi-level closed loop matrix
shown below is obtained. The effect of control, through $f_{13}$ and $f_{14}$ is that any decomposition of the system into levels or subsystems impossible*. It is, however, believed that this sort of pictorial analysis might aid the selection of variables to be measured (these would appear as entries in the last two rows of the closed loop multi-level matrix).

<table>
<thead>
<tr>
<th>Subsystems</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disturbances</td>
<td>Functions</td>
</tr>
<tr>
<td>Internal</td>
<td>Feedback</td>
</tr>
<tr>
<td>External</td>
<td>Feedback</td>
</tr>
</tbody>
</table>

* Impossible, that is, in terms of the analysis presented in Chapter 14.
17.9. a survey of recent literature

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Technological background</td>
<td>67,74,76-7</td>
</tr>
<tr>
<td>2. Surveys, reviews, bibliographies</td>
<td>52-3,63,83</td>
</tr>
<tr>
<td>3. Models</td>
<td>8,10,13,19,20,24-5,27,35, 37,39,40,47,49,51,56,58-9, 61-3,65,72,74-6,82-3</td>
</tr>
<tr>
<td>4. Stability</td>
<td>15,20,44</td>
</tr>
<tr>
<td>5. Simulation</td>
<td>6,15,27,32,49,51,59,63,76-7</td>
</tr>
<tr>
<td>6. Experimental approach</td>
<td>30,33-4,36,38,42,45,55,57, 60-1,63,80</td>
</tr>
<tr>
<td>7. Computer control</td>
<td>1-18,26,33,37,42,81</td>
</tr>
<tr>
<td>8. Modern control theory, optimal control</td>
<td>3,4,12,18-29,41,50,57,61, 64,73,81</td>
</tr>
<tr>
<td>9. Feedforward control</td>
<td>5,39,40,42,47,55,70,79</td>
</tr>
<tr>
<td>10. Complex control schemes</td>
<td>14,30-32,35,38,42-44,46, 48,52-6,65,68,70-1,78-80</td>
</tr>
<tr>
<td>11. Non-linearities</td>
<td>23,34,49,62,75</td>
</tr>
<tr>
<td>12. Adaptive control</td>
<td>33,37,47,60</td>
</tr>
<tr>
<td>13. Modal control</td>
<td>19,41,71,74</td>
</tr>
<tr>
<td>14. Control of temperature</td>
<td>24,34,36,43,47,49,75</td>
</tr>
</tbody>
</table>

17.10. references


[16] ALLEN, P. and PHILIPS, C. L. Batch process control using a digital computer, 3rd IFAC/IFIP Conf. on digital computer applications to process control, Helsinki, June 1971.

[17] JEDLOVSKY, P. and MIKA, I. Processing of measurement data in an oil refinery, Conf. on "Control 70", Miskolc (Hungary), July 1970.

[18] SER, V., INZELT, P., VADNAY, Sz. The project of computer control of the distillation unit in the "Danube Mineral Oil Industrial Enterprise", Conf. on "Control 70", Miskolc (Hungary), July 1970.


[40] —— Feedforward control solves problems, Control & Instrum., 5, pp. 34-7 (March 1973).


[51] WAGGONER, R. C. and HALL, E. B. Process simulations for distillation column control. See ref. [50].

[52] JONES, J. B. and HALL, E. B. A survey of difficult distillation control problems. See ref. [50].

[53] WILLIAMS, T. J. and HALL, E. B. Distillation column control systems: a review, a compilation and an analysis. See ref. [50].


[57] NOUGANET, M., BACK, H., CANJOLLE, J. C. Harmonic analysis via binary test signal with quasi sinusoidal autocorrelation function. See ref. [33].


[60] SMITH, C. L., CORRIPIO, A. B., HALL, E. B. On-line identification of distillation column dynamics. See ref. [50].


[73] —— A Lyapunov function with applications to some nonlinear physical systems, Automatica, 1, pp. 31-53 (1963).


DYNAMICS AND CONTROL OF CHEMICAL REACTORS

18.1. introduction

Chemical reactors frequently pose some of the most challenging control problems encountered in process control, especially when rapid and exothermic reactions are taking place. It is therefore particularly important to describe their dynamic behaviour as accurately as possible; inherent nonlinearities often make the use of a linear model questionable. However, with a linear model a qualitative feel for the dynamic response and effect of various parameters is usually possible, and in this chapter it is the linear models which will receive most of our attention.

We can distinguish between three types of chemical reactor on the grounds of the passage of material through the reactor:

i) the batch reactor, where all reactants are placed in the reactor, the reaction started, and no additional feedstuff or products allowed to enter or leave before the reaction is completed;

ii) the ideal tank reactor, where reactants flow continuously through the reactor and the mixing in the tank is ideal, and thirdly

iii) the ideal pipe reactor, where again reactants flow continuously through the reactor but this time in perfect plug flow.

Types (ii) and (iii) are referred to as continuous reactors; real continuous reactors operate somewhere between (ii) and (iii), where our naive ideas of ideal behaviour must be supplemented by studies of residence time distributions and mixing on a microscopic scale to account for the effects of non-ideal mixing in (ii) and axial diffusion in (iii).

The batch reactor has already been studied (p. 52, Chapter 4; Chapter 12) and we shall concern ourselves in this chapter only with continuous reactors. Furthermore, in the stated spirit of this book, only homogeneous – that is to say single phase – reactions are investigated. Heterogeneous reactions bring yet more complications into the picture.

18.2. the dynamics of an ideal tank reactor

18.2.1. isothermal reaction

We suppose that a certain chemical A reacts together with another, unspecified substance (present in unlimited quantity) to give a product B, in the tank reactor shown in Fig. 18.1.
The following assumptions will be made:

- The feed \( Q_0 \) (m³/s) contains component A in the concentration \( C_{A0} \) (mol/m³).

- The product stream \( Q \) contains both A (unreacted) and product B, in the concentrations \( C_A \) and \( C_B \) respectively. No volume change accompanies the chemical reactions, so that \( Q = Q_0 \).

- The tank reactor is ideally stirred so that the species concentrations in the tank are everywhere equal to the output concentrations.

- A first order, irreducible reaction takes place, \( A \xrightarrow{k} B \), where \( k \) is the reaction velocity constant*. The reaction rate, \( r_A \) (mol/m³ s) of the reaction is given by the formula:

\[
 r_A = kC_A
\]  

(18-1)

- Finally we assume that the reaction is isothermal, which implies that \( k \) is constant.

