Chapter 9
Fault Accommodation and Reconfiguration Methods

Abstract This chapter gives an overview of methods for re-adjusting the controller to faulty plants. Small faults can be tackled by fault accommodation, where the controller parameters are adapted to the parameters of the faulty plant. When accommodation cannot be used like in the case of an actuator or sensor breakdown, the control loop has to be reconfigured and a new control law designed.

9.1 Fault-Tolerant Model-Matching Design

9.1.1 Reconfiguration Problem

The basic scheme of fault-tolerant control is depicted in Fig. 1.1 on p. 2. At the execution level, a feedback controller

$$\boldsymbol{u}(t) = \boldsymbol{k}(\boldsymbol{y}(t),\ \boldsymbol{y}_{\mathrm{ref}}(t))$$

is used to attenuate the disturbance $\boldsymbol{d}$ and to ensure command tracking with respect to the command input $\boldsymbol{y}_{\mathrm{ref}}$. The control law $\boldsymbol{k}$ is designed so that the closed-loop system satisfies the given requirements for the faultless plant. Before a fault f occurs the supervision level shown in the figure only checks that the plant has its nominal behaviour.

If the diagnostic unit detects and identifies a fault, the adaptation of the controller to the faulty system is accomplished at the supervision level. This process results in new controller parameters and possibly in a new control configuration. If the sensors and actuators work differently as before but the faulty plant is still observable and controllable, the control configuration can remain as before but the controller parameters have to be adapted to the faulty system. This process is called fault accommodation.

However, if the sensor or actuator faults break the control loop, new sensors or actuators, respectively, have to be used. Then, the control loop has to be "reconfigured" in the sense that the whole process of selecting a suitable structure and

M. Blanke et al., *Diagnosis and Fault-Tolerant Control*,
DOI 10.1007/978-3-662-47943-8_9

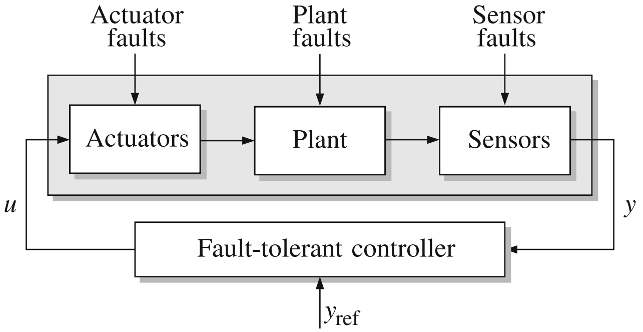


Fig. 9.1 Fault-tolerant controller

appropriate controller parameters has to be repeated after the fault is present. The control problem has to be considered "from the scratch" by appropriately choosing

- the signal vector $\boldsymbol{y}$ to be controlled and the input vector $\boldsymbol{u}$ to be used,
- the control law $\boldsymbol{k}$ including the controller parameters,
- the set-point $\boldsymbol{y}_{\rm ref}$ of underlying control loops.

Control reconfiguration can be thought of as an "analytical repair" of the closed-loop system, where instead of repairing the plant the control algorithm and, hence, the controller software is changed while exploiting the redundant measurement or control signals for satisfying the control specifications in spite of the fault (Fig. 9.1).

To solve the fault-tolerant control problem, it is assumed that a state-space model

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{g}(\boldsymbol{x}(t),\ \boldsymbol{u}(t),\ f), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{9.1}$$

$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t),\ \boldsymbol{u}(t),\ f) \tag{9.2}$$

with state $\boldsymbol{x} \in |\mathcal{R}^n$, input $\boldsymbol{u} \in |\mathcal{R}^m$ and output $\boldsymbol{y} \in |\mathcal{R}^r$ is available, which also describes the dependence of the plant dynamics upon the faults $f \in \mathcal{F}$. Furthermore, it is assumed that a diagnostic algorithm has identified the current fault f.

According to these assumptions, the fault-tolerant control problem can be summarised as follows:

Problem 9.1 (*Fault-tolerant control problem*)

Given: Model (9.1), (9.2) of the plant
Nominal controller $\boldsymbol{k}$
Control specifications
Fault f
Find: Control configuration and new control law $\boldsymbol{k}_f$.

Note that in contrast to the usual controller design problem, also a nominal controller $\boldsymbol{k}$ is given. One of the important aspects of fault-tolerant control is to take advantage of the knowledge of the nominal controller $\boldsymbol{k}$ when solving the control problem stated above.

9.1.2 Pseudo-Inverse Method

One of the earliest methods for the controller redesign is based on model-matching. As the nominal closed-loop system is known, the model of this system can be used as a description of the dynamical properties that the new controller should produce in connection with the faulty plant. That is, the closed-loop system should match the model of the nominal loop.

