CHAPTER 2

Preliminary Material

2.1 Introduction

This chapter describes the modelling and analysis tools that will be used throughout the book. They all concern Linear Time Invariant systems or transfer functions and can be classified in three categories.

Closed-loop system representation and analysis: We briefly define two possible representations of a system G_0 in closed loop with some stabilising controller K. The first one is a general representation based on a standard block-diagram description of the closed loop (Section 2.2) and on which we shall base the notions of closed-loop stability and generalised stability margin. The second one is based on a linear fractional transformation (LFT) (Section 2.3), which is sometimes more convenient for the manipulation of multivariable closed-loop systems or for robust control design and analysis. We show, in Section 2.3, that any closed-loop system represented by means of an LFT can be put in the general form defined in Section 2.2, allowing use of standard stability analysis tools.

Linear systems analysis: Two important tools are presented. The first one is coprime factorisation (Section 2.4). A coprime factorisation is a way of representing a possibly unstable transfer function or matrix by two stable ones (a numerator and a denominator) with particular properties related, *e.g.*, to closed-loop stability. Procedures are given to build such (possibly normalised) factorisations. It is also shown how the generalised closed-loop transfer matrix of a system can be expressed in function of the plant and controller coprime factors. The second tool is the ν-gap metric between two transfer functions (Section 2.5). It is a control-oriented measure of the distance between two transfer functions or matrices, of great importance for robustness analysis.

System modelling: The two classical black-box modelling tools that we consider in this book are prediction-error identification and balanced truncation.

The first one, described in Section 2.6, uses data measured on the actual plant to compute the best model of this plant in some set of parametrised transfer functions, with respect to a criterion that penalises the prediction errors attached to this model. The second one, described in Section 2.7, is a tool aimed to reduce the order of a given linear high-order model of the plant (obtained, for instance, by identification, first-principles-based physical modelling, finite-element modelling, or linearisation of a nonlinear simulator) to derive a lower order model, or of any high-order transfer function, by discarding the least controllable and observable modes. It will be used to design low-order controllers for high-order processes.

2.2 General Representation of a Closed-loop System and Closed-loop Stability

2.2.1 General Closed-loop Set-up

Let us consider a (possibly multi-input multi-output) linear time-invariant system described by

$$y(t) = G_0(z)u(t) + v(t) \quad \text{or} \quad y(t) = G_0(s)u(t) + v(t) \tag{2.1}$$

where $G_0(z)$ (discrete-time case) or $G_0(s)$ (continuous-time case) is a rational transfer function or matrix. Here, z^{-1} is the backward shift operator[1] ($z^{-1}x(t) = x(t-1)$ with t expressed in multiples of the sampling period) and s is the time differentiation operator ($sx(t) = \dot{x}(t)$). $u(t)$ is the input of the system, $y(t)$ its output, $v(t)$ an output disturbance.

We shall often consider the representation of Figure 2.1 when the plant G_0 operates in closed loop with a controller K. In this representation, $r_1(t)$ and $r_2(t)$ are two possible sources of exogenous signals (typically, $r_1(t)$ will be a reference or set-point signal for $y(t)$, while $r_2(t)$ will be either a feed-forward control signal or an input disturbance). $g(t)$ and $f(t)$ denote respectively the input and the output of the controller.

[1]The time-domain backward shift operator used in discrete-time systems is often represented by q^{-1} in the literature, while the notation z is generally used for the corresponding frequency-domain Z-transform variable. Here, for the sake of simplicity and although mathematical rigour would require such distinction, we shall use the same notation for both the operator and the variable. Similarly, the time differentiation operator used in continuous-time systems is often represented by p, but we shall make no distinction between it and the frequency-domain Laplace-transform variable s.

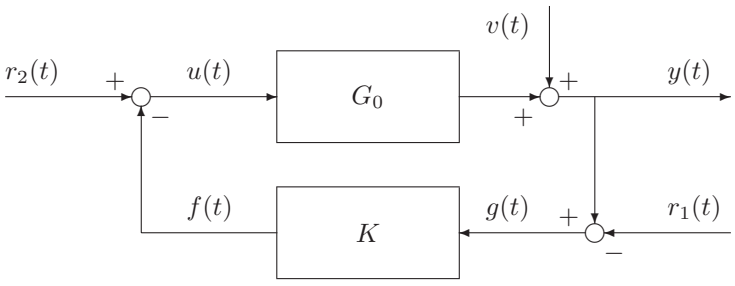

Figure 2.1. General representation of a system in closed loop

2.2.2 Closed-loop Transfer Functions and Stability

In mainstream robust control, the following generalised closed-loop transfer matrices are often considered[2]:

$$\begin{aligned} T_i(G_0,\, K) &= \begin{bmatrix} -I & G_0 \\ K & I \end{bmatrix}^{-1} + \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} G_0(I+KG_0)^{-1}K & G_0(I+KG_0)^{-1} \\ (I+KG_0)^{-1}K & (I+KG_0)^{-1} \end{bmatrix} \end{aligned} \tag{2.2}$$

and

$$\begin{aligned} T_o(G_0,\, K) &= \begin{bmatrix} -I & K \\ G_0 & I \end{bmatrix}^{-1} + \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} K(I+G_0K)^{-1}G_0 & K(I+G_0K)^{-1} \\ (I+G_0K)^{-1}G_0 & (I+G_0K)^{-1} \end{bmatrix} \end{aligned} \tag{2.3}$$

The entries of $T_i(G_0,\, K)$ are the transfer functions between the exogenous reference signals and the input and output signals of the plant defined in Figure 2.1:

$$\begin{bmatrix} y(t) \\ u(t) \end{bmatrix} = T_i(G_0,\, K) \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} + N_i(G_0,\, K)v(t) \tag{2.4}$$

where

$$N_i(G_0,\, K) = \begin{bmatrix} (I+G_0K)^{-1} \\ -(I+KG_0)^{-1}K \end{bmatrix} \tag{2.5}$$

(the entry $(I+G_0K)^{-1}$ is called the *closed-loop sensitivity function* of the system), while those of $T_o(G_0,\, K)$ are the transfer functions between the exogenous reference signals and the output and input signals of the controller:

$$\begin{bmatrix} f(t) \\ g(t) \end{bmatrix} = T_o(G_0,\, K) \begin{bmatrix} r_2(t) \\ -r_1(t) \end{bmatrix} + N_o(G_0,\, K)v(t) \tag{2.6}$$

[2]All transfer functions and matrices must be understood as rational functions of z (discrete-time case) or s (continuous-time case). However, when no confusion is possible, we shall often omit these symbols to ease the notations.

where

$$N_o(G_0,\, K) = \begin{bmatrix} K(I + G_0K)^{-1} \\ (I + G_0K)^{-1} \end{bmatrix} \tag{2.7}$$

In the SISO case, we define more simply

$$T(G_0,\, K) = \begin{bmatrix} \dfrac{G_0K}{1+G_0K} & \dfrac{G_0}{1+G_0K} \\ \dfrac{K}{1+G_0K} & \dfrac{1}{1+G_0K} \end{bmatrix} \triangleq \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \tag{2.8}$$

and

$$N(G_0,\, K) = \begin{bmatrix} \dfrac{1}{1+G_0K} \\ \dfrac{-K}{1+G_0K} \end{bmatrix} = \begin{bmatrix} T_{22} \\ -T_{21} \end{bmatrix} \triangleq \begin{bmatrix} S \\ -KS \end{bmatrix} \triangleq \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} \tag{2.9}$$

so that

$$\begin{bmatrix} y(t) \\ u(t) \end{bmatrix} = T(G_0,\, K) \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} + N(G_0,\, K)v(t) \tag{2.10}$$

Definition 2.1. (INTERNAL STABILITY) *The closed loop* $(G_0,\, K)$ *of Figure 2.1 is called 'internally stable' if all four entries of* $T_i(G_0,\, K)$ *or, equivalently, all four entries of* $T_o(G_0,\, K)$*, are stable, i.e., if they belong to* $\mathcal{H}_\infty$*.*

The generalised stability margin $b_{G_0,K}$ is an important measure of the internal stability of the closed loop. It is defined as

$$b_{G_0,K} \triangleq \begin{cases} \|T_i(G_0,\, K)\|_\infty^{-1} & \text{if } (G_0,\, K) \text{ is stable} \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

Note that $\|T_i(G_0,\, K)\|_\infty = \|T_o(G_0,\, K)\|_\infty$, as shown in (Georgiou and Smith, 1990). An alternative definition, in the SISO case, is the following:

$$b_{G_0,K} = \min_\omega \kappa \left(G_0(e^{j\omega}),\, \frac{-1}{K(e^{j\omega})} \right) \tag{2.12}$$

where $\kappa \left(G_0(e^{j\omega}),\, \frac{-1}{K(e^{j\omega})} \right)$ is the chordal distance at frequency ω between G_0 and $\frac{-1}{K(e^{j\omega})}$, as defined in (Vinnicombe, 1993*a*): see Section 2.5.

The margin $b_{G_0,K}$ plays an important role in robust optimal control design. The following results hold in the SISO case (Vinnicombe, 1993*b*):

$$\text{gain margin} \geq \frac{1 + b_{G_0,K}}{1 - b_{G_0,K}} \tag{2.13}$$

and

$$\text{phase margin} \geq 2 \arcsin\left(b_{G_0,K}\right) \tag{2.14}$$

Note that there is a maximum attainable value of the generalised stability margin $b_{G_0,K}$ over all controllers stabilising G_0 (Vinnicombe, 1993*b*):

$$\sup_K b_{G_0,K} = \sqrt{1 - \lambda_{max}(\mathbf{P}_{rcf}\mathbf{Q}_{rcf})} = \sqrt{1 - \left\| \begin{bmatrix} N_0 \\ M_0 \end{bmatrix} \right\|_H^2} \tag{2.15}$$

and (Georgiou and Smith, 1990)

$$\sup_K b_{G_0,K} \leq \inf_{\substack{z \in \mathbb{D}_1^+ \\ \text{or} \\ s \in \mathbb{C}_0^+}} \underline{\sigma}\left(\begin{bmatrix} N_0 \\ M_0 \end{bmatrix} \right) \tag{2.16}$$

Here, $\begin{bmatrix} N_0 \\ M_0 \end{bmatrix}$ is a normalised right coprime factorisation of G_0 (see Section 2.4). $\mathbf{P}_{rcf}$ and $\mathbf{Q}_{rcf}$ are, respectively, the controllability and observability Gramians (see Section 2.7) of the right coprime factors $\begin{bmatrix} N_0 \\ M_0 \end{bmatrix}$. $\|\cdot\|_H$ denotes the Hankel norm. It should be clear that it is easier to design a stabilising controller for a system with a large $\sup_K b_{G_0,K}$ than for a system with a small one.

We refer the reader to (Zhou and Doyle, 1998) and references therein for more detail about the links between the generalised stability margin and controller performance.

The following proposition will be used several times throughout this book.

Proposition 2.1. (Anderson *et al.*, 1998) *Let* (G_0, K) *and* (G, K) *be two stable closed-loop systems such that*

$$\|T_i(G, K) - T_i(G_0, K)\|_\infty < \varepsilon \tag{2.17a}$$

or, equivalently,

$$\|T_o(G, K) - T_o(G_0, K)\|_\infty < \varepsilon \tag{2.17b}$$

for some $\varepsilon > 0$. *Then,*

$$|b_{G,K} - b_{G_0,K}| < \varepsilon \tag{2.18}$$

It tells us that the stability margin achieved by a controller K connected to a system G will be close to that achieved by the same K with G_0 if the closed-loop transfer matrices $T_i(G, K)$ and $T_i(G_0, K)$ are close in the $\mathcal{H}_\infty$ norm.

2.2.3 Some Useful Algebra for the Manipulation of Transfer Matrices

The following relations are very useful when it comes to manipulating closed-loop transfer matrices.

A. Block-matrix inversion.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} \quad (2.19)$$

provided A and D are square, and A and $(D - CA^{-1}B)$ are invertible.

B. Other formulae.

$$A(I + BA)^{-1} = (I + AB)^{-1}A \quad (2.20a)$$

$$\begin{aligned} BA(I + BA)^{-1} &= (I + BA)^{-1}BA \\ &= B(I + AB)^{-1}A \\ &= I - (I + BA)^{-1} \end{aligned} \quad (2.20b)$$

2.3 LFT-based Representation of a Closed-loop System

In the MIMO case, it is often easier to represent a closed-loop system using Linear Fractional Transformations, as depicted in Figure 2.2, where Γ_0 is called the generalised plant and Q the generalised controller.

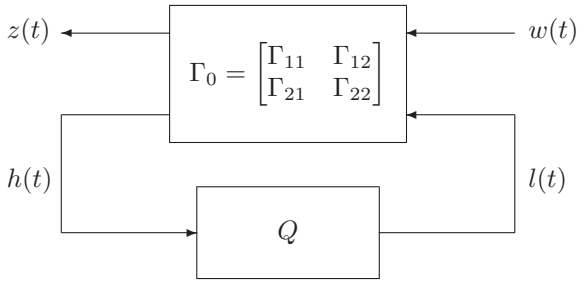

Figure 2.2. LFT representation of a system in closed loop

In this representation, all exogenous signals are contained in $w(t)$. $l(t)$ is the control signal and $h(t)$ is the input of the controller Q, *e.g.*, $l(t) = f(t)$ and $h(t) = g(t)$ if $Q = K$. $z(t)$ contains all inner signals that are useful for the considered application. For instance, if the objective is to design a control law for the tracking of the reference signal $r_1(t)$, $z(t)$ will typically contain the tracking error signal $g(t) = y(t) - r_1(t)$ and, possibly, the control signal $f(t)$ if the control energy is penalised by the control law.