A component A mass balance yields

\[
 V \frac{dC_A}{dt} = QC_{A0} - QC_A - kC_AV,
\]  

(18-2)

which upon substitution of the average residence time, \( \tau_V (\equiv V/Q) \) in seconds reduces to

\[
 \tau_V \frac{dC_A}{dt} = C_{A0} - C_A - k\tau_A C_A
\]  

(18-3)

* Also called the rate constant or specific reaction rate.
It is possible to write now a transfer function, from the feed reactant concentration $C_{A_0}$ to the outlet concentration of remaining, unreacted $A$ as

$$\frac{C_A(s)}{C_{A_0}(s)} = \frac{T_1}{1 + sT_1}, \tag{18-4}$$

where

$$T_1 = \frac{\tau_V}{1 + k\tau_V} \tag{18-5}$$

is the time constant of the process. Clearly, from this expression $T_1 < \tau_V$; for very slow reactions where $k > 0$, then $T_1 \approx \tau_V$ and the transfer function just derived approaches $1/(1 + s\tau_V)$, the transfer function of a perfect mixer. At the opposite end of the scale, if the reaction is extremely fast ($k \rightarrow \infty$), then $T_1 \rightarrow 0$ and the limit of the transfer function is zero.

An overall mass balance gives

$$\frac{d}{dt}(C_A + C_B) = Q(C_{A_0} - Q(C_A + C_B)). \tag{18-6}$$

Laplace transforming and substituting $\tau_V$, we find that

$$\frac{C_A(s) + C_B(s)}{C_{A_0}(s)} = \frac{1}{1 + s\tau_V} \tag{18-7}$$

which is again the transfer function of an ideal mixer. Of course, there is no distinction made by the overall mass balance between reacted and unreacted components. Eqs. (18-4) and (18-7) give us directly the transfer function of the feed concentration $C_{A_0}$ to the product output concentration:

$$\frac{C_B}{C_{A_0}} = \frac{1}{1 + s\tau_V} - \frac{T_1/\tau_V}{1 + sT_1} = \frac{kT_1}{(1 + s\tau_V)(1 + sT_1)}. \tag{18-8}$$

This second order transfer function is characterized by one time constant $\tau_V$ resulting from the mixing in the reactor, and one time constant $T_1 = \tau_V/(1 + k\tau_V)$ resulting from the reaction itself. By substituting $s = 0$ in eq. (18-8) we can find which fraction $K$ of component $A$ is converted into product $B$ when the reactor is operating in a steady state. Thus

$$K = \frac{k\tau_V}{1 + k\tau_V}, \tag{18-9}$$

where $K$ is the so-called conversion.

It is often possible to describe the dynamic behaviour of more complex reactions, such as equilibrium and consecutive reactions, in an analogous fashion.
Example 18.1.

Consider the reaction just described, but with the extra complication that the product B can decompose back into A, with reaction velocity constant $k_2$. The velocity constant from A to B we will call $k_1$ instead of $k$, so that

$$
\frac{k_1}{k_2} \quad A \xrightarrow{k_1} B.
$$

For the reaction rate we have that

$$
r = k_1C_A - k_2C_B.
$$

(18-10)

In analogy with eq. (18-3), the component balance on A yields:

$$
\tau_V \frac{dC_A}{dt} = C_{A_0} - C_A - k_1\tau_VC_A + k_2\tau_VC_B
$$

(18-11)

or in the Laplace domain:

$$
s\tau_V C_A^\ast - C_{A_0}^\ast = C_A^\ast - k_1\tau_VC_A^\ast + k_2\tau_VC_B^\ast.
$$

(18-12)

The overall mass balance (compare with eq. (18-6)) after transforming becomes

$$
s\tau_V(C_A^\ast + C_B^\ast) = C_{A_0}^\ast - (C_A^\ast + C_B^\ast).
$$

(18-13)

We now have, in eqs. (18-12) and (18-13) two equations with two unknowns $C_A^\ast$ and $C_B^\ast$. After some manipulation:

$$
\frac{C_A^\ast}{C_{A_0}^\ast} = \frac{(l + k_1\tau_V + s\tau_V)T_2/\tau_V}{(l + s\tau_V)(l + sT_2)}
$$

(18-14)

and

$$
\frac{C_B^\ast}{C_{A_0}^\ast} = \frac{k_1T_2}{(l + s\tau_V)(l + sT_2)}.
$$

(18-15)

where

$$
T_2 \equiv \frac{\tau_V}{l + (k_1 + k_2)\tau_V}.
$$

(18-16)

On substituting $s = 0$ it follows from eq. (18-15) that the conversion $K$ is given by

$$
K = k_1T_2 = \frac{k_1\tau_V}{l + (k_1 + k_2)\tau_V}.
$$

(18-17)

For large values of $\tau_V$, the conversion approaches the value $k_1/k_1 + k_2$.

From this example it can be seen that the dynamics of the simple equilibrium reaction is second order. For a consecutive reaction the dynamics exhibit higher order behaviour, however.

Q. 85. Suppose that in the ideal tank reactor of Fig. 18.1 the first order reaction of feed A to product B proceeds to completion. However, we assume that simultaneously another first order irreversible reaction is responsible for the decomposition of product B to hy-
product C, so that

\[ \frac{k_1}{A} \rightarrow B \rightarrow \frac{k_2}{C}. \]

Calculate the transfer functions \( \frac{\bar{C}_A}{\bar{C}_{A_0}}, \frac{\bar{C}_B}{\bar{C}_{B_0}} \) and \( \frac{\bar{C}_C}{\bar{C}_{C_0}}. \)

### 18.2.2. non-isothermal reaction

Because of our previous assumption of an isothermal reaction we have only had to contend so far with mass balances and not with energy balances. In order now to contrast the situation when this assumption no longer holds we consider exactly the same process and conditions as in section 18.2.1 with the exception that the reaction is now taken to be exothermic. The liberated heat must, of course, be removed; this is done by incorporating a cooling circuit in the reactor, see Fig. 18.2.