The closed-loop transfer function between $w(t)$ and $z(t)$ is given by

$$T_{zw} = \mathcal{F}_l(\Gamma_0, Q) \triangleq \Gamma_{11} + \Gamma_{12}Q(I - \Gamma_{22}Q)^{-1}\Gamma_{21} \quad (2.21)$$

This representation can easily be transposed into the standard one of Figure 2.1 as we now show by means of three examples.

► **Example 2.1.** A single-degree-of-freedom controller K is used, and the signals of interest are the tracking error $g(t)$ and the control signal $f(t)$. Let us set

$$
\begin{aligned}
Q &= K \\
z(t) &= \operatorname{col}\big(f(t),\, g(t)\big) & w(t) &= \operatorname{col}\big(r_2(t),\, -r_1(t),\, v(t)\big) \\
h(t) &= g(t) & l(t) &= f(t) \\
\Gamma_{11} &= \begin{bmatrix} 0 & 0 & 0 \\ G_0 & I & I \end{bmatrix} & \Gamma_{12} &= \begin{bmatrix} I \\ -G_0 \end{bmatrix} \\
\Gamma_{21} &= \begin{bmatrix} G_0 & I & I \end{bmatrix} & \Gamma_{22} &= \begin{bmatrix} -G_0 \end{bmatrix}
\end{aligned}
$$

in Figure 2.2. Then,

$$T_{zw}(\Gamma_0,\, Q) = \begin{bmatrix} T_o(G_0,\, K) & N_o(G_0,\, K) \end{bmatrix}$$

◄

► **Example 2.2.** A single-degree-of-freedom controller K is used, and the signals of interest are the output $y(t)$ and the control signal $u(t)$. Let us consider Figure 2.2 and set

$$
\begin{aligned}
Q &= K \\
z(t) &= \operatorname{col}\big(y(t),\, u(t)\big) & w(t) &= \operatorname{col}\big(r_1(t),\, r_2(t),\, v(t)\big) \\
h(t) &= g(t) & l(t) &= f(t) \\
\Gamma_{11} &= \begin{bmatrix} 0 & G_0 & I \\ 0 & I & 0 \end{bmatrix} & \Gamma_{12} &= \begin{bmatrix} -G_0 \\ -I \end{bmatrix} \\
\Gamma_{21} &= \begin{bmatrix} -I & G_0 & I \end{bmatrix} & \Gamma_{22} &= \begin{bmatrix} -G_0 \end{bmatrix}
\end{aligned}
$$

In this case,

$$T_{zw}(\Gamma_0,\, Q) = \begin{bmatrix} T_i(G_0,\, K) & N_i(G_0,\, K) \end{bmatrix}$$

◄

► **Example 2.3.** A two-degree-of-freedom controller $C = \begin{bmatrix} K & F \end{bmatrix}$ is used and the signals of interest are the output $y(t)$ and the control signal $u(t)$. The control law is

$$u(t) = Fr_2(t) - Kg(t)$$

If we set

$$\begin{aligned} Q = C &= \begin{bmatrix} K & F \end{bmatrix} \\ z(t) &= \operatorname{col}\big(y(t),\, u(t)\big) & w(t) &= \operatorname{col}\big(v(t),\, r_1(t),\, r_2(t)\big) \\ h(t) &= \operatorname{col}\big(-g(t),\, r_2(t)\big) & l(t) &= u(t) \\ \Gamma_{11} &= \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \Gamma_{12} &= \begin{bmatrix} G_0 \\ I \end{bmatrix} \\ \Gamma_{21} &= \begin{bmatrix} -I & I & 0 \\ 0 & 0 & I \end{bmatrix} & \Gamma_{22} &= \begin{bmatrix} -G_0 \\ 0 \end{bmatrix} \end{aligned}$$

we find that

$$T_{zw}(\Gamma_0,\, Q) = \begin{bmatrix} N_i(G_0,\, K) & T_i(G_0,\, K) \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix} \end{bmatrix}$$

Observe that it is possible to rewrite

$$T_i(G_0,\, K) \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & I \end{bmatrix} T_i \left(\begin{bmatrix} G_0 \\ 0 \end{bmatrix},\, \begin{bmatrix} K & F \end{bmatrix} \right) \begin{bmatrix} I & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix}$$

with 0 matrices of appropriate dimensions, which can be convenient if one desires to treat $\begin{bmatrix} K & F \end{bmatrix}$ as a single object C, for instance if F and K share a common state-space representation. ◀

2.4 Coprime Factorisations

2.4.1 Coprime Factorisations of Transfer Functions or Matrices

We shall often use left and right coprime factorisations of systems and controllers in the sequel. Here, we define formally the notion of coprimeness and we explain how to compute the coprime factors of a dynamic system. Details can be found in, *e.g.*, (Francis, 1987), (Vidyasagar, 1985, 1988) and (Varga, 1998).

Definition 2.2. (Coprimeness) *Two matrices $\tilde{M}$ and $\tilde{N}$ in $\mathcal{RH}_\infty$ are left coprime over $\mathcal{RH}_\infty$ if they have the same number of rows and if there exist matrices X_l and Y_l in $\mathcal{RH}_\infty$ such that*

$$\begin{bmatrix} \tilde{M} & \tilde{N} \end{bmatrix} \begin{bmatrix} X_l \\ Y_l \end{bmatrix} = \tilde{M} X_l + \tilde{N} Y_l = I \tag{2.25}$$

Similarly, two matrices M and N in $\mathcal{RH}_\infty$ are right coprime over $\mathcal{RH}_\infty$ if they have the same number of columns and if there exist matrices X_r and Y_r in $\mathcal{RH}_\infty$ such that

$$\begin{bmatrix} Y_r & X_r \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} = Y_r N + X_r M = I \tag{2.26}$$

Definition 2.3. (Left coprime factorisation) *Let P be a proper real rational transfer matrix. A left coprime factorisation of P is a factorisation $P = \tilde{M}^{-1} \tilde{N}$ where $\tilde{M}$ and $\tilde{N}$ are left coprime over $\mathcal{RH}_\infty$.*

If P has the following state-space realisation:

$$P = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \tag{2.27a}$$

i.e.,

$$P(s) = C(sI - A)^{-1}B + D \quad \text{(continuous-time case)} \tag{2.27b}$$

or

$$P(z) = C(zI - A)^{-1}B + D \quad \text{(discrete-time case)} \tag{2.27c}$$

with $(A,\, C)$ detectable[3], a left coprime factorisation $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}$ can be constructed according to the following proposition.

Proposition 2.2. *Let $(A,\ B,\ C,\ D)$ be a detectable realisation of a transfer matrix P, let L be any constant injection matrix stabilising the output of P and let Z be any nonsingular matrix. Define*

$$\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix} = \left[\begin{array}{c|cc} A + LC & B + LD & L \\ \hline ZC & ZD & Z \end{array}\right] \tag{2.28}$$

i.e., (continuous-time case)

$$\begin{bmatrix}\tilde{N}(s) & \tilde{M}(s)\end{bmatrix} = ZC\big(sI - (A + LC)\big)^{-1}\begin{bmatrix}B + LD & L\end{bmatrix} + \begin{bmatrix}ZD & Z\end{bmatrix} \tag{2.29a}$$

or (discrete-time case)

$$\begin{bmatrix}\tilde{N}(z) & \tilde{M}(z)\end{bmatrix} = ZC\big(zI - (A + LC)\big)^{-1}\begin{bmatrix}B + LD & L\end{bmatrix} + \begin{bmatrix}ZD & Z\end{bmatrix} \tag{2.29b}$$

Then, $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix} \in \mathcal{RH}_\infty$ and

$$P = \tilde{M}^{-1}\tilde{N} \tag{2.30}$$

A normalised left coprime factorisation, *i.e.*, a left coprime factorisation such that $\tilde{N}\tilde{N}^\star + \tilde{M}\tilde{M}^\star = I$, can be obtained as follows.

Proposition 2.3. *Let $(A,\ B,\ C,\ D)$ be a detectable realisation of a continuous-time transfer matrix $P(s)$. Define*

$$\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix} = \left[\begin{array}{c|cc} A + LC & B + LD & L \\ \hline JC & JD & J \end{array}\right] \tag{2.31}$$

where

$$L = -\big(XC^T + BD^T\big)\tilde{R}^{-1} \tag{2.32}$$

$$J = \tilde{R}^{-1/2} \tag{2.33}$$

[3]The pair $(A,\, C)$ is called *detectable* if there exists a real matrix L of appropriate dimensions such that $A + LC$ is Hurwitz, *i.e.*, if all eigenvalues of $A + LC$ have a strictly negative real part: $\mathrm{Re}(\lambda_i(A + LC)) < 0$.

X is the stabilising solution of the following Riccati equation:

$$\big(A - BD^T\tilde{R}^{-1}C\big)X + X\big(A - BD^T\tilde{R}^{-1}C\big)^T - X\big(C^T\tilde{R}^{-1}C\big)X + BR^{-1}B^T = 0 \quad (2.34)$$

and

$$R = I + D^TD \quad (2.35)$$

$$\tilde{R} = I + DD^T \quad (2.36)$$

Then, $\begin{bmatrix}\tilde{N}(s) & \tilde{M}(s)\end{bmatrix} \in \mathcal{RH}_\infty$,

$$\forall\omega \quad \tilde{N}(j\omega)\tilde{N}^\star(j\omega) + \tilde{M}(j\omega)\tilde{M}^\star(j\omega) = I \quad (2.37)$$

and

$$P(s) = \tilde{M}^{-1}(s)\tilde{N}(s) \quad (2.38)$$

▶ **Remark.** The same result holds in the discrete-time case with X the stabilising solution of the following discrete algebraic Riccati equation:

$$X = AXA^T + BB^T - \big(AXC^T + BD^T\big)\big(\tilde{R} + CXC^T\big)^{-1}\big(CXA^T + DB^T\big) \quad (2.39)$$

and L and J respectively given by

$$L = -\big(AXC^T + BD^T\big)\big(\tilde{R} + CXC^T\big)^{-1} \quad (2.40)$$

and

$$J = \big(\tilde{R} + CXC^T\big)^{-1/2} \quad (2.41)$$

◀

Definition 2.4. (RIGHT COPRIME FACTORISATION) *Let P be a proper real rational transfer matrix. A right coprime factorisation of P is a factorisation $P = NM^{-1}$ where N and M are right coprime over $\mathcal{RH}_\infty$.*

If P is stabilisable[4], such a right coprime factorisation can be constructed as follows.

Proposition 2.4. *Let $(A,\ B,\ C,\ D)$ be a stabilisable realisation of a transfer matrix P, let F be any constant feedback matrix stabilising P and let Z be any nonsingular matrix. Define*

$$\begin{bmatrix}N\\M\end{bmatrix} = \left[\begin{array}{c|c} A+BF & BZ \\ \hline C+DF & DZ \\ F & Z \end{array}\right] \quad (2.42)$$

[4]A transfer matrix P with realisation $(A,\ B,\ C,\ D)$ is stabilisable if the pair $(A,\ B)$ is stabilisable, *i.e.*, if there exists a real matrix F of appropriate dimensions such that $A + BF$ is Hurwitz, meaning that all eigenvalues of $A + BF$ have a strictly negative real part: $\mathrm{Re}(\lambda_i(A + BF)) < 0$.

i.e., (continuous-time case)

$$\begin{bmatrix} N(s) \\ M(s) \end{bmatrix} = \begin{bmatrix} C + DF \\ F \end{bmatrix} \big(sI - (A + BF)\big)^{-1} BZ + \begin{bmatrix} DZ \\ Z \end{bmatrix} \tag{2.43a}$$

or (discrete-time case)

$$\begin{bmatrix} N(z) \\ M(z) \end{bmatrix} = \begin{bmatrix} C + DF \\ F \end{bmatrix} \big(zI - (A + BF)\big)^{-1} BZ + \begin{bmatrix} DZ \\ Z \end{bmatrix} \tag{2.43b}$$

Then, $\left[\begin{smallmatrix} N \\ M \end{smallmatrix}\right] \in \mathcal{RH}_\infty$ *and*

$$P = NM^{-1} \tag{2.44}$$

The following proposition gives the procedure to build a normalised right coprime factorisation, *i.e.*, a right coprime factorisation such that $N^\star N + M^\star M = I$.