![Fig. 18.2. Tank reactor with cooler.](image)

Besides the points mentioned in the last section, the following assumptions and notational questions should be noted:

- the heat of reaction is denoted by \( \Delta H \) (J/mol)
- the heat capacity of the contents of the reactor is constant and given by \( c \) (J/°Cm³)
- the cooling loop has a heat transfer coefficient \( h \) (J/°Cm²s) and surface area \( A \) (m²).
- the temperature of the coolant is overall \( \theta_k \) (°C). The reactor feed temperature is \( \theta_0 \) (°C) and everywhere within the reactor the contents are at \( \theta \) (°C).
The reaction velocity constant, \( k \), depends upon the temperature according to Arrhenius relationship:

\[
k = k_0 e^{-\frac{E}{R(\theta + 273)}},
\]

(18-18)

where \( E \) is the activation energy, \( R \) the gas constant and \( (\theta + 273) \) converts the temperature \( \theta \) (°C) to the absolute temperature (K). For a first order reaction which proceeds to completion the reaction rate is given by

\[
r_A = kC_A = k_0 C_A e^{-\frac{E}{R(\theta + 273)}}.
\]

(18-19)

Linearization of this expression will give a linear equation having the form

\[
\Delta r_A = a_c \Delta C_A + a_\theta \Delta \theta,
\]

(18-20)

where

\[
a_c = k_0 e^{-\frac{E}{R(\theta + 273)}}
\]

and

\[
a_\theta = \frac{k_0 C_A e^{-\frac{E}{R(\theta + 273)}}}{R(\theta + 273)^2}.
\]

An energy balance around the reactor is now made. There results

\[
Vc \frac{d\theta}{dt} = Qc\theta_0 - Qc\theta + aVr_A - hA(\theta - \theta_k),
\]

(18-22)

and this expression is also valid for small perturbations \( \Delta \theta, \Delta \theta_0 \) and \( \Delta \theta_k \) about \( \theta, \theta_0 \) and \( \theta_k \). If we divide all terms of the perturbed form of eq. (18-22) by \( Qc \), substitute \( \tau_v \) for \( V/Q \) and take Laplace transforms, then we find eventually that

\[
s \tau_v \Delta \theta = \Delta \theta_0 - \Delta \theta + \frac{a}{c} \tau_v \Delta r_A - \frac{hA}{Qc} (\Delta \theta - \Delta \theta_k) =
\]

\[
= \Delta \theta_0 - \Delta \theta (1 + \frac{hA}{Qc} + \Delta \theta \frac{hA}{kQc} + \frac{a}{c} \tau_v \Delta r_A).\]

(18-23)

Of course, the overall mass balance and the component balance for \( A \) are still valid. Thus, using perturbation variables in eq. (18-3) and (18-6) and Laplace transforming results in

\[
s \tau_v \Delta C_A = \Delta C_{A_0} - \Delta C_A - \tau_v \Delta r_A
\]

(18-24)

and

\[
s \tau_v (\Delta C_A + \Delta C_B) = \Delta C_{A_0} - \Delta C_A - \Delta C_B.
\]

(18-25)

Equations (18-20), (18-23), (18-24) and (18-25) together form a system of four equations with three independent variables \( \Delta \theta_0, \Delta C_{A_0}, \Delta \theta_k \) and four
dependent variables $\Delta r_A$, $\Delta \theta$, $\Delta C_A$, and $\Delta C_B$.

We are therefore in a position to determine the transfer functions from $\Delta C_{A_0}$, $\Delta \theta_0$ and $\Delta \theta_k$ to $\Delta C_A$, $\Delta C_B$ and $\Delta \theta$.

Q. 86: Calculate the transfer functions $\Delta C_A/\Delta C_{A_0}$, $\Delta \theta/\Delta \theta_0$ and $\Delta \theta/\Delta \theta_k$. Is there anything remarkable about the denominators of these transfer functions when they are compared with one another?

In what follows we take just one of the possible transfer functions. If $\theta_0$ and $\theta_k$ do not change (i.e. $\Delta \theta_0$, $\Delta \theta_k = 0$) then the transfer function between $\Delta C_{A_0}$ and $\Delta \theta$ can be written:

$$
\Delta \theta = \frac{a}{c} \frac{\tau_1 a_c}{s^2 \tau_1 \tau_V + s(\tau_1 + \tau_V - \frac{a}{c} a_\theta \tau_1 \tau_V + a_c \tau_1 \tau_V) + 1 - \frac{a}{c} \tau_1 a_\theta + \tau_V a_c} (18-26)
$$

where

$$
\tau_1 = \frac{\tau_V}{1 + \frac{hA}{Qc}} (18-27)
$$

In order to guarantee stability, the real parts of the roots of the denominator (i.e. the poles of the transfer function) of eq. (18-26) must be negative. Since the coefficient of $s^2$ is positive, it follows from the Routh-Hurwitz criterion (p. 118, Chapter 11) that the other coefficients of the denominator polynomial must also all be positive. In other words, two conditions for stable dynamic operation of the reactor are:

I) $1 - \frac{a}{c} \tau_1 a_\theta + \tau_V a_c > 0$ (18-28)

II) $\tau_1 + \tau_V - \frac{a}{c} a_\theta \tau_1 \tau_V + a_c \tau_1 \tau_V > 0$ (18-29)

To provide some insight into these conditions, suppose for a moment that the reactor is in some stationary state; then from eq. (18-24) with $s = 0$ and $\Delta C_{A_0} = 0$

$$
\Delta C_A = -\tau_V \Delta r_A. (18-30)
$$

Substituting this in eq. (18-20):

$$
(1 + a_c \tau_V) \Delta r_A = a_\theta \Delta \theta (18-31)
$$

and so the first condition for stability becomes

$$
\frac{a}{c} \tau_1 \Delta r_A < \Delta \theta, (18-32)
$$

which in view of the definition, eq. (18-27), for $\tau_1$ is the same as

$$
aV \Delta r_A < (Qc + hA) \Delta \theta. (18-33)
$$
The last equation states that any increase in heat liberated during the reaction must be removed by a corresponding increase in the dissipated heat. If this last quantity is denoted by \( H_d \) (J/s) and the heat developed during reaction by \( H_l \), then

\[
H_d = hA(\theta - \theta_k) + Q_c(\theta - \theta_0) = (hA + Q_c)\theta + \text{constant} \quad (18-34)
\]

and

\[
H_l = ar_A V. \quad (18-35)
\]

In Fig. 18.3 \( H_d \) and \( H_l \) are shown as a function of \( \theta \).

Fig. 18.3. Rates of heat liberation and removal.

The condition, eq. (18-33), can be interpreted in the light of Fig. 18.3 as saying that the slope of \( H_l = H_l(\theta) \) must be less than the slope of \( H_d = H_d(\theta) \).

So although at both points A and B of Fig. 18.3 the energy balance is satisfied, it is now clear that point A is a point of unstable operation, while B is a stable point.

If it is permissible to loosely speak of the first condition (eq. (18-28)) as being a "stationary stability condition" then the second condition, eq. (18-29), is a dynamic stability condition; it can be shown that it plays an important role if there is too strong a coupling between the mass and energy balances, which creates oscillatory behaviour.

Q. 87: Write down the energy balance for the consecutive reaction \( A \xrightarrow{k_1} B \xrightarrow{k_2} C \), where the reactions go to completion and are of first order. The heats of reaction are \( a_1 \) and \( a_2 \).

One last comment is in order before leaving the dynamics of stirred tank reactors. We have assumed first order reactions throughout, primarily because they lead to sets of linear equations. If we must treat higher order reactions, the picture becomes very quickly extremely complicated. For example, suppose
that in Section 18.2.1 a second order reaction occurs in the isothermal reactor, so that

\[ r_A = kC_A^2. \]  

(18-36)

The mass balance for component A now becomes

\[ \frac{dC_A}{dt} = \frac{Q C_A}{V} - \frac{Q C_A}{V} - kC_A^2 V. \]  

(18-37)

This nonlinear Riccati equation (see Chapter 15) is not, in general, amenable to analytic solution. Whether linearization gives a reasonable solution depends very largely upon the type of nonlinearity and the purpose to which we intend to put the solution. We have, however, managed by using first order reactions to gain some qualitative insight into several fundamental characteristics of tank reactors.

18.3. the dynamics of an ideal pipe reactor

18.3.1. isothermal reaction

We now investigate the ideal pipe reactor, which we regard for our purposes as being modelled by a single pipe, Fig. 18.4. Just as with the tank reactor, a substance A reacts with another substance, present in excess, to give a product B. The following notation and assumptions are first recorded:

- the concentration of component A in the feed \( Q_0 \) (m\(^3\)/s) is \( C_A(0) = C_{A0} \) (mol/m\(^3\)), while in the flow \( Q \) leaving the reactor it is \( C_A(L) = C_{AL} \). The concentration of product B in \( Q \) is \( C_B(L) = C_{BL} \). The specific gravities of the reactants are assumed constant and there is no change in volume associated with the reaction, i.e. \( Q_0 = Q \).

- in the pipe reactor we assume that pure plug flow exists throughout, i.e.

complete mixing in the radial direction and no mixing in the axial direction. As a result of this assumption flat velocity and concentration profiles in the radial direction must be expected.

- the reaction is of the first order and irreducible, so that we can represent it as \( A \rightarrow^k B \), where \( k \) is the reaction velocity constant. The reaction rate, \( r_A \) (mol/m\(^3\)/s) is still given by
\[ \tau_A = kC_A \] (18-38)

and since the reaction is isothermal, \( k \) is constant.

- the length of the reactor is \( L \) (m) and its (constant) cross sectional area is \( \phi \) (m\(^2\)).

A component balance for \( A \) about the element \( dx \) (Fig. 18.4) a distance \( x \) from the start of the reactor gives, in analogy with eq. (18-2):

\[ \phi \frac{dC_A(x)}{dx} = Q \frac{dC_A(x)}{dx} - Q \frac{dC_A(x + dx)}{dx} - kC_A(x) \phi dx \] (18-39)

Substitution of \( C_A(x + dx) = C_A(x) + \frac{dC_A}{dx} dx \) (18-40) into (18-39) results in the partial differential equation (after division by \( \phi dx \)):

\[ \frac{\partial C_A}{\partial t} = -v \frac{\partial C_A}{\partial x} - kC_A \] (18-41)

where \( v \) represents the linear velocity of the material transport (\( v = Q/\phi \)). In the Laplace domain (see p. 22, Chapter 2) eq. (18-41) may be written as:

\[ s \bar{C}_A = -v \frac{d\bar{C}_A}{dx} - k\bar{C}_A \] (18-42)

which is an ordinary differential equation in \( x \). The solution of this equation is

\[ \bar{C}_A(x) = \bar{C}_{A_0} e^{-(s+k)x/v} \] (18-43)

or, in terms of the residence time \( \tau = x/v \) over the length \( x \) of the reactor

\[ \bar{C}_A(x) = \bar{C}_{A_0} e^{-s\tau} e^{-k\tau} \] (18-44)

As far as the overall mass balance is concerned, the reaction plays no role so that we can immediately write:

\[ \bar{C}_A(x) + \bar{C}_B(x) = \bar{C}_{A_0} e^{-s\tau} \] (18-45)

Combining eqs. (18-44) and (18-45) gives

\[ \bar{C}_B(x) = \bar{C}_{A_0} e^{-s\tau} (1 - e^{-k\tau}) \] (18-46)

We are now in a position to be able to write down the transfer functions of the input concentration \( C_{A_0} \) to the output concentrations \( C_{A_L} \) and \( C_{B_L} \). With \( x = L \) and substitution of the residence time \( \tau_v = L/v \) for the whole reactor it follows from eqs. (18-44) and (18-46) that

\[ \frac{\bar{C}_{A_L}}{\bar{C}_{A_0}} = e^{-s\tau_v} e^{-k\tau_v} \] (18-47)
and
\[
\frac{C_{BL}}{C_{A_0}} = e^{-st_v} (1 - e^{-kr_v}). \quad (18-48)
\]

In these transfer functions the term \(e^{-st_v}\) indicates the presence of a dead time \(\tau_v\), while \(e^{-kr_v}\) is an attenuation factor. It is often possible to describe the dynamics of equilibrium and consecutive reactions in a similar fashion to that just presented.

**Example 18.2.**

Consider the equilibrium reaction \(A \xrightarrow{k_1} B\). The reaction rate becomes
\[
r_A = k_1 C_A - k_2 C_B, \quad (18-49)
\]
while the component mass balance for substance \(A\) is (cf. eq. (18-42)):
\[
sC_A = -v \frac{dC_A}{dx} - k_1 C_A + k_2 C_B. \quad (18-50)
\]

The overall mass balance may be found by setting \(k_1\) and \(k_2\) zero in the above equation and replacing \(C_A\) by \(C_A + C_B\). The result is
\[
s(C_A + C_B) = -v \frac{d(C_A + C_B)}{dx}. \quad (18-51)
\]

Solving eqs. (18-50) and (18-51) in conjunction with the (known) boundary conditions gives the transfer functions:
\[
\frac{C_{AL}}{C_{A_0}} = e^{-st_v} \left( \frac{k_2}{k_1 + k_2} + \frac{k_1}{k_1 + k_2} e^{-(k_1 + k_2)\tau_v} \right) \quad (18-52)
\]
and
\[
\frac{C_{BL}}{C_{A_0}} = e^{-st_v} \frac{k_1}{k_1 + k_2} (1 - e^{-(k_1 + k_2)\tau_v}). \quad (18-53)
\]

The evaluation of the dynamics of a consecutive reaction is somewhat simpler.