Proposition 2.5. *Let* $(A,\ B,\ C,\ D)$ *be a stabilisable realisation of a continuous-time transfer matrix* $P(s)$*. Define*

$$\begin{bmatrix} N \\ M \end{bmatrix} = \left[\begin{array}{c|c} A + BF & BH \\ \hline C + DF & DH \\ F & H \end{array}\right] \tag{2.45}$$

where

$$F = -R^{-1}\big(B^T X + D^T C\big) \tag{2.46}$$

$$H = R^{-1/2} \tag{2.47}$$

X *is the stabilising solution of the following Riccati equation:*

$$\begin{aligned} \big(A - BR^{-1}D^T C\big)^T X + X\big(A - BR^{-1}D^T C\big) \\ + X\big(-BR^{-1}B^T\big)X + C^T \tilde{R}^{-1} C = 0 \end{aligned} \tag{2.48}$$

and

$$R = I + D^T D \tag{2.49}$$

$$\tilde{R} = I + DD^T \tag{2.50}$$

Then, $\left[\begin{smallmatrix} N(s) \\ M(s) \end{smallmatrix}\right] \in \mathcal{RH}_\infty$*,*

$$\forall \omega \quad N^\star(j\omega)N(j\omega) + M^\star(j\omega)M(j\omega) = I \tag{2.51}$$

and

$$P(s) = N(s)M^{-1}(s) \tag{2.52}$$

▶ **Remark.** The same result holds in the discrete-time case with X the stabilising solution of the following discrete algebraic Riccati equation:

$$X = A^T X A + C^T C - \big(A^T X B + C^T D\big)\big(R + B^T X B\big)^{-1}\big(B^T X A + D^T C\big) \tag{2.53}$$

and F and H respectively given by

$$F = -\left(R + B^T X B\right)^{-1}\left(B^T X A + D^T C\right) \tag{2.54}$$

and

$$H = \left(R + B^T X B\right)^{-1/2} \tag{2.55}$$

◀

In the sequel, we shall generally use the notations $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}$ (respectively $\begin{bmatrix}N\\M\end{bmatrix}$) for the left (respectively right) coprime factorisations of a system $G = \tilde{M}^{-1}\tilde{N} = NM^{-1}$ and $\begin{bmatrix}\tilde{U} & \tilde{V}\end{bmatrix}$ (respectively $\begin{bmatrix}U\\V\end{bmatrix}$) for the left (respectively right) coprime factorisations of a controller $K = \tilde{V}^{-1}\tilde{U} = UV^{-1}$.

2.4.2 The Bezout Identity and Closed-loop Stability

Consider a closed-loop system as in Figure 2.1. Its closed-loop transfer matrix $T_o(G_0, K)$, given by (2.3), can be expressed in function of any pair of left coprime factors $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}$ of the plant $G_0 = \tilde{M}^{-1}\tilde{N}$ and of any pair of right coprime factors $\begin{bmatrix}U\\V\end{bmatrix}$ of the controller $K = UV^{-1}$ as

$$\begin{aligned} T_o(G_0, K) &= \begin{bmatrix}K\\I\end{bmatrix}(I + G_0K)^{-1}\begin{bmatrix}G_0 & I\end{bmatrix} \\ &= \begin{bmatrix}UV^{-1}\\I\end{bmatrix}(I + \tilde{M}^{-1}\tilde{N}UV^{-1})^{-1}\begin{bmatrix}\tilde{M}^{-1}\tilde{N} & I\end{bmatrix} \\ &= \begin{bmatrix}U\\V\end{bmatrix}\Phi^{-1}\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix} \end{aligned} \tag{2.56}$$

where

$$\Phi = \begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}\begin{bmatrix}U\\V\end{bmatrix} \tag{2.57}$$

Lemma 2.1. *The closed-loop transfer matrix $T_o(G_0, K)$ of (2.56) is stable if and only if Φ is a unit (i.e., Φ, $\Phi^{-1} \in \mathcal{RH}_\infty$).*

Proof. This follows from the fact that $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}, \begin{bmatrix}U\\V\end{bmatrix} \in \mathcal{RH}_\infty$, by definition of the coprime factors, hence the only remaining necessary and sufficient condition of closed-loop stability is that Φ be inversely stable. □

Lemma 2.2. (Bezout identity) *The closed-loop transfer matrix $T_o(G_0, K)$ of (2.56) is stable if and only if there exist plant and controller coprime factors $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}$ and $\begin{bmatrix}U\\V\end{bmatrix}$ such that*

$$\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}\begin{bmatrix}U\\V\end{bmatrix} = I \tag{2.58}$$

This equation is called a Bezout identity.

Proof. The sufficient condition is a direct consequence of Lemma 2.1, since I is a unit. The proof of the necessary condition is constructive: let $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}$ and $\begin{bmatrix}U\\V\end{bmatrix}$ be any pairs of plant and controller coprime factors. By closed-loop stability hypothesis, $\Phi = \begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}\begin{bmatrix}U\\V\end{bmatrix}$ is a unit, hence $\Phi^{-1} \in \mathcal{RH}_\infty$. Let us define $\begin{bmatrix}U'\\V'\end{bmatrix} \triangleq \begin{bmatrix}U\\V\end{bmatrix}\Phi^{-1}$. Then, $\begin{bmatrix}U'\\V'\end{bmatrix} \in \mathcal{RH}_\infty$ and $U'V'^{-1} = U\Phi^{-1}(V\Phi^{-1})^{-1} = UV^{-1} = K$, which means that $\begin{bmatrix}U'\\V'\end{bmatrix}$ is a coprime factorisation of K, and it follows from its definition that $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}\begin{bmatrix}U'\\V'\end{bmatrix} = I$. Hence, $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}$ and $\begin{bmatrix}U'\\V'\end{bmatrix}$ are plant and controller coprime factors satisfying the Bezout identity. □

In a similar way, one could define $\begin{bmatrix}\tilde{N}' & \tilde{M}'\end{bmatrix} \triangleq \Phi^{-1}\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}$ and verify that it is a pair of plant coprime factors satisfying the Bezout identity with $\begin{bmatrix}U\\V\end{bmatrix}$. This means that, starting from any two pairs of plant and controller coprime factors $\begin{bmatrix}\tilde{N} & \tilde{M}\end{bmatrix}$ and $\begin{bmatrix}U\\V\end{bmatrix}$ and if the closed-loop system is stable, it is always possible to satisfy the Bezout identity by altering only one pair of coprime factors (either those of the plants or those of the controller), which leaves all freedom for the other pair which could be, for instance, normalised. The following proposition is a direct consequence of this observation.

Proposition 2.6. *Consider a stable closed-loop system with transfer matrix $T_o(G_0, K)$ given by (2.56). Let $G_0 = \tilde{M}^{-1}\tilde{N}$ and $K = UV^{-1}$ define, respectively, a left coprime factorisation of the plant and a right coprime factorisation of the controller. Then, two of the following three equalities can always be satisfied simultaneously:*

- *normalisation of the left coprime factors of G_0:*

$$\tilde{N}\tilde{N}^\star + \tilde{M}\tilde{M}^\star = I \tag{2.59}$$

- *normalisation of the right coprime factors of K:*

$$U^\star U + V^\star V = I \tag{2.60}$$

- *Bezout identity:*

$$\Phi = \tilde{N}U + \tilde{M}V = I \tag{2.61}$$

All these derivations could also be made for the closed-loop transfer matrix $T_i(G_0, K)$ of (2.2), which can be recast as

$$\begin{aligned} T_i(G_0, K) &= \begin{bmatrix}G_0\\I\end{bmatrix}(I + KG_0)^{-1}\begin{bmatrix}K & I\end{bmatrix} \\ &= \begin{bmatrix}NM^{-1}\\I\end{bmatrix}(I + \tilde{V}^{-1}\tilde{U}NM^{-1})^{-1}\begin{bmatrix}\tilde{V}^{-1}\tilde{U} & I\end{bmatrix} \\ &= \begin{bmatrix}N\\M\end{bmatrix}\tilde{\Phi}^{-1}\begin{bmatrix}\tilde{U} & \tilde{V}\end{bmatrix} \end{aligned} \tag{2.62}$$

where

$$\tilde{\Phi} = \begin{bmatrix} \tilde{U} & \tilde{V} \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} \tag{2.63}$$

using plant right coprime factors $G_0 = NM^{-1}$ and controller left coprime factors $K = \tilde{V}^{-1}\tilde{U}$. The existence of plant and controller coprime factors yielding the Bezout identity $\tilde{U}N + \tilde{V}M = I$ is then a necessary and sufficient condition for closed-loop stability. The following proposition summarises these results.

Proposition 2.7. (Double Bezout identity) *Consider the closed-loop system of Figure 2.1. This system is internally stable if and only if there exist plant and controller left and right coprime factorisations* $G_0 = \tilde{M}^{-1}\tilde{N} = NM^{-1}$ *and* $K = \tilde{V}^{-1}\tilde{U} = UV^{-1}$ *satisfying the double Bezout identity*

$$\begin{bmatrix} \tilde{M} & -\tilde{N} \\ \tilde{U} & \tilde{V} \end{bmatrix} \begin{bmatrix} V & N \\ -U & M \end{bmatrix} = I \tag{2.64}$$

2.5 The ν-gap Metric

2.5.1 Definition

The ν-gap metric between two continuous-time transfer matrices $G_1(s)$ and $G_2(s)$ is a measure of distance between these two systems. It was first introduced by G. Vinnicombe in (Vinnicombe, 1993*a*, 1993*b*).

Definition 2.5. (ν-gap metric) *The ν-gap metric between two transfer matrices G_1 and G_2 is defined as*

$$\delta_\nu(G_1, G_2) = \begin{cases} \left\| \kappa\big(G_1(j\omega), G_2(j\omega)\big) \right\|_\infty & \text{if } \det \Xi(j\omega) \neq 0 \quad \forall \omega \\ & \text{and } \mathrm{wno}\big(\det \Xi(s)\big) = 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.65}$$

where

- $\Xi(s) \triangleq N_2^\star(s)N_1(s) + M_2^\star(s)M_1(s)$;
- $\kappa(G_1(j\omega), G_2(j\omega)) \triangleq -\tilde{N}_2(j\omega)M_1(j\omega) + \tilde{M}_2(j\omega)N_1(j\omega)$ *is called the* chordal distance *between G_1 and G_2 at frequency ω;*
- $G_1(s) = N_1(s)M_1^{-1}(s)$ *and* $G_2(s) = N_2(s)M_2^{-1}(s) = \tilde{M}_2^{-1}(s)\tilde{N}_2(s)$ *are normalised coprime factorisations of G_1 and G_2;*
- $\mathrm{wno}(P(s)) = \eta(P^{-1}(s)) - \eta(P(s))$ *is called the* winding number *of the transfer function $P(s)$ and is defined as the number of counterclockwise encirclements (a clockwise encirclement counts as a negative encirclement) around the origin of the Nyquist contour of $P(s)$ indented around the right of any imaginary axis pole of $P(s)$;*
- $\eta(P(s))$ *denotes the number of poles of $P(s)$ in $\mathbb{C}_0^+$.*

The definition also holds in the discrete-time case by means of the use of the bilinear transformation $s = \frac{z-1}{z+1}$.

It has been shown in (Vinnicombe, 1993*a*) that

$$\delta_\nu(G_1,\, G_2) = \delta_\nu(G_2,\, G_1) = \delta_\nu(G_1^T,\, G_2^T) \tag{2.66}$$

An alternative definition of the ν-gap is the following:

$$\delta_\nu(G_1,\, G_2) = \begin{cases} \big\|\kappa\big(G_1(j\omega),\, G_2(j\omega)\big)\big\|_\infty \\ \qquad \text{if } \det\big(I + G_2^\star(j\omega)G_1(j\omega)\big) \neq 0\ \forall\omega \text{ and} \\ \qquad \text{wno}\Big(\det\big(I + G_2^\star(s)G_1(s)\big)\Big) + \eta\big(G_1(s)\big) \\ \qquad\qquad\qquad -\eta\big(G_2(s)\big) - \eta_0\big(G_2(s)\big) = 0 \\ 1 \qquad \text{otherwise} \end{cases} \tag{2.67}$$

where $\eta_0(P(s))$ is the number of imaginary axis poles of $P(s)$ and where $\kappa(G_1(j\omega),\, G_2(j\omega))$ can be written as

$$\begin{aligned} \kappa\big(G_1(j\omega),\, G_2(j\omega)\big) &= \big(I + G_2(j\omega)G_2^\star(j\omega)\big)^{-1/2} \\ &\quad \times \big(G_1(j\omega) - G_2(j\omega)\big) \times \big(I + G_1^\star(j\omega)G_1(j\omega)\big)^{-1/2} \end{aligned} \tag{2.68}$$

In the SISO case, the ν-gap metric has a nice geometric interpretation. Indeed, the chordal distance at frequency ω, $\kappa(G_1(j\omega),\, G_2(j\omega))$, is the distance between the projections onto the Riemann sphere of the points of the Nyquist plots of G_1 and G_2 corresponding to that frequency (hence the appellation 'chordal distance'). The Riemann sphere is a unit-diameter sphere tangent at its south pole to the complex plane at its origin and the points of the Nyquist plots are projected onto the sphere using its north pole as centre of projection. Due to this particular projection, the chordal distance has a maximum resolution at frequencies where $|G_1| \approx 1$ and/or $|G_2| \approx 1$, *i.e.* around the cross-over frequencies of G_1 and G_2, since the corresponding points are projected onto the equator of the Riemann sphere. This property makes the ν-gap a control-oriented measure of distance between two systems.

► **Example 2.4.** Consider the systems $G_1(s) = \frac{1}{s+1}$ and $G_2(s) = (\frac{1}{s+1})^3$. Their Nyquist diagrams and their projections onto the Riemann sphere are depicted in Figure 2.3. Consider, for instance, the points $G_1(j\omega_1)$ and $G_1(j\omega_1)$ with $\omega_1 = 0.938$ rad/s. They are respectively located at the coordinates $(0.532, -0.499, 0)$ and $(-0.247, -0.299, 0)$. Their projections on the Riemann sphere are respectively located at the coordinates $(0.347, -0.326, 0.347)$ and $(-0.214, -0.260, 0.131)$. The distance between these two points, represented by a line segment inside the sphere in Figure 2.3, is 0.606, which is precisely $|\kappa(G_1(j\omega_1),\, G_2(j\omega_1))|$. ◄

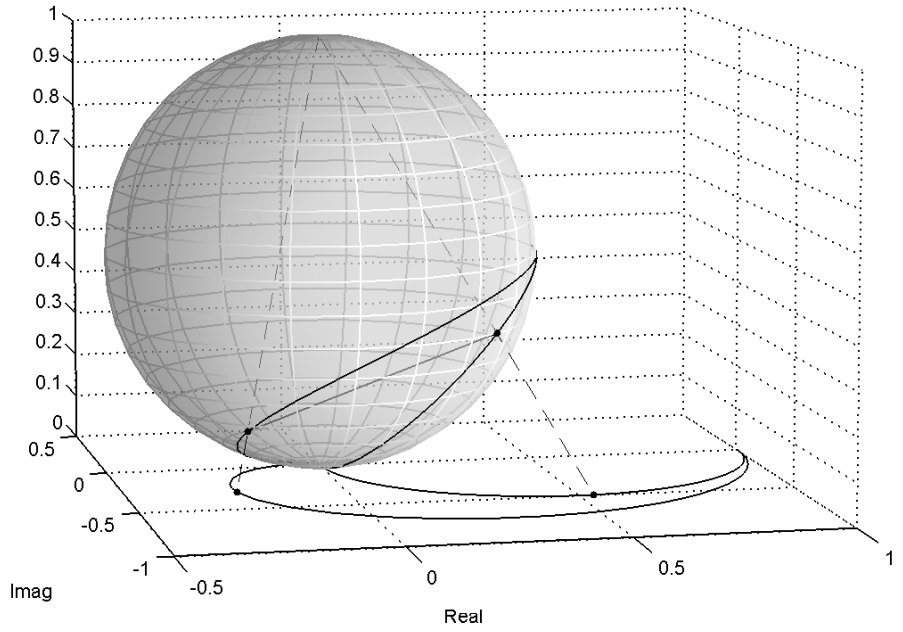

Figure 2.3. Projection onto the Riemann sphere of the Nyquist plots of G_1 and G_2, and chordal distance between $G_1(j\omega_1)$ and $G_2(j\omega_1)$

2.5.2 Stabilisation of a Set of Systems by a Given Controller and Comparison with the Directed Gap Metric

The definition of the ν-gap metric and in particular the winding number condition it involves, is based on a robust stability argument. In the theory of robust control, coprime-factor uncertainties are often considered. Let $G_1(s) = N_1(s)M_1^{-1}(s)$ define the normalised right coprime factorisation of a nominal system G_1 and let $\Delta = \begin{bmatrix} \Delta_N \\ \Delta_M \end{bmatrix}$ be a coprime-factor perturbation. If K is a feedback controller that stabilises G_1, then K also stabilises all G_2 in the set

$$\mathcal{G}_\beta^d = \left\{ G_2 = (N_1 + \Delta_N)(M_1 + \Delta_M)^{-1} \mid \Delta \in \mathcal{H}_\infty,\, \|\Delta\|_\infty \leq \beta \right\} \tag{2.69}$$

if and only if (Georgiou and Smith, 1990)

$$b_{G_1,K} > \beta \tag{2.70}$$

An alternative definition of this set is the following:

$$\mathcal{G}_\beta^d = \left\{ G_2 \mid \vec{\delta}_g(G_1,\, G_2) < \beta \right\} \tag{2.71}$$

where $\vec{\delta}_g(G_1, G_2)$ is the directed gap defined as

$$\vec{\delta}_g(G_1, G_2) = \inf_{Q(s)\in\mathcal{H}_\infty} \left\| \begin{bmatrix} N_1(s) \\ M_1(s) \end{bmatrix} - \begin{bmatrix} N_2(s) \\ M_2(s) \end{bmatrix} Q(s) \right\|_\infty \tag{2.72}$$

and where $G_2(s) = N_2(s)M_2^{-1}(s)$ defines the normalised right coprime factorisation of G_2. However, it can be shown (Vinnicombe, 1993*b*) that the largest class of systems that can be guaranteed to be stabilised *a priori* by K consists of those G_2 satisfying

$$\inf_{Q(s)\in\mathcal{L}_\infty} \left\| \begin{bmatrix} N_1(s) \\ M_1(s) \end{bmatrix} - \begin{bmatrix} N_2(s) \\ M_2(s) \end{bmatrix} Q(s) \right\|_\infty < \beta, \quad \text{wno}\big(\det \Xi(s)\big) = 0 \tag{2.73}$$

(where $\Xi(s)$ is defined in Definition 2.5), which is precisely the set

$$\begin{aligned} \mathcal{G}_\beta^\nu = \Big\{ & G_2 = (N_1 + \Delta_N)(M_1 + \Delta_M)^{-1} \mid \Delta \in \mathcal{L}_\infty, \, \|\Delta\|_\infty \le \beta \\ & \text{and } \eta(G_2) = \text{wno}(M_1 + \Delta_M) \Big\} = \Big\{ G_2 \mid \delta_\nu(G_1, G_2) \le \beta \Big\} \end{aligned} \tag{2.74}$$

Hence, one can define a larger set of plants that are guaranteed to be stabilised by a given controller K with the ν-gap metric than with directed gap, since the ν-gap allows coprime-factor perturbations in $\mathcal{L}_\infty$ rather than in $\mathcal{H}_\infty$. Another serious advantage of the ν-gap over the directed gap is the fact that the former is much easier to compute than the latter. However, in order to be valid, the robust stability theory with the ν-gap metric requires the verification of the winding number condition (which can take various forms depending on the chosen definition of the ν-gap). The demonstration of the necessity of the winding number condition is very complicated and is outside the scope of this book. The interested reader is referred to (Vinnicombe, 1993*a*, 1993*b*, 2000).

2.5.3 The ν-gap Metric and Robust Stability

The main interest of the ν-gap metric is its use in a range of robust stability results. One of these results relates the size of the set of robustly stabilising controllers of a ν-gap uncertainty set (*i.e.*, a set of the form (2.74) defined with the ν-gap) to the size of this uncertainty set, as summarised in the following two propositions.

Proposition 2.8. (Vinnicombe, 2000) *Let us consider the uncertainty set $\mathcal{G}_\gamma^\kappa$, centred at a model G_1, defined by*

$$\mathcal{G}_\gamma^\kappa = \Big\{ G_2 \mid \kappa\big(G_1(e^{j\omega}), G_2(e^{j\omega})\big) \le \gamma(\omega) \ \forall\omega \text{ and } \delta_\nu(G_1, G_2) < 1 \Big\} \tag{2.75}$$

with $0 \le \gamma(\omega) < 1 \ \forall\omega$. Then, a controller K stabilising G_1 stabilises all plants in the uncertainty set $\mathcal{G}_\gamma^\kappa$ if and only if it lies in the controller set

$$\mathcal{C}_\gamma^\kappa = \Big\{ K(z) \mid \kappa\left(G_1(e^{j\omega}), \tfrac{-1}{K(e^{j\omega})}\right) > \gamma(\omega) \quad \forall\omega \Big\} \tag{2.76}$$

The second proposition is a Min-Max version of the first one:

Proposition 2.9. (Vinnicombe, 2000) *Let us consider the ν-gap uncertainty set $\mathcal{G}_\beta^\nu$ of size $\beta < 1$ centred at a model G_1:*

$$\mathcal{G}_\beta^\nu = \Big\{ G_2 \mid \delta_\nu(G_1, G_2) \leq \beta \Big\} \tag{2.77}$$

Then, a controller K stabilising G_1 stabilises all plants in the uncertainty set $\mathcal{G}_\beta^\nu$ if and only if it lies in the controller set

$$\mathcal{C}_\beta^\nu = \Big\{ K(z) \mid b_{G_1,K} > \beta \Big\} \tag{2.78}$$

The size β of a ν-gap uncertainty set $\mathcal{G}_\beta^\nu$ is thus directly connected to the size of the set of all controllers that robustly stabilise $\mathcal{G}_\beta^\nu$. Moreover, the smaller this size β, the larger the set of controllers that robustly stabilises $\mathcal{G}_\beta^\nu$. This result will be of the highest importance in Chapter 5.

2.6 Prediction-error Identification

Prediction-error identification is the only modelling tool, considered in this book, that uses data collected on the process to obtain a mathematical representation of it under the form of a transfer function or matrix. Contrary to, *e.g.*, first-principles modelling, the objective here is not to build a good *knowledge* model of the process, but to obtain a (generally black-box) *representation* model that exhibits a good qualitative and quantitative matching of the process behaviour. Such a model is generally delivered with an uncertainty region around it, which can be used for robust control design.

Our intention here is not to give a thorough theoretical description of the method. The interested reader is kindly referred to the existing literature on the subject and more particularly to (Ljung, 1999) for the detailed theory of prediction-error identification and (Zhu, 2001) for its application to multivariable systems in a process control framework.

We make the assumption that the true system is the possibly multi-input multi-output, linear time-invariant[5] system described by

$$\mathcal{S} : \begin{cases} y(t) = G_0(z)u(t) + v(t) \\ v(t) = H_0(z)e(t) \end{cases} \tag{2.79}$$

where $G_0(z)$ and $H_0(z)$ are rational transfer functions or matrices. $G_0(z)$ is strictly proper and has p outputs and m inputs. $H_0(z)$ is a stable and inversely

[5]This may seem a restrictive hypothesis as, in practice, all industrial systems exhibit at least a little amount of nonlinearities and have a tendency to alter with the time. Conceptually, however, the idea of an LTI system is generally perfectly acceptable if the plant is regarded around a given operating point.

stable, minimum-phase, proper and monic[6] $p \times p$ transfer matrix. $u(t) \in \mathbb{R}^m$ is the control input signal, $y(t) \in \mathbb{R}^p$ is the observed output signal and $e(t) \in \mathbb{R}^p$ is a white noise process with zero mean and covariance matrix Λ_0 (or variance λ_0 in the single-input single-output case).

For the sake of simplicity, however, we shall restrict the derivations of this section to the SISO case except where explicitely indicated.

2.6.1 Signals Properties

The assumption is made that all signals are quasi-stationary (Ljung, 1999). A quasi-stationary signal $y(t)$ is a signal for which the following auto-correlation function exists:

$$R_y(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} Ey(t)y(t-\tau) \triangleq \bar{E}y(t)y(t-\tau) \tag{2.80}$$

where the expectation is taken with respect to the stochastic components of the signal. *e.g.*, if $y(t)$ is the output of the system (2.79) with $u(t)$ deterministic and $v(t)$ zero-mean stochastic, then $Ey(t) = G_0(z)u(t)$. This quasi-stationarity assumption is useful to treat deterministic, stochastic or mixed signals in a common framework with theoretical exactness and it allows defining spectra and cross-spectra of signals as follows.

The spectrum (or power spectral density) of a quasi-stationary signal $y(t)$ is defined by

$$\phi_y(\omega) = \sum_{\tau=-\infty}^{\infty} R_y(\tau)e^{-j\omega\tau} \tag{2.81}$$

The cross-correlation function and the cross-spectrum of two quasi-stationary signals are respectively defined by

$$R_{yu}(\tau) = \bar{E}y(t)u(t-\tau) \tag{2.82}$$

and

$$\phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau)e^{-j\omega\tau} \tag{2.83}$$

When (2.79) is satisfied, the following relations hold:

$$\phi_v(\omega) = \left|H_0(e^{j\omega})\right|^2 \phi_e(\omega) = \left|H_0(e^{j\omega})\right|^2 \lambda_0 \tag{2.84}$$

$$\phi_y(\omega) = \phi_y^u(\omega) + \phi_y^e(\omega) \triangleq \left|G_0(e^{j\omega})\right|^2 \phi_u(\omega) + \left|H_0(e^{j\omega})\right|^2 \lambda_0 \tag{2.85}$$

$$\phi_{yu}(\omega) = G_0(e^{j\omega})\phi_u(\omega) \tag{2.86}$$

[6] A monic filter is a filter whose impulse-response zeroth coefficient is 1 or the unit matrix.

etc. Spectra are real-valued functions of the frequency ω, while cross-spectra are complex-valued functions of ω. $\phi_y^u(\omega)$ denotes the spectrum of that part of $y(t)$ that originates from $u(t)$. It is not the same as the cross-spectrum $\phi_{yu}(\omega)$.