**Q. 88:** If the reaction scheme for a given ideal pipe reactor is \(A \rightarrow B \rightarrow C\), what are the transfer functions \(\frac{C_{AL}}{C_{A_0}}, \frac{C_{BL}}{C_{A_0}}\) and \(\frac{C_{CL}}{C_{C_0}}\)?

**18.3.2. non-isothermal reaction**

In the previous section we considered only isothermal reactions; it is instructive now to look at exactly the same situation as in Section 18.3.1, but when an exothermic reaction is taking place. The reactor is supplied with a means of removing the excess liberated heat, as shown in Fig. 18.5.
Besides the points already noted in Section 18.3.1 we draw the reader’s attention to the following:

- the liberated heat of reaction is $Q$ (J/mol)
- the contents of the pipe reactor have a constant heat capacity $c$ (J/°C m$^3$)
- the heat transfer coefficient of the cooling circuit is $h$ (J/°C m$^2$s); moreover its area available for heat transfer is $A$ (m$^2$)
- the temperature of the cooling medium is overall $\theta_k$ (°C). The pipe reactor inlet temperature is $\theta(0) = \theta_0$, its outlet temperature $\theta(L) = \theta_L$ and the pure plug flow insures a flat temperature profile in the radial direction.

Just as for the tank reactor, the equations for $k$, $r_A$ and $\Delta r_A$ are

$$k = k_0 e^{E/(\theta + 273)}$$

$$r_A = k C_A = k_0 C_A e^{E/(\theta + 273)}$$

and

$$\Delta r_A = a_c \Delta C_A + a_\theta \Delta \theta.$$ (18-20)

The energy balance around an element $dx$ of the reactor a distance $x$ from the inlet is (cf. eq. (18-22)):

$$\phi dx c \frac{\partial \theta(x)}{\partial t} = Q c \theta(x) - Q c \theta(x + dx) + a \phi dx r_A(x) - h\phi dx \{\theta(x) - \theta_k\}$$ (18-54)

Substituting $\theta(x + dx) = \theta(x) + \frac{\partial \theta}{\partial x} dx$ in this equation and dividing throughout by $\phi dx c$ gives

$$\frac{\partial \theta}{\partial t} = -v \frac{\partial \theta}{\partial x} + \frac{a}{c} r_A - \frac{h}{c} (\theta - \theta_k),$$ (18-55)

wherein $v$ is the linear velocity of the material in the reactor ($= Q/\phi$). Since this equation is linear it also holds for small perturbations $\Delta \theta$ and $\Delta \theta_k$; this
form, in the Laplace domain, may be written:

\[ s \Delta \theta = -v \frac{d \Delta \theta}{dx} + \frac{a}{c} \Delta \rho - \frac{h}{c} (\Delta \rho - \Delta \theta_k). \]  

(18-56)

The overall and component A mass balances may also be written for perturbation variables. In analogy with eq. (18-42) the component balance is

\[ s \Delta C_A = -v \frac{d \Delta C_A}{dx} - \Delta \rho_A. \]  

(18-57)

The overall mass balance may once again be found by setting \( \Delta \rho_A \) zero in the above component balance and replacing \( \Delta C_A \) by \( \Delta C_A + \Delta C_B \):

\[ s (\Delta C_A + \Delta C_B) = -v \frac{d (\Delta C_A + \Delta C_B)}{dx}. \]  

(18-58)

Equations (18-20), (18-56), (18-57) and (18-58) are a set of four ordinary differential equations in \( x \), with four dependent variables \( \Delta \rho_A \), \( \Delta \theta \), \( \Delta C_A \) and \( \Delta C_B \), and the initial conditions \( \Delta C_A (0) = \Delta C_A_0 \), \( \Delta C_B (0) = 0 \) and \( \Delta \theta (0) = \Delta \theta_0 \).

The variables \( \Delta C_A_0 \), \( \Delta \theta_0 \) and \( \Delta \theta_k \) are determined by the particular problem being studied.

We can now derive the transfer functions from \( \Delta C_A_0 \), \( \Delta \theta_0 \) and \( \Delta \theta_k \) to \( \Delta C_A \), \( \Delta C_B \) and \( \Delta \theta \); we take as an example here the transfer function \( \frac{\Delta C_A}{\Delta C_A_0} \). To calculate this, we must set both \( \Delta \theta_0 \) and \( \Delta \theta_k \) to zero. Then substitution of eq. (18-20) into eq. (18-56) and eq. (18-57) and some rearrangement leads to the relationships

\[ v \frac{d \Delta \theta}{dx} + \left( s + \frac{h - a \alpha_\theta}{c} \right) \Delta \theta = \frac{a a_c}{c} \Delta C_A, \]  

(18-59)

and

\[ v \frac{d \Delta C_A}{dx} + (s + a_c) \Delta C_A = -a_\theta \Delta \theta. \]  

(18-60)

Elimination of \( \Delta \theta \) from these two equations gives:

\[ a_0 \frac{d^2 \Delta C_A}{dx^2} + a_1 \frac{d \Delta C_A}{dx} + a_2 = 0, \]  

(18-61)

where

\[ a_0 = v^2 \]  

(18-62)

\[ a_1 = v (2s + \frac{h - a a_\theta}{c} + a_c) \]  

(18-63)

\[ a_2 = \left( s + \frac{h - a a_\theta}{c} \right) (s + a_c) + \frac{a a_a \alpha_\theta}{c}. \]  

(18-64)

274
The two initial conditions are
\[ \Delta C_A(0) = \Delta C_{A_0} \]
and for eq. (18-60)
\[ \frac{d\Delta C_A}{dx} \bigg|_{x=0} = - \frac{s + a}{\nu} \Delta C_{A_0}. \]

The general solution of eq. (18-61) is, of course,
\[ \Delta C_A = D_1 e^\gamma_1 x + D_2 e^\gamma_2 x, \]
where \( \gamma_1 \) and \( \gamma_2 \) are the roots of the characteristic equation:
\[ a_0 \gamma^2 + a_1 \gamma + a_2 = 0 \]
and \( D_1, D_2 \) are found from the initial conditions. Usually \( D_1, D_2, \gamma_1 \) and \( \gamma_2 \) are functions of \( s \), while in the expressions for \( \gamma_1 \) and \( \gamma_2 \) terms having the form \( \sqrt{\text{function of } s} \) can appear. This, of course, does not make the task of interpreting eq. (18-67) any easier! Finally, substitution of \( x = L \) in eq. (18-67) gives \( \Delta C_A \) and hence the required transfer function.