In the MIMO case, these expressions become a little more complicated:

$$\phi_y^u(\omega) = G_0(e^{j\omega})\phi_u(\omega)\left[G_0(e^{j\omega})\right]^\star \tag{2.87}$$

etc.

The following equality, called Parseval's relationship, holds:

$$E\left|y(t)\right|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi}\phi_y(\omega)\,d\omega \tag{2.88}$$

It says that the power of the signal $y(t)$ is equal to the power contained in its spectrum. This relationship will have important consequences regarding the distribution of the modelling error.

In practice, the following formulae can be used to estimate auto-correlation or cross-correlation functions:

$$\hat{R}_y^N(\tau) = \frac{1}{N}\sum_{t=1}^{N} y(t)y(t-\tau) \tag{2.89}$$

$$\hat{R}_{yu}^N(\tau) = \frac{1}{N}\sum_{t=1}^{N} y(t)u(t-\tau) \tag{2.90}$$

and these estimates can be used to calculate spectra and cross-spectra as

$$\hat{\phi}_y(\omega) = \sum_{\tau=-\tau_m}^{\tau_m}\hat{R}_y(\tau)e^{-j\omega\tau} \tag{2.91}$$

$$\hat{\phi}_{yu}(\omega) = \sum_{\tau=-\tau_m}^{\tau_m}\hat{R}_{yu}(\tau)e^{-j\omega\tau} \tag{2.92}$$

with a suitable τ_m like, *e.g.*, $\tau_m = N/10$. It can be shown that, for $N \to \infty$, these estimates will converge with probability one to the true $R_y(\tau)$, $R_{yu}(\tau)$, $\phi_y(\omega)$ and $\phi_{yu}(\omega)$, provided the signals are ergodic[7].

2.6.2 The Identification Method

The objective of system identification is to compute a parametrised model $\mathcal{M}(\theta)$ for the system:

$$\mathcal{M}(\theta) : \hat{y}(t) = G(z,\,\theta)u(t) + H(z,\,\theta)e(t) \tag{2.93}$$

[7] An ergodic stochastic process $x(t)$ is a process whose time average $\lim_{N\to\infty}\frac{1}{2N}\sum_{t=-N}^{N}x(t)$ tends to its ensemble average, *i.e.*, to its expectation $Ex(t)$.

This model lies in some model set $\mathcal{M}$ selected by the designer:

$$\mathcal{M} \triangleq \Big\{ \mathcal{M}(\theta) \mid \theta \in D_\theta \subseteq \mathbb{R}^n \Big\} \tag{2.94}$$

i.e., $\mathcal{M}$ is the set of all models with the same structure as $\mathcal{M}(\theta)$. The parameter vector θ ranges over a set $D_\theta \subseteq \mathbb{R}^n$ that is assumed to be compact and connected. We say that the true system is in the model set, which is denoted by $\mathcal{S} \in \mathcal{M}$, if

$$\exists \theta_0 \in D_\theta : G(z, \theta_0) = G_0,\ H(z, \theta_0) = H_0 \tag{2.95}$$

Otherwise, we say that there is *undermodelling* of the system dynamics. The case where the noise properties cannot be correctly described within the model set but where

$$\exists \theta_0 \in D_\theta : G(z, \theta_0) = G_0 \tag{2.96}$$

will be denoted by $G_0 \in \mathcal{G}$.

The prediction-error identification procedure uses a finite set of N input-output data

$$Z^N = \Big\{ u(1),\, y(1),\, \ldots,\, u(N),\, y(N) \Big\} \tag{2.97}$$

to compute the one-step-ahead prediction of the output signal at each time sample $t \in [1,\, N]$, using the available past data samples[8] and the model with its parameter vector θ:

$$\hat{y}(t \mid t-1,\, \theta) = H^{-1}(z,\, \theta) G(z,\, \theta) u(t) + \big(1 - H^{-1}(z,\, \theta)\big) y(t) \tag{2.98}$$

(Observe that, because $G(z,\, \theta)$ is strictly proper and $H(z,\, \theta)$ is monic, the two terms of the right-hand side depend only on past u's and y's.) The prediction error at time t is

$$\varepsilon(t,\, \theta) = y(t) - \hat{y}(t \mid t-1,\, \theta) = H^{-1}(z,\, \theta)\big(y(t) - G(z,\, \theta) u(t)\big) \tag{2.99}$$

Observe that it would be equal to the white noise $e(t)$, *i.e.*, to the only absolutely unpredictable part of $y(t)$, if $G(z,\, \theta)$ and $H(z,\, \theta)$ were equal to $G_0(z)$ and $H_0(z)$, respectively. The objective of prediction-error identification is to find a parameter vector $\hat{\theta}$ such that $\varepsilon(t,\, \hat{\theta})$ be whitened, meaning that all the useful information contained in the data Z^N (*i.e.*, everything but the stochastic white noise $e(t)$) has been exploited.

Given the chosen model structure (2.93) and measured data (2.97), the prediction-error estimate of θ is determined through

$$\hat{\theta} = \arg \min_{\theta \in D_\theta} V_N\big(\theta,\, Z^N\big) \tag{2.100}$$

---

[8] The difference between *simulation* and *prediction* is that the former only uses the measured input signal and filters it through the transfer function $G(z,\, \theta)$ of the model to compute an estimate of the output signal, while the latter uses all available information, including past output samples, to build an estimation, or prediction, of the future outputs.

where $V_N(\theta, Z^N)$ is a quadratic criterion:

$$V_N\big(\theta, Z^N\big) = \begin{cases} \frac{1}{N}\sum_{t=1}^{N} \varepsilon_F^T(t, \theta)\Lambda^{-1}\varepsilon_F(t, \theta) & \text{(MIMO case)} \\ \\ \frac{1}{N}\sum_{t=1}^{N} \varepsilon_F^2(t, \theta) & \text{(SISO case)} \end{cases} \tag{2.101}$$

In this expression, Λ is a symmetric positive-definite weighting matrix and $\varepsilon_F(t, \theta)$ are the possibly filtered prediction errors:

$$\varepsilon_F(t, \theta) = L(z, \theta)\varepsilon(t, \theta) \tag{2.102}$$

where $L(z, \theta)$ is any linear, stable, monic and possibly parametrised prefilter. Since

$$\varepsilon_F(t, \theta) = L(z, \theta)H^{-1}(z, \theta)\big(y(t) - G(z, \theta)u(t)\big) \tag{2.103}$$

this filter can be included in the noise model structure and, without loss of generality, we shall make the assumption that $L(z, \theta) = I$ in the sequel. (Observe, apropos, that if $L(z, \theta) = H(z, \theta)$, then $\varepsilon_F(t, \theta) = y(t) - G(z, \theta)u(t)$, which is the simulation error at time t. The noise model disappears then from the identification criterion, meaning that no noise model is identified. This is the output-error case: see below.)

The following notation will often be used for the estimates $G(z, \hat{\theta})$, $H(z, \hat{\theta})$, *etc.*:

$$\hat{G}(z) = G(z, \hat{\theta}) \quad \text{and} \quad \hat{H}(z) = H(z, \hat{\theta}) \tag{2.104}$$

2.6.3 Usual Model Structures

Some commonly used polynomial model structures are the following.

- FIR (Finite Impulse Response model structure):

$$y(t) = B(z)z^{-k}u(t) + e(t) \tag{2.105}$$

- ARX (Auto-Regressive model structure with eXogenous inputs):

$$A(z)y(t) = B(z)z^{-k}u(t) + e(t) \tag{2.106}$$

- ARMAX (Auto-Regressive Moving-Average model structure with eXogenous inputs):

$$A(z)y(t) = B(z)z^{-k}u(t) + C(z)e(t) \tag{2.107}$$

- OE (Output-Error model structure):

$$y(t) = F^{-1}(z)B(z)z^{-k}u(t) + e(t) \tag{2.108}$$

- BJ (Box-Jenkins model structure):

$$y(t) = F^{-1}(z)B(z)z^{-k}u(t) + D^{-1}(z)C(z)e(t) \tag{2.109}$$

In these expressions, k is the length of the dead time of the transfer function (expressed in number of samples), $B(z)$ is a polynomial (SISO case) or a polynomial matrix (MIMO case) of order n_b in z^{-1}, and $A(z)$, $C(z)$, $D(z)$ and $F(z)$ are monic polynomials[9] or polynomial matrices in z^{-1}, respectively of orders n_a, n_c, n_d and n_f.

► **Remark.** Prediction-error identification only works with stable predictors, *i.e.*, the product $H^{-1}(z, \theta)G(z, \theta)$ in (2.98) is constrained to be stable (otherwise the prediction errors might not be bounded). This means that the only way to identify an unstable system is to use a model structure where the unstable poles of $G(z, \theta)$ are also in $H(z, \theta)$, *e.g.*, an ARX or ARMAX structure, allowing the predictor to be stable although the model is unstable. On the contrary, the use of an OE model structure will enforce stability of the estimate. ◄

2.6.4 Computation of the Estimate

Depending on the chosen model structure, $\hat{\theta}$ can be obtained algebraically or *via* an optimisation procedure.

A. The FIR and ARX cases. With a FIR or ARX model structure, the model is linear in the parameters:

$$\mathcal{M}(\theta) : \hat{y}(t) = \varphi^T(t)\theta + e(t) \tag{2.110}$$

hence

$$\hat{y}(t \mid t-1, \theta) = \varphi^T(t)\theta \tag{2.111}$$

where

$$\theta = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_0 & \dots & b_{n_b} \end{bmatrix}^T \tag{2.112}$$

and

$$\varphi(t) = \begin{bmatrix} -y(t-1) & \dots & -y(t-n_a) & u(t-n_k) & \dots & u(t-n_b-n_k) \end{bmatrix}^T \tag{2.113}$$

are respectively a parameter vector and a regression vector. The minimising argument of $V_N(\theta, Z^N)$ is then obtained by the standard least-squares method:

$$\hat{\theta} = \left[\frac{1}{N} \sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^{N} \varphi(t)y(t) \tag{2.114}$$

[9] A polynomial is monic if its independent term is 1.

B. Other cases. Other model structures require numerical optimisation to find the estimate. A standard search routine is the following:

$$\hat{\theta}^{i+1} = \hat{\theta}^i - \mu^i \left[R_N^i\right]^{-1} V_N'\big(\hat{\theta}^i,\, Z^N\big) \tag{2.115}$$

where $\hat{\theta}^i$ is the estimate at iteration i,

$$V_N'\big(\hat{\theta}^i,\, Z^N\big) = -\frac{1}{N}\sum_{t=1}^{N} \psi\big(t,\, \hat{\theta}^i\big)\varepsilon\big(t,\, \hat{\theta}^i\big) \tag{2.116}$$

is the gradient of $V_N(\theta,\, Z^N)$ with respect to θ evaluated at $\hat{\theta}^i$, R_N^i is a matrix that determines the search direction, μ^i is a factor that determines the step size and

$$\psi(t,\, \theta) = -\frac{d}{d\theta}\varepsilon(t,\, \theta) = \frac{d}{d\theta}\hat{y}(t \mid t-1,\, \theta) \tag{2.117}$$

is the negative gradient of the prediction error. A common choice for the matrix R_N^i is

$$R_N^i = \frac{1}{N}\sum_{t=1}^{N} \psi\big(t,\, \hat{\theta}^i\big)\psi^T\big(t,\, \hat{\theta}^i\big) + \delta^i I \tag{2.118}$$

which gives the Gauss-Newton direction if $\delta^i = 0$; see, *e.g.*, (Dennis and Schnabel, 1983). δ^i is a regularisation parameter chosen so that $R_N^i > 0$.

2.6.5 Asymptotic Properties of the Estimate

The results of this subsection are given for both MIMO and SISO cases.

A. Asymptotic bias and consistency. Under mild conditions, in the MIMO case, there hold (Ljung, 1978)

$$V_N\big(\theta,\, Z^N\big) \to \bar{V}(\theta) = \bar{E}\varepsilon^T(t,\, \theta)\Lambda^{-1}\varepsilon(t,\, \theta) \quad \text{w.p. 1 as } N \to \infty \tag{2.119}$$

and

$$\hat{\theta} \to \theta^* = \arg\min_{\theta \in D_\theta} \bar{V}(\theta) \qquad \text{w.p. 1 as } N \to \infty \tag{2.120}$$

Note that we can write $\bar{V}(\theta)$ as

$$\begin{aligned} \bar{V}(\theta) &= \bar{E}\,\mathrm{trace}\big(\Lambda^{-1}\varepsilon(t,\, \theta)\varepsilon^T(t,\, \theta)\big) \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathrm{trace}\big(\Lambda^{-1}\phi_\varepsilon(\omega)\big)d\omega \end{aligned} \tag{2.121}$$

where $\phi_\varepsilon(\omega)$ is the power spectral density of the prediction error $\varepsilon(t, \theta)$. The second equality comes from Parseval's relationship. As a result, there holds[10]

$$\theta^* = \arg\min_{\theta\in D_\theta} \frac{1}{2\pi}\int_{-\pi}^{\pi} \operatorname{trace}\Bigg(\begin{bmatrix} G_0 - G(\theta) & H_0 - H(\theta)\end{bmatrix} \begin{bmatrix} \phi_u & \phi_{ue} \\ \phi_{eu} & \Lambda_0 \end{bmatrix} \times \begin{bmatrix} \big(G_0 - G(\theta)\big)^\star \\ \big(H_0 - H(\theta)\big)^\star \end{bmatrix} \big(H(\theta)\Lambda H^\star(\theta)\big)^{-1}\Bigg) d\omega \quad (2.122)$$

which can be recast as

$$\theta^* = \arg\min_{\theta\in D_\theta} \frac{1}{2\pi}\int_{-\pi}^{\pi} \operatorname{trace}\Bigg(\Big[\big(G_0 - G(\theta) + B(\theta)\big)\phi_u\big(G_0 - G(\theta) + B(\theta)\big)^\star + \big(H_0 - H(\theta)\big)\big(\Lambda_0 - \phi_{eu}\phi_u^{-1}\phi_{ue}\big)\big(H_0 - H(\theta)\big)^\star\Big] \times \big(H(\theta)\Lambda H^\star(\theta)\big)^{-1}\Bigg) d\omega \quad (2.123)$$

where

$$B(e^{j\omega}, \theta) = \big(H_0(e^{j\omega}) - H(e^{j\omega}, \theta)\big)\phi_{eu}(\omega)\phi_u^{-1}(\omega) \quad (2.124)$$

is a bias term that will vanish only if $\phi_{eu} = 0$, *i.e.*, if the data are collected in open loop so that u and e are uncorrelated, or if the noise model $H(z, \theta)$ is flexible enough so that $\mathcal{S} \in \mathcal{M}$; see (Forssell, 1999).

In the SISO case, these expressions become

$$\bar{V}(\theta) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \phi_\varepsilon(\omega) d\omega \quad (2.125)$$

and

$$\theta^* = \arg\min_{\theta\in D_\theta} \Bigg\{ \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\big|G_0 - G(\theta) + B(\theta)\big|^2}{\big|H(\theta)\big|^2}\phi_u d\omega + \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\big|H_0 - H(\theta)\big|^2}{\big|H(\theta)\big|^2}\frac{\lambda_0\phi_u^r}{\phi_u} d\omega \Bigg\} \quad (2.126)$$

where the second term has been obtained by noting that $\lambda_0 - \frac{|\phi_{ue}|^2}{\phi_u} = \frac{\lambda_0\phi_u^r}{\phi_u}$.

Hence, under the condition that $\mathcal{S} \in \mathcal{M}$, we find that

$$G(\hat{\theta}) \to G_0 \quad \text{and} \quad H(\hat{\theta}) \to H_0 \quad \text{w.p. 1 as } N \to \infty \quad (2.127)$$

[10]The $e^{j\omega}$ or ω arguments of the transfer functions or spectra will often be omitted to simplify the notation.

B. Asymptotic variance in transfer function space. In the MIMO case, the asymptotic covariance of the estimate, as N and n both tend to infinity, is given by

$$\operatorname{cov}\operatorname{col}\Big(\begin{bmatrix}\hat{G}(e^{j\omega}) & \hat{H}(e^{j\omega})\end{bmatrix}\Big) \approx \frac{n}{N}\begin{bmatrix}\phi_u(\omega) & \phi_{ue}(\omega)\\ \phi_{eu}(\omega) & \Lambda_0\end{bmatrix}^{-T} \otimes \phi_v(\omega) \tag{2.128}$$

where $\otimes$ denotes the Kronecker product. In open loop, $\phi_{ue} = 0$ and

$$\operatorname{cov}\operatorname{col}\big(\hat{G}(e^{j\omega})\big) \approx \frac{n}{N}\phi_u^{-T}(\omega) \otimes \phi_v(\omega) \tag{2.129a}$$

$$\operatorname{cov}\operatorname{col}\big(\hat{H}(e^{j\omega})\big) \approx \frac{n}{N}\Lambda_0^{-1} \otimes \phi_v(\omega) \tag{2.129b}$$

In the SISO case, these expressions become

$$\operatorname{cov}\big(\hat{G}(e^{j\omega})\big) \approx \frac{n}{N}\frac{\phi_v(\omega)}{\phi_u(\omega)} \tag{2.130a}$$

$$\operatorname{cov}\big(\hat{H}(e^{j\omega})\big) \approx \frac{n}{N}\left|H_0(e^{j\omega})\right|^2 \tag{2.130b}$$

in open loop while, in closed loop, the covariance of $\hat{G}(e^{j\omega})$ is

$$\operatorname{cov}\big(\hat{G}(e^{j\omega})\big) \approx \frac{n}{N}\frac{\phi_v(\omega)}{\phi_u^r(\omega)} \tag{2.131}$$

where ϕ_u^r is the power spectral density of that part of the input $u(t)$ that originates from the reference $r(t)$. This shows the necessity to have a nonzero exogenous excitation at $r(t)$ to be able to identify the system. An input signal $u(t)$ that would only be generated by feedback of the disturbances through the controller would be useless, as it would yield an estimate with infinite variance.

These results were established in (Ljung, 1985) for the SISO case and in (Zhu, 1989) for the MIMO case. They show the importance of the experiment design (open-loop or closed-loop operation, spectrum of the excitation signal, *etc.*) in the tuning of the modelling error. This question will receive more attention in Chapter 3.

► **Remark.** These asymptotic variance expressions are widely used in practice, although they are not always reliable. It has been shown in (Ninness *et al.*, 1999) that their accuracy could depend on choices of fixed poles or zeroes in the model structure; alternative variance expressions with greatly improved accuracy and which make explicit the influence of any fixed poles or zeroes are given. Observe for instance that the use of a fixed prefilter $L(z)$ during identification amounts to impose fixed poles and/or zeroes in the noise model. In this case, the extended theory of (Ninness *et al.*, 1999) should be used, the more so if the number of fixed poles and zeroes is large with respect to the model order. ◄

C. Asymptotic variance in parameter space. The parameter vector estimate tends to a random vector with normal distribution as the number of data samples N tends to infinity (Ljung, 1999):

$$\hat{\theta} \to \theta^* \qquad \text{w.p. 1 as } N \to \infty \tag{2.132a}$$

$$\sqrt{N}(\hat{\theta} - \theta^*) \in AsN(0,\, P_\theta) \tag{2.132b}$$

where

$$P_\theta = R^{-1} Q R^{-1} \tag{2.132c}$$

$$R = \bar{V}''(\theta^*) > 0 \tag{2.132d}$$

$$Q = \lim_{N\to\infty} N \cdot E\Big(\big[V_N'(\theta^*,\, Z^N)\big] \big[V_N'(\theta^*,\, Z^N)\big]^T \Big) \tag{2.132e}$$

The prediction-error identification algorithms of the MATLAB® Identification Toolbox, used with standard model structures, deliver an estimate $\hat{P}_\theta$ of P_θ.

2.6.6 Classical Model Validation Tools

Once a model $\mathcal{M}(\hat{\theta})$ has been identified, it is necessary to make it pass a range of validation tests to assess its quality. The classical validation tests are the following.

A. Model fit indicator. The simulated output and the one-step-ahead predicted output are respectively given by

$$\hat{y}_s\big(t \mid \mathcal{M}(\hat{\theta})\big) = G(z,\, \hat{\theta}) u(t) \tag{2.133a}$$

and

$$\hat{y}_p\big(t \mid \mathcal{M}(\hat{\theta})\big) = H^{-1}(z,\, \hat{\theta}) G(z,\, \hat{\theta}) u(t) + \big(1 - H^{-1}(z,\, \hat{\theta})\big) y(t) \tag{2.133b}$$

Let us define the following model fit indicator:

$$J_x\big(\mathcal{M}(\hat{\theta})\big) = \frac{1}{N} \sum_{t=1}^{N} \Big| y(t) - \hat{y}_x\big(t \mid \mathcal{M}(\hat{\theta})\big) \Big|^2 \tag{2.134}$$

where x is either p or s depending on whether we are interested in one-step-ahead prediction or in simulation (several-steps-ahead prediction can also be considered: see (Ljung, 1999)). A normalised measure of this fit is given by

$$R_x^2\big(\mathcal{M}(\hat{\theta})\big) = 1 - \frac{J_x\big(\mathcal{M}(\hat{\theta})\big)}{\frac{1}{N}\sum_{t=1}^{N} |y(t)|^2} \tag{2.135}$$

A value of R_x close to 1 means that the predicted or simulated output fits the process output well, *i.e.*, that the observed output variations are well explained by the model, while a value close to 0 means that the model is unable to correctly explain the data.

The quality of the indicator J_x depends very much on the data set that is used to compute it. As a result, it would be nice to be able to evaluate $\bar{J}_x = EJ_x$, where the expectation is taken with respect to the data, the model $\mathcal{M}(\hat{\theta})$ being fixed. It can be shown that J_x will be an unbiased estimate of $\bar{J}_x$ only if it is computed from a different data set that the one used during the identification of the model. Therefore, it is always recommended to use two different data sets: one for estimation, the other for validation. See (Ljung, 1999) for more details.

Finally, note that a model with a good fit in prediction can have a bad fit in simulation. It is easier to find a good model for prediction than for simulation, because prediction takes the noise into account while simulation does not, meaning that the best model obtained by prediction-error identification will always produce a simulation error at least equal to $v(t) = H_0(z)e(t)$, which is usually an auto-correlated signal, while it will produce a prediction error close to $e(t)$ and approximately white provided an appropriate model structure is used. When $H(z, \theta) \equiv 1$, *i.e.*, in the output-error case, the simulation error and prediction error are of course the same. This means that the choice of an OE model structure is ideal when the objective is to find a good simulation model, although the resulting model can be far from optimal in prediction if $H_0(z) \neq 1$.

B. Residuals analysis. Since the objective of prediction-error identification is to make the prediction errors white, it is natural to test this whiteness in order to assess the quality of the model.

Let us define the residuals as

$$\varepsilon(t) = \varepsilon\big(t \mid \mathcal{M}(\hat{\theta})\big) \triangleq y(t) - \hat{y}_p\big(t \mid \mathcal{M}(\hat{\theta})\big) \tag{2.136}$$

The residuals are the prediction errors, ideally built from a validation data set different from the one used during identification as explained above. The experimental auto-correlation function of the residuals is given by

$$\hat{R}_\varepsilon^N(\tau) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t)\varepsilon(t-\tau) \tag{2.137}$$

If $\varepsilon(t)$ is a real white noise sequence of variance λ, $\hat{R}_\varepsilon^N(\tau)$ will tend to 0 for $\tau \neq 0$ and to λ^2 for $\tau = 0$, as N tends to infinity. As a result, a good way to test the whiteness of $\varepsilon(t)$ is to check whether $\hat{R}_\varepsilon^N(\tau)$ is close enough to 0 for $\tau \neq 0$. The MATLAB® Identification Toolbox offers the possibility to plot $\hat{R}_\varepsilon^N(\tau)$ in function of τ, as well as the threshold beyond which it cannot be considered as 'sufficiently close' to zero, and which depends on a probability (confidence) level chosen by the user. More technically, the residuals will be considered as white with probability p if

$$\frac{N}{\big(\hat{R}_\varepsilon^N(0)\big)^2}\sum_{\tau=1}^{M}\big(\hat{R}_\varepsilon^N(\tau)\big)^2 < \chi_p^2(M) \tag{2.138}$$

where M is the number of time lags for which $\hat{R}_\varepsilon^N(\tau)$ is computed, N is the number of data samples used for the computation and $\chi_p^2(M)$ is the p level of the chi-square distribution with M degrees of freedom. Indeed,

$$\begin{aligned} \varepsilon(t) \in N(0,\, \lambda) \quad &\Longrightarrow \quad \sqrt{N}\hat{R}_\varepsilon^N(\tau) \in AsN(0,\, \lambda^2) \\ &\Longrightarrow \quad \frac{N}{\lambda^2}\sum_{\tau=1}^{M}\left(\hat{R}_\varepsilon^N(\tau)\right)^2 \in As\chi^2(M) \end{aligned} \tag{2.139}$$

See (Ljung, 1999) for details.

If the model fails to pass this test, it means that there is more information in the data than explained by the model, *i.e.*, that a higher-order or more flexible model structure should be used. This test will systematically fail if an output-error model structure is used while the real system output is subject to significantly nonwhite disturbances.