Q. 89: Set up the energy balance describing an ideal pipe reactor wherein the consecutive reaction \( A \rightarrow B \rightarrow C \) is taking place. The reactions are first order, they proceed to completion, and liberate a heat of reaction.

Just as in Section 18.2 we have consistently used first order reactions in order to arrive at sets of linear differential equations. Even so, the solutions quickly become complicated when the energy balance has to be considered; it is not hard to imagine that with higher order reactions the picture becomes intractable. Still, we can state with some justification, just as in Section 18.2, that our analysis with first order reactions has increased our insight into the qualitative behaviour of pipe reactors.

18.4. non-ideal reactors

In Section 18.2 we found (eq. (18-4)) that the transfer function \( G_t(s) = \frac{C_A}{C_{A_0}} \) of an ideal tank reactor with an isothermal, first order reaction could be written in terms of a gain factor \( k_t \) and a time constant \( \tau_t \). The transfer function \( G_p(s) = \frac{C_A}{C_{A_0}} \) of an ideal pipe reactor with a first order, isothermal reaction can be written (cf. eq. (18-47) as the product of a gain factor \( k_p \) and a dead time, \( T_p \). For simplicity assume that \( k_t = k_p = k \) and that \( \tau_t = T_p = \tau \), so that
\[ G_t(s) = \frac{k}{1 + s\tau} \]
and

\[ G_p(s) = k e^{-st}. \]  

(18-70)

Now consider the transitional transfer function \( G_n(s) \), between \( G_t(s) \) and \( G_p(s) \), defined by

\[ G_n(s) = \frac{k}{\left(1 + \frac{st}{n}\right)^n}, \quad n = 1, 2, \ldots \]  

(18-71)

We see that \( G_1(s) = G_t(s) \), while from our knowledge of transfer functions (p. 44, Chapter 4) it is clear that \( G_\infty(s) = G_p(s) \). In Fig. 18.6 the step response, \( x_n(t) \), of \( G_n(s) \) is shown for \( n = 0, \infty \) and a value of \( n \) lying in between.

![Fig. 18.6. Step response of \( G_n(s) \).](image)

It is apparent from the figure that the most general form of the step response of \( G_n(s) \) for \( 1 < n < \infty \) will be an S-shape. This just happens to be the measured response of many real (as distinct from ideal) reactors, both of the tank and pipe variety. Thus the real reactor lies in general, as far as its dynamic behaviour goes, between the ideal tank reactor and the ideal pipe reactor, and can be represented by a transfer function of the type \( G_n(s) \), with \( 1 < n < \infty \).

The non-ideal mixing, which is at the heart of the non-ideal behaviour of real tank reactors, can be put down to various factors such as the viscous properties of the reactor contents, the design of the tank and mixer and the position of the feed and product lines.

The non-ideal behaviour of real pipe reactors can be explained as the result of axial mixing and radial gradients. Let us now consider real pipe reactors in some detail.
Up till now a pure plug flow has been assumed to exist throughout the pipe reactor. In practice a certain amount of axial transport of mass and energy will take place as a result of concentration and temperature gradients being established, and possibly due also to turbulence. This transportation can often be thought of as being the result of a large number of events taking place on a microscopic scale—just as with packed bed reactors. The dynamic behaviour of the real pipe reactor can then be sufficiently accurately described by extending the mass balance of the ideal reactor with an extra diffusion term and its energy balance with an extra thermal conductivity term. In this way we obtain from eqs. (18-41) and (18-55) the following modified model for a pipe reactor with axial mixing

\[
\frac{\partial C_A}{\partial t} = -v \frac{\partial C_A}{\partial x} - r_A + \frac{\partial^2 C_A}{\partial x^2} \quad (18-72)
\]

\[
\frac{\partial \theta}{\partial t} = -v \frac{\partial \theta}{\partial x} + \frac{a}{c_r A} \frac{\partial \theta}{\partial r} + \frac{1}{c_r} (\theta - \theta_k) + \frac{\lambda_a}{c_r} \frac{\partial^2 \theta}{\partial r^2} \quad (18-73)
\]

where \(\Gamma_a\) and \(\lambda_a\) represent the coefficients of effective diffusion and effective thermal conductivity in the axial direction, respectively. The word “effective” is used to underline the fact that these coefficients include not only molecular but also turbulent diffusion and conduction effects. The units of \(\Gamma_a\) are \((\text{m}^2/\text{s})\) and of \(\lambda_a\) \((\text{J/m}\cdot\text{s}\cdot\text{C})\).

With strongly exothermic reactions the radial concentration and temperature gradients can no longer be ignored. Thus, whenever large quantities of heat must be removed from the centre of the pipe reactor to the coolant surrounding the pipe, large radial temperature gradients can be expected. Since the reaction rate is temperature dependent there must also be a corresponding radial concentration gradient.

To give these ideas some mathematical form, imagine a round pipe reactor with a radius \(r\) in the \(y\)-coordinate direction and effective radial diffusion and thermal conductivity coefficients \(\Gamma_r\) and \(\lambda_r\). Again from eqs. (18-41) and (18-55) we obtain the following modified pipe reactor model with radial gradients:

\[
\frac{\partial C_A}{\partial t} = -v \frac{\partial C_A}{\partial x} - r_A + \frac{\partial^2 C_A}{\partial y^2} + \frac{1}{y} \frac{\partial C_A}{\partial y} \quad (18-74)
\]

\[
\frac{\partial \theta}{\partial t} = -v \frac{\partial \theta}{\partial x} + \frac{a}{c_r A} \frac{\partial \theta}{\partial r} + \frac{\lambda_r}{c_r} \frac{\partial^2 \theta}{\partial y^2} + \frac{1}{y} \frac{\partial \theta}{\partial y} \quad (18-75)
\]

For packed bed reactors it should be noted that the coefficients \(\Gamma_a\) and \(\lambda_a\) are about ten times as large as the radial coefficients \(\Gamma_r\) and \(\lambda_r\). Of course, it is possible to set up a model which accounts for both axial mixing and radial gradients; however, we will keep the story simple and not pursue that particular line further, here.