The cross-correlation between residuals and past inputs tells us if the residuals $\varepsilon(t)$ still contain information coming from the input signal $u(t)$. It is defined as

$$\hat{R}_{\varepsilon u}^N(\tau) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)u(t-\tau) \tag{2.140}$$

If it is significantly different of 0 for some time lag τ, then there is information coming from $u(t-\tau)$ that is present in $y(t)$ but not in $\hat{y}_p(t \mid \mathcal{M}(\hat{\theta}))$. This means that $G(z,\, \hat{\theta})$ is not representing the transfer from $u(t-\tau)$ to $y(t)$ correctly. This will typically be the case if the system delay or order is incorrectly estimated and it means that something has to be done with the structure of $G(z,\, \theta)$. Incidentally, note that there will never be any correlation between $\varepsilon(t)$ and future inputs $u(t+\tau)$, because of the causality of the system $G_0(z)$, except if the data are collected in closed loop, because $u(t)$ then depends on past outputs, hence on past disturbances $v(t)$ (and hence, $e(t)$). More technically, it can be shown (Ljung, 1999) that

$$\sqrt{N}\hat{R}_{\varepsilon u}^N(\tau) \in AsN(0,\, P), \quad P = \sum_{k=-\infty}^{\infty} R_\varepsilon(k)R_u(k) \tag{2.141}$$

and the uncorrelation test with probability p amounts to check if

$$\left|\hat{R}_{\varepsilon u}^N(\tau)\right| \leq \sqrt{\frac{P}{N}}N_p \tag{2.142}$$

where N_p denotes the p level of the $N(0,\, 1)$ distribution. $R_\varepsilon(k)$ and $R_u(k)$ are the unknown true auto-correlation functions of $\varepsilon(t)$ and $u(t)$, and they can only be approximated from a finite number of data. Once again, the MATLAB® Identification Toolbox offers the possibility to plot $\hat{R}_{\varepsilon u}^N(\tau)$ in function of τ, as well as the threshold beyond which it cannot be considered as 'sufficiently close' to zero and which depends on the confidence level p chosen by the user.

C. Pole-zero cancellations. While the residuals analysis tells us whether the model structure and order are respectively flexible and high enough to capture all the system dynamics, it does not detect a too high model order. In particular, it is important to avoid (near) pole-zero cancellations, as they do generally not represent the reality, are a cause of increased model variance by virtue of the n factor in (2.128) and lead to a (nearly) nonminimal model that may pose control design issues. Pole-zero cancellations can be detected by plotting the poles and zeroes of the model in the complex plane. Any such cancellation means that both the numerator and denominator orders of the model can be reduced by 1.

2.6.7 Closed-loop Identification

When the system G_0 operates in closed loop with some stabilising controller K as in Figure 2.1, it is possible to use closed-loop data for the identification of a model $\hat{G}$. This can be done using different approaches, among which the following three will be used in this book.

A. The indirect approach. This approach, proposed by (Söderström and Stoica, 1989), uses measurements of either $r_1(t)$ or $r_2(t)$ and either $y(t)$ or $u(t)$ to identify one of the four entries of the matrix $T(G_0, K)$ of (2.8). From this estimate, a model $\hat{G}$ for G_0 is derived, using knowledge of the controller, which has to be LTI. Such knowledge is required for this method.

B. The coprime-factor approach. This approach, proposed by (Van den Hof and Schrama, 1995), uses measurements of $r_1(t)$ or $r_2(t)$ and of $y(t)$ and $u(t)$ to identify the two entries of a column of $T(G_0, K)$. A model $\hat{G}$ for G_0 is then given by the ratio of these two entries. This method requires that the controller be LTI, but it can be unknown.

C. The direct approach. This approach consists in identifying a model $\hat{G}$ for G_0 directly from measurements of $u(t)$ and $y(t)$ collected in closed loop.

This is, of course, only a rough description of these approaches. Variants exist (*e.g.*, indirect identification with a tailor-made parametrisation), as well as other methods (*e.g.*, the dual Youla parametrisation method (Hansen, 1989; De Bruyne *et al.*, 1998), which will be used and described in Chapter 4). It has recently been shown that the qualitative properties of the different closed-loop identification methods are essentially equivalent, by observing that these methods can be seen as different parametrisations of one and the same prediction-error method (Forssell and Ljung, 1999).

2.6.8 Data Preprocessing

We cannot conclude this section on system identification without a word on data preprocessing, as it is a crucial phase upon which the whole identification procedure will rest.

As we have said in the beginning of this section, prediction-error identification is based on the assumption that the true system is LTI, which is generally acceptable if variations around a given operating point are considered. It is therefore important to remove the value of this operating point from the input and output signals that are used for identification, in such a way that only small variations around the setpoint are considered. Depending on the process, this operating point will be the initial value of the signals (*e.g.*, if the data used for identification are those of a step response), or its mean, or a slow trend. The removal of the continuous component can be done either algebraically, or by high-pass data filtering. The latter can also be used for data detrending.

The sampling frequency of the data must be chosen carefully and an anti-aliasing filter must be used during the acquisition procedure and during any possible subsequent step of downsampling. If the sampling period is too small with respect to the system time constant, the poles and zeroes of the identified model will drift towards the point $1 + 0j$ of the unit circle in the complex plane, *i.e.*, the obtained model will be close to instability and numerically ill conditioned.

► **Example 2.5.** To illustrate this, consider for instance the first-order continuous-time system $G(s) = \frac{1}{s+1}$. Its time constant is 1 second. If it is sampled with a very small sampling period t_s, the derivative can be approximated by a finite difference, *i.e.*, by setting $s \approx \frac{1-z^{-1}}{t_s}$. Hence, $G(s)$ can be approximated by the discrete-time transfer function $G(z) = \frac{t_s}{(1+t_s)-z^{-1}}$, whose pole tends to $z = 1$ as t_s tends to 0. ◄

On the contrary, if the sampling period is too large, useful signal information will be lost. There are several rules for choosing the appropriate sampling period. (Zhu, 2001) proposes the following:

- to choose $t_s \approx \frac{T_{min}}{3}$, where T_{min} is the smallest time constant of interest;
- to choose $f_s = 10 f_o$, where f_s is the sampling frequency and f_o is the cut-off frequency of the process;
- to choose t_s in the range $\frac{T_{set}}{100} \leq t_s \leq \frac{T_{set}}{20}$, where T_{set} is the settling time of the process.

Other important operations in data preprocessing are peak shaving, removal of outliers, selection of a data set not affected by unmeasured disturbances or

changes in the operating conditions, and division of the data into two subsets, respectively for parameter estimation and for validation.

2.7 Balanced Truncation

Here, we make the assumption that a strictly stable, high-order, continuous-time model $G_n(s)$ of the system is available (see Subsection 2.7.6 for some remarks on the discrete-time case), and that it has the following state-space representation:

$$G_n = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \tag{2.143a}$$

i.e.,

$$G_n(s) = C(sI - A)^{-1}B + D \tag{2.143b}$$

The balanced truncation procedure, which was first proposed in (Moore, 1981), consists in computing a state-space representation of $G_n(s)$ in which the most controllable modes coincide with the most observable ones. The least observable and controllable modes, which have little influence on the input-output behaviour of the model, are then discarded. This procedure can be extended to the case where frequency weightings are used (Enns, 1984*a*, 1984*b*). We shall first briefly review the notions of controllability and observability and then describe the balanced truncation procedure without and with frequency weightings.

2.7.1 The Concepts of Controllability and Observability

A. Controllability.

Definition 2.6. (CONTROLLABILITY) *A state $x_0 \in \mathbb{R}^n$ of the system G_n is called controllable if there exists a control input signal $u(t)$, $t \in [0, T]$, that brings the system from initial state $x(0) = x_0$ to $x(T) = 0$ in a finite time T. The controllable subspace of G_n, $X^{cont} \subseteq \mathbb{R}^n$, is the set of all controllable states of G_n. G_n is called controllable if $X^{cont} = \mathbb{R}^n$.*

The controllability matrix of G_n is

$$\mathbf{C} \triangleq \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \tag{2.144}$$

The columns of $\mathbf{C}$ generate X^{cont}. As a result, G_n (or, equivalently, the pair (A, B)) is controllable if and only if $\operatorname{rank} \mathbf{C} = n$. The controllability of G_n implies that it is always possible to find a finite input signal $u(t)$ that brings the state from any initial condition $x(0) = x_1$ to any desired value $x(T) = x_2$ in a finite time $T > 0$.

The controllability Gramian of G_n at time $0 \leq t < \infty$ is defined as

$$\mathbf{P}(t) \triangleq \int_0^t e^{A\tau} B B^T e^{A^T \tau} d\tau, \qquad t \in \mathbb{R}^+ \tag{2.145}$$

It has the following properties:

- $\forall t \in \mathbb{R}^+ \quad \mathbf{P}(t) = \mathbf{P}^T(t) \geq 0$, *i.e.*, it is symmetric and positive semi-definite;
- $\forall t \in \mathbb{R}^+ \quad \operatorname{im} \mathbf{P}(t) = \operatorname{im} \mathbf{C}$, *i.e.*, the columns of $\mathbf{P}(t)$ generate X^{cont} for all positive t.

For a strictly stable system, the infinite controllability Gramian is defined as

$$\mathbf{P} \triangleq \int_0^\infty e^{A\tau} B B^T e^{A^T \tau} d\tau \tag{2.146}$$

It has the same properties as $\mathbf{P}(t)$. As a result, a stable system G_n is fully controllable if and only if $\mathbf{P} > 0$, *i.e.*, if it is positive definite. This infinite controllability Gramian, which we shall simply call the controllability Gramian in the sequel, can be computed by solving the following Lyapunov equation:

$$A\mathbf{P} + \mathbf{P}A^T + BB^T = 0 \tag{2.147}$$

The controllability Gramian $\mathbf{P}$ gives the minimum input energy that would be necessary to bring the system from the free equilibrium $x(-\infty) = 0$ to a state $x(0) = x_0$:

$$\min_u \left\{ \int_{-\infty}^0 u^T(t) u(t) dt \mid x(0) = x_0 \right\} = x_0^T \mathbf{P}^{-1} x_0 \quad \text{when } x(-\infty) = 0 \tag{2.148}$$

B. Observability.

Definition 2.7. (Observability) *A state $x_0 \in \mathbb{R}^n$ of the system G_n is called unobservable if the free response of the output of G_n to this state is identically zero for all $t \geq 0$, i.e., if this state cannot be distinguished from the zero state. The unobservable subspace of G_n, $\mathbb{C}_{\mathbb{R}^n} X^{obs} \subseteq \mathbb{R}^n$, is the set of all unobservable states of G_n. G_n is called observable if $\mathbb{C}_{\mathbb{R}^n} X^{obs} = \{0\}$ or, equivalently, if $X^{obs} = \mathbb{R}^n_0$.*

The observability matrix of G_n is

$$\mathbf{O} \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{2.149}$$

The null space of $\mathbf{O}$ is $\mathbb{C}_{\mathbb{R}^n} X^{obs}$, hence G_n (or, equivalently, the pair (A, C)) is observable if and only if $\operatorname{rank} \mathbf{O} = n$. G_n is observable means that equal outputs for equal inputs imply equal initial state conditions.

The observability Gramian of G_n at time $0 \leq t < \infty$ is defined as

$$\mathbf{Q}(t) \triangleq \int_0^t e^{A^T(t-\tau)} C^T C e^{A(t-\tau)} d\tau, \qquad t \in \mathbb{R}^+ \tag{2.150}$$

It has the following properties:

- $\forall t \in \mathbb{R}^+ \quad \mathbf{Q}(t) = \mathbf{Q}^T(t) \geq 0$, *i.e.*, it is symmetric and positive semi-definite;
- $\forall t \in \mathbb{R}^+ \quad \ker \mathbf{Q}(t) = \ker \mathbf{O}$, *i.e.*, the null space of $\mathbf{Q}(t)$ is $\complement_{\mathbb{R}^n} X^{obs}$ for all positive t.

For a strictly stable system, the infinite observability Gramian is defined as

$$\mathbf{Q} \triangleq \int_0^\infty e^{A^T \tau} C^T C e^{A\tau} d\tau \tag{2.151}$$

It has the same properties as $\mathbf{Q}(t)$. As a result, G_n is fully observable if and only if $\mathbf{Q} > 0$, *i.e.*, if it is positive definite. This infinite observability Gramian, which we shall simply call the observability Gramian in the sequel, can be computed by solving the following Lyapunov equation:

$$A^T \mathbf{Q} + \mathbf{Q} A + C^T C = 0 \tag{2.152}$$

The observability Gramian $\mathbf{Q}$ gives the total energy of the free output response of the system to an initial state $x(0) = x_0$:

$$\int_0^\infty y^T(t) y(t) dt = x_0^T \mathbf{Q} x_0 \quad \text{when } u(t) = 0 \; \forall t \geq 0 \text{ and } x(0) = x_0 \tag{2.153}$$

2.7.2 Balanced Realisation of a System

The following observations can be made regarding the controllability and observability Gramians:

- the controllability and observability Gramians $\mathbf{P}$ and $\mathbf{Q}$ depend on the state-space realisation of the system G_n;
- their eigenvalues give information about the 'level' of observability or controllability of the state variables;
- depending on the chosen realisation (2.143), some state variables (*i.e.*, some dynamics) can be very observable but little controllable, or vice-versa.

If the realisation (2.143) is minimal, *i.e.*, if it is both controllable and observable ($\mathbf{P} > 0$ and $\mathbf{Q} > 0$), it is possible to find a state transformation that brings the system to a form where the most observable dynamics are also the most controllable ones. This is called a *balanced realisation*. When the system is in balanced form, its Gramians are diagonal and equal:

$$\mathbf{P} = \mathbf{Q} = \Sigma \triangleq \operatorname{diag}(\varsigma_1, \ldots, \varsigma_n) \tag{2.154}$$

where the ς_i's are the Hankel singular values of G_n in decreasing order:

$$\varsigma_i = \sqrt{\lambda_i(\mathbf{PQ})} > 0, \quad \varsigma_i \geq \varsigma_{i+1} \tag{2.155}$$

Observe that, although the Gramians depend on the state realisation of the system, the Hankel singular values are independent of this realisation. Once a balanced realisation has been obtained, it is possible to reduce the order of the system by simply discarding the state variables that correspond to the least observable and controllable dynamics. Indeed, these are the dynamics that contribute the least to the global input-output behaviour of the system.

Starting from any realisation (2.143) of G_n, one can compute a balanced realisation as follows.

1. Compute the controllability and observability Gramians $\mathbf{P}$ and $\mathbf{Q}$ by solving the Lyapunov equations (2.147) and (2.152).
2. Compute
$$\Sigma = \text{diag}(\varsigma_1, \ldots, \varsigma_n) \tag{2.156}$$
where the ς_i's are given by (2.155).
3. Compute R such that
$$\mathbf{P} = R^T R \tag{2.157}$$
It can be obtained as
$$R = \Upsilon \sqrt{\Lambda} \Upsilon^T \tag{2.158}$$
where Λ is a diagonal matrix containing the eigenvalues of $\mathbf{P}$ and Υ is a matrix whose columns are the eigenvectors of $\mathbf{P}$ associated with the entries of Λ.
4. Make a singular-value decomposition of $R\mathbf{Q}R^T$:
$$R\mathbf{Q}R^T = U\Sigma^2 U^T \tag{2.159}$$
5. Then
$$T = \Sigma^{1/2} U^T R^{-T} \tag{2.160}$$
is the balancing transformation matrix and there holds
$$T\mathbf{P}T^T = T^{-T}\mathbf{Q}T^{-1} = \Sigma \tag{2.161}$$
6. Compute
$$\breve{A} = TAT^{-1} \qquad \breve{B} = TB \qquad \breve{C} = CT^{-1} \qquad \breve{D} = D \tag{2.162}$$
Then,
$$\breve{G}_n = \left[\begin{array}{c|c} \breve{A} & \breve{B} \\ \hline \breve{C} & \breve{D} \end{array}\right] \tag{2.163}$$
is a balanced realisation of $G_n(s)$.

2.7.3 Balanced Truncation

The diagonal of Σ contains the Hankel singular values of G_n in decreasing order. The state variables of the balanced realisation $\breve{G}_n$ follow the same order: x_i is more observable and controllable than x_j for $1 \leq i < j \leq n$. More precisely, $\breve{G}_n$ is more observable and controllable in the direction of $\vec{e}_i$ than in the direction of $\vec{e}_j$ for $1 \leq i < j \leq n$. Let us then consider the following partition:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} \tag{2.164a}$$

$$\Sigma_{11} = \operatorname{diag}(\varsigma_1, \ldots, \varsigma_r) \tag{2.164b}$$

$$\Sigma_{22} = \operatorname{diag}(\varsigma_{r+1}, \ldots, \varsigma_n) \tag{2.164c}$$

The corresponding partition of $\breve{G}_n$ is

$$\breve{G}_n = \left[\begin{array}{cc|c} \breve{A}_{11} & \breve{A}_{12} & \breve{B}_1 \\ \breve{A}_{21} & \breve{A}_{22} & \breve{B}_2 \\ \hline \breve{C}_1 & \breve{C}_2 & \breve{D} \end{array}\right] \tag{2.165}$$

Let us define

$$\begin{aligned} \hat{G}_r &= \left[\begin{array}{c|c} \breve{A}_{11} & \breve{B}_1 \\ \hline \breve{C}_1 & \breve{D} \end{array}\right] \\ &\triangleq \operatorname{bt}(G_n, r) \end{aligned} \tag{2.166}$$

which is obtained by truncating the least observable and controllable dynamics of $\breve{G}_n$. The obtained reduced-order system $\hat{G}_r$ is a stable suboptimal solution to the following minimisation problem:

$$\min_{G_r(s) \text{ of order } r} \|G_n(s) - G_r(s)\|_\infty \tag{2.167}$$

which is hard to solve exactly. Upper and lower bounds of the reduction error in the $\mathcal{H}_\infty$ norm can easily be computed from the Hankel singular values of G_n (Glover, 1984), (Enns, 1984*a*, 1984*b*):

$$\varsigma_r \leq \left\| G_n(s) - \hat{G}_r(s) \right\|_\infty \leq 2 \sum_{i=r+1}^{n} \varsigma_i \tag{2.168}$$

The level r of truncation can be chosen by plotting the ς_i's: it is best to have $\varsigma_r >> \varsigma_{r+1}$, which means that the dynamics corresponding to $\varsigma_{r+1}, \ldots, \varsigma_n$ can really be neglected because of their relatively poor degree of observability and controllability. A tighter upper bound in (2.168) can be obtained by counting the ς_i's of multiplicity larger than 1 only once in the summation.

2.7.4 Numerical Issues

The balancing transformation of Subsection 2.7.2 is a frequent source of numerical difficulty, as is often the case with nonorthogonal transformations. There

exist, however, algorithms that compute the reduced-order system $\hat{G}_r(s)$ starting from any realisation of the full-order system $G_n(s)$ without performing explicitly this balancing transformation. The following algorithm, for instance, has been proposed by (Safonov and Chiang, 1989).

1. Starting from the Gramians $\mathbf{P}$ and $\mathbf{Q}$ of any state-space realisation of $G_n(s)$, compute ordered Schur decompositions of the product $\mathbf{PQ}$:
$$V_A^T \mathbf{PQ} V_A = S_A \qquad V_D^T \mathbf{PQ} V_D = S_D \tag{2.169}$$
where S_A and S_D are upper triangular matrices with the eigenvalues of $\mathbf{PQ}$ on their diagonals, respectively in ascending and descending order and V_A and V_D are orthogonal.
2. Compute the following submatrices, where r is the order of the desired reduced-order system $\hat{G}_r(s)$:
$$V_a = V_A \begin{bmatrix} 0_{(n-r)\times r} \\ I_{r\times r} \end{bmatrix} \qquad V_d = V_D \begin{bmatrix} I_{r\times r} \\ 0_{(n-r)\times r} \end{bmatrix} \tag{2.170}$$
3. Compute a singular-value decomposition of $V_a^T V_d$:
$$U_L S U_R^T = V_a^T V_d \tag{2.171}$$
where S is diagonal with positive entries and U_L and U_R are orthogonal.
4. Transformation matrices are obtained as
$$L = S^{-\frac{1}{2}} U_L^T V_a^T \qquad R = V_d U_R S^{-\frac{1}{2}} \tag{2.172}$$
($LR = I$, meaning that the transformation is orthogonal) and a state-space realisation of $G_r(s)$ is given by
$$A_r = LAR \qquad B_r = LB \qquad C_r = CR \qquad D_r = D \tag{2.173}$$

2.7.5 Frequency-weighted Balanced Truncation

A very common case is when the reduction criterion contains stable input and/or output frequency-weighting filters W_r and/or W_l. The minimisation problem becomes

$$\min_{G_r(s) \text{ of order } r} \left\| W_l(s)\big(G_n(s) - G_r(s)\big)W_r(s) \right\|_\infty \tag{2.174}$$

to which a suboptimal solution can be computed by frequency-weighted balanced truncation.

Let us consider any minimal realisation of the stable system $G_n(s)$ defined in (2.143) and realisations of the two filters as follows:

$$W_l = \left[\begin{array}{c|c} A_l & B_l \\ \hline C_l & D_l \end{array}\right] \qquad W_r = \left[\begin{array}{c|c} A_r & B_r \\ \hline C_r & D_r \end{array}\right] \tag{2.175}$$

Then, a realisation of $W_l(s)G_n(s)W_r(s) \triangleq \tilde{G}_n(s) = \tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + \tilde{D}$ is given by

$$\tilde{G}_n = \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right] = \left[\begin{array}{ccc|c} A_l & B_lC & B_lDC_r & B_lDD_r \\ 0 & A & BC_r & BD_r \\ 0 & 0 & A_r & B_r \\ \hline C_l & D_lC & D_lDC_r & D_lDD_r \end{array}\right] \tag{2.176}$$

The controllability and observability Gramians of this input-output frequency-weighted system are respectively the solutions of the following Lyapunov equations:

$$\tilde{A}\tilde{\mathbf{P}} + \tilde{\mathbf{P}}\tilde{A}^T + \tilde{B}\tilde{B}^T = 0 \tag{2.177}$$

$$\tilde{A}^T\tilde{\mathbf{Q}} + \tilde{\mathbf{Q}}\tilde{A} + \tilde{C}^T\tilde{C} = 0 \tag{2.178}$$

These Gramians can be partitioned similarly to $\tilde{A}$ in (2.176):

$$\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} \end{bmatrix} \qquad \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} & \mathbf{P}_{13} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} & \mathbf{P}_{23} \\ \mathbf{Q}_{31} & \mathbf{Q}_{32} & \mathbf{P}_{33} \end{bmatrix} \tag{2.179}$$

$\mathbf{P}_{22}$ and $\mathbf{Q}_{22}$, which correspond to the A block in (2.176), are then the frequency-weighted controllability and observability Gramians of $G_n(s)$. If $W_r(s) \neq I$, then $\mathbf{P}_{22} \neq \mathbf{P}$ given by (2.147), which means that the input weighting filter modifies the controllability Gramian of the system. Similarly, if $W_l(s) \neq I$, then $\mathbf{Q}_{22} \neq \mathbf{Q}$ given by (2.152), which means that the output weighting filter modifies the observability Gramian of the system.

The rest of the procedure consists of finding a transformation matrix T such that $T\mathbf{P}_{22}T^T = T^{-T}\mathbf{Q}_{22}T^{-1} = \Sigma = \mathrm{diag}(\varsigma_1, \ldots, \varsigma_n)$ where the ς_i's are the frequency-weighted Hankel singular values of $G_n(s)$ and to apply this transformation to the realisation (2.143). This produces a frequency-weighted balanced realisation $\breve{G}_n$ of the system $G_n(s)$. Its order can then be reduced, as in unweighted balanced truncation, by discarding the modes corresponding to the smallest Hankel singular values. In the sequel,

$$\hat{G}_r = \mathrm{fwbt}\,(G_n,\, W_l,\, W_r,\, r) \tag{2.180}$$

will denote the r-th order system produced by frequency-weighted balanced truncation of $G_n(s)$. An upper bound of the approximation error in the $\mathcal{H}_\infty$ norm is given by (Kim *et al.*, 1995)

$$\begin{aligned} &\big\|W_l(s)\big(G_n(s) - G_r(s)\big)W_r(s)\big\|_\infty \\ &\qquad \leq 2\sum_{i=r+1}^{n} \sqrt{\varsigma_i^2 + (\alpha_i + \beta_i)\varsigma_i^{3/2} + \alpha_i\beta_i\varsigma_i} \end{aligned} \tag{2.181}$$

where

$$\alpha_i = \|\Xi_{i-1}(s)\|_\infty \left\| C_r \Phi_r(s) \mathbf{P}_{33}^{1/2} \right\|_\infty \tag{2.182a}$$

$$\beta_i = \left\| \mathbf{Q}_{11}^{1/2} \Phi_l(s) B_l \right\|_\infty \|\Gamma_{i-1}(s)\|_\infty \tag{2.182b}$$

$$\Xi_{i-1}(s) = A_{21}^{i-1}(sI - A_{i-1})^{-1} B_{i-1} + b_i \tag{2.182c}$$

$$\Gamma_{i-1}(s) = C_{i-1}(sI - A_{i-1})^{-1} A_{12}^{i-1} + c_i \tag{2.182d}$$

$$\Phi_r(s) = (sI - A_r)^{-1} \tag{2.182e}$$

$$\Phi_l(s) = (sI - A_l)^{-1} \tag{2.182f}$$

$$A_i = \begin{bmatrix} A_{i-1} & A_{12}^{i-1} \\ A_{21}^{i-1} & a_{ii} \end{bmatrix} \tag{2.182g}$$

$$B_i = \begin{bmatrix} B_{i-1} \\ b_i \end{bmatrix} \tag{2.182h}$$

$$C_i = \begin{bmatrix} C_{i-1} & c_i \end{bmatrix} \tag{2.182i}$$

and b_i and c_i are the i-th row of B_i and the i-th column of C_i, respectively, and $A_n = A$, $B_n = B$, $C_n = C$.

Let us finally remark that the reduced-order model $\hat{G}_r$ is guaranteed to be stable if frequency weighting is used at only one side of the reduction criterion.

2.7.6 Balanced Truncation of Discrete-time Systems

The balanced truncation and frequency-weighted balanced truncation procedures are exactly the same in the discrete-time case, except that the controllability and observability Gramians are respectively the solutions of the discrete Lyapunov equations

$$A\mathbf{P}A^T - \mathbf{P} + BB^T = 0 \tag{2.183}$$

and

$$A^T\mathbf{Q}A - \mathbf{Q} + C^TC = 0 \tag{2.184}$$

Observe that, since the $\mathcal{H}_2$ norm of a stable system is upper bounded by its $\mathcal{H}_\infty$ norm in the discrete-time case (Boyd and Doyle, 1987), the $\mathcal{H}_\infty$ upper bound of the approximation error, computed from the Hankel singular values (both in the unweighted and single-side weighted cases), is also an $\mathcal{H}_2$ upper bound of it.