The above models, represented by eqs. (18-72), (18-73) and (18-74) only permit analytical solutions in very special cases. For example, suppose that we wish to find the dynamic response of a pipe reactor having axial mixing
and a first order, isothermal, reaction. Radial gradients, we are told, may be neglected. In this special case it is only necessary to consider the mass balance given by eq. (18-72); when transformed into the s-domain this gives the second-order, linear, differential equation

\[ sC_A^\prime - v \frac{dC_A}{dx} - kC_A + D_a \frac{d^2C_A}{dx^2} = 0, \]  

(18-76)

where \( k \) is the reaction velocity constant.

Q. 90: The initial conditions applicable to eq. (18-76) are \( \frac{dC_A}{dx} \bigg|_{x=0} = 0 \) and \( \frac{dC_A}{dx} \bigg|_{x=0} = C_A(0) = C_A_0 \). What is the solution, \( C_A(x) \)?

From the solution \( C_A(x) \) to eq. (18-76) we are in a position to find the transfer function from \( C_A \) to \( C_A^L \). This turns out to be

\[ \frac{C_A^L}{C_A_0} = e^{\frac{-1}{2}Pe} \left[ e^{2C_0\sqrt{1+4(s+k)^2}Pe} + e^{\frac{-1}{2}Pe\sqrt{1+4(s+k)^2}\tau_v/Pe} \right], \]

(18-77)

where \( Pe = \frac{vL}{ID} \) is the Peclet number. Making use of a table of transforms, the impulse response (p. 59, Chapter 5) of this special pipe reactor can be seen to be:

\[ g_p(t) = \tau_v \frac{\tau_v Pe}{2t} \sqrt{\frac{\tau_v Pe}{\pi t}} e^{-\frac{1}{4}P\left(\frac{t^2}{\tau_v^2}+\frac{1}{\tau_v^2}\right)} - \frac{1}{4}kt. \]

(18-78)

In Fig. 18.7 this impulse response (weighting function) is drawn for \( Pe = 1 \) and \( Pe = 100 \).

Fig. 18.7. The impulse response of a pipe reactor with axial mixing.

It can be seen that for large Peclet numbers \( g_p(t) \) is very nearly the same as would be expected from an ideal pipe reactor (pure dead time — compare with Fig. 5.1, p. 58); for small Peclet numbers — corresponding to much diffus-
18.5. the control of chemical reactors

The objectives most frequently encountered in the control of chemical reactors are the maintenance of steady temperatures and steady concentrations. In Fig. 18.8 a typical temperature control scheme for a tank reactor is shown.

![Fig. 18.8. Temperature control of a tank reactor.](image)

The reactor itself is surrounded by a mantle which contains circulating coolant. The temperature control loop is supplemented with feedforward control of the most important disturbance source — the load of the reactor. This then, is an example of the feedback-feedforward control discussed in Chapter 10.

Suppose that the tank reactor and cooling circuit can be described by the model that was developed in Section 18.2 for an ideal tank reactor wherein a first order, exothermic chemical reaction takes place. If the controller action is the proportional mode, then

$$\Delta \theta_k = -k \Delta \theta. \quad (18-79)$$

Q. 91: What, in analogy to eq. (18-26), is the new transfer function $\Delta \theta/\Delta C_{\Lambda_0}$, if we take into account proportional temperature control, eq. (18-79)?
Now in exactly the same way as previously, Section 18.2, the two conditions for the stable dynamic operation of the closed loop reactor can be derived. Similarly to eqs. (18-28) and (18-29) we find that the conditions are

I) \[ 1 - \frac{a}{c} \tau_1' \alpha_0 + \tau_v \alpha_c > 0 \] (18-80)

II) \[ \tau_1' + \tau_v - \frac{a}{c} \alpha_0 \tau_1' \tau_v + \alpha_c \tau_1' \tau_v > 0, \] (18-81)

where the only difference, \( \tau_1' \), is given by (cf. eq. (18-27))

\[ \tau_1' = \frac{\tau_1}{1 + k}. \] (18-82)

From eqs. (18-80) and (18-81) it is apparent that reactor stability can always be ensured by making \( k \), the proportional gain, large enough.

We now turn to the control of concentrations in a tank reactor. Often, one of the components, \( A \), of the chemical reaction is needed in smaller quantity than the other reactants. When this is the case the concentration control scheme shown in Fig. 18.9 is frequently encountered.

![Diagram of concentration control of a tank reactor](image)

**Fig. 18.9.** Concentration control of a tank reactor.

**Q. 92:** What means are available for measuring the concentration of the product? What does the symbol \( AC \) of Fig. 18.9 represent?

From Section 18.2 it can be shown that the response of the unreacted component in the product stream is faster than the response of the product itself (first- and second-order responses, respectively) so that it is wiser to measure, for control
purposes, the concentration of the unreacted feed in the product stream rather than the concentration of the product itself.

Q. 93: Using eqs. (18-4) and (18-8) prove that the above statement is true.

If both concentration and temperature control is applied, then there is the possibility that the interaction resulting from the coupling between concentration and temperature can give problems. These problems may be reduced by designing a decoupling controller, for which the reader is referred to Chapters 10 and 13.

The control of distributed parameter systems, of which the pipe reactor is a classic example, is a far more complicated problem than the control of tank reactors. One reason for this is that the mathematical models used to describe the dynamical behaviour of these systems are in general very complicated. Another reason is that pure time delays go hand in hand with distributed systems and the control of dead time elements, we already know, is a difficult problem. We will content ourselves here with one or two general remarks concerning the control of pipe reactors.

As just pointed out, the pipe reactor is difficult to control because of the dead times and other delaying effects. On the other hand, it should be possible by introducing optimal or near optimal axial temperature profiles to significantly improve the yield of the reactor. The application of temperature profiles would, of course, bring with it additional control hardware such as apart temperature control loops for various parts of the reactor and additional temperature sensing locations. In practice the extra costs associated with the above would have to be balanced against the profitability of the extra yield attainable.

18.6. references and bibliographical notes

The basic ideas of the control and dynamic behaviour of chemical reactors can be found in the books by


A restricted, but very clear survey of models and especially control schemes appears in the book by


For models of non-ideal pipe reactors the reader is referred to


Recommended references for the treatment of reactor stability are:

For a mathematical treatment of chemical reactors in a modern vein, we recommend:


APPENDIX C
THE EXTENSION OF THE EULER-LAGRANGE METHOD OF
DYNAMIC OPTIMIZATION TO MULTIDIMENSIONAL CONTROL
PROCESSES

Tou [1] suggests that it is possible to extend the Euler-Lagrange method
of unconstrained dynamic optimization to multidimensional control processes.
Even for the elementary case of a linear system and a quadratic performance
index (LQ for short) many authors [2] to [6] do not state that in general
this is impossible. We show here that, for the LQ problem, only an unusual
case which is not likely often to be encountered in practice admits a solution.
Unfortunately this unrepresentative example is used by some authors [2], [6]
to illustrate the Euler-Lagrange method.
The problem statement is as follows.
Given the system

\[ \dot{x} = Ax + Bu \]  

(C-1)

and initial conditions

\[ x(t_0) = x_0 \]  

(C-2)

minimize the cost functional

\[ \int_{t_0}^{t_f} \{x^T Q x + u^T R u\} \, dt \]  

(C-3)

by using the Euler-Lagrange method. We assume that \( \text{dim} (x) = n \) and
\( \text{dim} (u) = r \).

Using the usual devices of variational calculus (e.g. see p. 213 [1]) we de­
define two new time functions, \( \eta(t) \) and \( \xi(t) \) such that

\[ \dot{x} = x^0 + \varepsilon \eta; \quad \text{dim} (\eta) = n \]

and

\[ \dot{u} = u^0 + \varepsilon \xi; \quad \text{dim} (\xi) = r, \]

where \( \varepsilon \) is a scalar constant and

\[ \eta(t_0) = \eta(t_f) = 0 \]  

(C-4)

\[ \xi(t_0) = \xi(t_f) = 0. \]  

(C-5)

Expanding the performance index function, \( F \), by a Taylor series gives:

\[ F(x, u) = F(x^0, u^0) + \varepsilon x^T Q x + \varepsilon u^T R u + \ldots \]  

(C-6)

where we've assumed \( Q \) and \( R \) to be symmetric. For \( J(\varepsilon) \) to be an extremal
which we assume here is the global minimum \( \frac{\partial J(\varepsilon)}{\partial \varepsilon} \bigg|_{\varepsilon=0} \) must be zero, so from eqs. (C-3) and (C-6) we have the condition for a minimum that

\[
0 = \int_{t_0}^{t_f} \{\dot{x}^0 T Q \dot{\eta} + \dot{u}^0 T R \dot{\xi}\} dt. \tag{C-7}
\]

Suppose now that a transformation is possible such that

\[
R = \Gamma B \tag{C-8}
\]

where \( \Gamma \) is a \((r \times n)\) transformation matrix. Then from eqs. (C-1), (C-7) and (C-8) we may write

\[
0 = \int_{t_0}^{t_f} \{\dot{x}^0 T Q \eta + \dot{u}^0 T \Gamma \eta - \dot{u}^0 T \Gamma A \eta\} dt. \tag{C-9}
\]

Integrating the second term by parts:

\[
\int_{t_0}^{t_f} \dot{u}^0 T \Gamma \eta dt = -\int_{t_0}^{t_f} \dot{u}^0 T \Gamma \eta dt. \tag{C-10}
\]

We have used eq. (C-4) to obtain this result. It follows that

\[
\int_{t_0}^{t_f} \{\dot{x}^0 T Q - \dot{u}^0 T \Gamma A - \dot{u}^0 T \Gamma\} \eta dt = 0. \tag{C-11}
\]

For this equation to be satisfied for all \( \eta \) it follows that

\[
\dot{x}^0 T Q - \dot{u}^0 T \Gamma A - \dot{u}^0 T \Gamma = 0. \tag{C-12}
\]

This Euler-Lagrange equation must now be solved for \( \dot{u} \) if we are to find the control policy. Taking Laplace transforms:

\[
\dot{x}^0 T Q - \dot{u}^0 T (\Gamma A - s \Gamma) + \dot{u}^0 T (t_0) \Gamma = 0. \tag{C-13}
\]

Any solution which is non-trivial requires that \(|\Gamma A - s \Gamma| \neq 0\). However, since \( \Gamma \) is \((r \times n)\) this is only possible if \( n = r \).

Also, since \(|\Gamma A - s \Gamma| = |\Gamma||A - s I| \neq 0\)

\(|\Gamma| \neq 0, \ i.e. \)

\( \text{rank } \Gamma = n. \)

Using these results in eq. (C-8), it follows that \( \text{rank } R = \text{rank } B. \)

The results of this analysis can be summarized as follows. For a linear process and a quadratic performance index we must be sure that
(i) \[ n = r \]
(ii) \[ \text{rank } R = \text{rank } B \]

before the Euler-Lagrange equations may be applied.

references Appendix C


APPENDIX D
SOME REMARKS CONCERNING VECTORS AND MATRICES

We saw in Appendix B that for a stochastic, scalar signal \( w(t) \):

\[
\text{expectation } \eta(t) = E\{w(t)\} = \int_{-\infty}^{\infty} wp(w,t)dw
\]

and

\[
\text{variance } \sigma^2(t) = E\{(w(t) - \eta(t))^2\}.
\]

The equivalent expression for a vector of stochastic signals, \( \mathbf{w}(t) \), having zero mean values is

\[
\text{variance matrix } E\{\mathbf{w}(t)\mathbf{w}^T(t)\} = \begin{bmatrix}
E\{e_1^2\} & \ldots & E\{e_1e_n\} \\
\vdots & \ddots & \vdots \\
E\{e_ne_1\} & \ldots & E\{e_n^2\}
\end{bmatrix}
\]

while the sum of the diagonal elements of the variance matrix (its trace) is given by

\[
E\{\mathbf{w}^T(t)\mathbf{w}(t)\} = E\{e_1^2\} + \ldots + E\{e_n^2\}.
\]

Note that the variance matrix of \( \mathbf{w}(t) \) is symmetrical. If \( \mathbf{w}(t) \) and \( \mathbf{v}(t) \) are uncorrelated, then \( E\{\mathbf{w}(t)\mathbf{v}^T(t)\} = 0 \).

A few definitions concerning a matrix \( A \) and a vector \( x \) are now given.

- \( A \) is real if and only if all the elements of \( A \) are real
- \( A \) is symmetric when \( A = A^T \)
- \( A \) is positive definite if \( x^TAx > 0 \) for all \( x \neq 0 \)
- A is semi-positive definite if \( x^T A x \geq 0 \) for all \( x \neq 0 \).

These last two conditions are sometimes written shorthand as \( A > 0 \) and \( A \geq 0 \).
A is non-negative definite if $\mathbf{x}^T A \mathbf{x} \geq 0$ for all $\mathbf{x} \neq \mathbf{0}$. These last two conditions are sometimes written shortened as $A \succeq 0$ and $A \succeq 0$. 