The idea of model-matching is depicted in Fig. 9.2. The nominal closed-loop system is composed of the linear nominal plant

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{9.3}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \tag{9.4}$$

and a nominal controller, which is assumed to be a state feedback controller $\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t)$. Both components yield the model of the closed-loop system

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\,\boldsymbol{x}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t).$$

If the controller does not use all the inputs u_i of the input vector $\boldsymbol{u}$, the matrix $\boldsymbol{K}$ has zero rows, which is typical for plants with redundant actuators. When the fault f occurs, the faulty plant is given by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_f\boldsymbol{x}(t) + \boldsymbol{B}_f\boldsymbol{u}(t) \tag{9.5}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}_f\boldsymbol{x}(t), \tag{9.6}$$

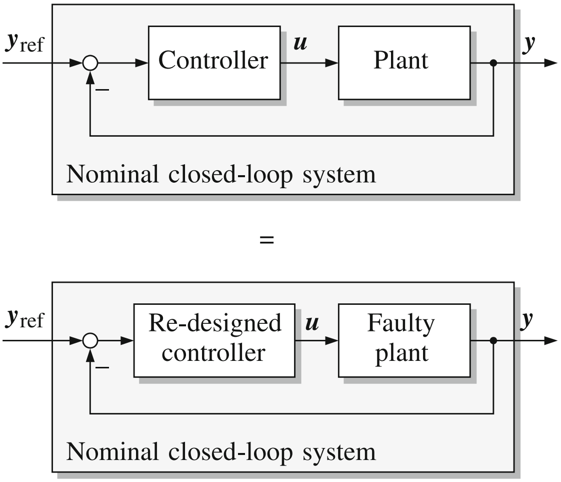


Fig. 9.2 Idea of the model-matching approach to control reconfiguration

where the fault f has changed the system properties, which are now described by the matrices $\boldsymbol{A}_f$, $\boldsymbol{B}_f$ and $\boldsymbol{C}_f$. If the sets of available input or output signals have changed, the matrices $\boldsymbol{B}_f$ and $\boldsymbol{C}_f$ have vanishing columns or rows, respectively. A new state feedback controller

$$\boldsymbol{u}(t) = -\boldsymbol{K}_f \boldsymbol{x}(t)$$

should be found such that the closed-loop system

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= (\boldsymbol{A}_f - \boldsymbol{B}_f \boldsymbol{K}_f)\, \boldsymbol{x}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}_f \boldsymbol{x}(t)\end{aligned}$$

behaves like the nominal loop. For the models used here, model-matching means to satisfy the relation

$$\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K} = \boldsymbol{A}_f - \boldsymbol{B}_f \boldsymbol{K}_f, \tag{9.7}$$

which means that both closed-loop systems have similar dynamics.

Equation (9.7) cannot be satisfied unless $\boldsymbol{B}$ and $\boldsymbol{B}_f$ have the same image (like in the case of a redundant actuator). Therefore, the new controller $\boldsymbol{K}_f$ is chosen so as to minimise the difference

$$\|(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}) - (\boldsymbol{A}_f - \boldsymbol{B}_f \boldsymbol{K}_f)\|. \tag{9.8}$$

The solution to this problem is given by

$$\boldsymbol{K}_f = \boldsymbol{B}_f^{+} \left(\boldsymbol{A}_f - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right) = \left(\boldsymbol{B}_f^{\mathrm{T}} \boldsymbol{B}_f\right)^{-1} \boldsymbol{B}_f^{\mathrm{T}} \left(\boldsymbol{A}_f - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right), \tag{9.9}$$

where $\boldsymbol{B}_f^{+}$ denotes the pseudoinverse of $\boldsymbol{B}_f$ given on the right-hand side of (9.9). Its use provides the reason for the name *pseudo-inverse method* of this approach.

The new controller (9.9) is adapted to the faulty system and minimises the difference (9.8) between the dynamical properties of the nominal loop and the closed-loop system with the faulty plant. Although the controller $\boldsymbol{K}_f$ is the best possible solution to the controller redesign problem, it does not ensure that the closed-loop system behaves satisfactorily. In particular, it does not ensure the stability of the closed-loop system. Therefore, the stability of $\boldsymbol{A}_f - \boldsymbol{B}_f \boldsymbol{K}_f$ and the performance of the control loop have to be evaluated separately. Extensions of this method ensure the stability without a separate test.

Fault accommodation and control reconfiguration. The method described so far is rather general. It includes both fault accommodation and control reconfiguration. Depending on the sensors and the actuators used, the controller is simply adapted to the new plant dynamics or it uses sensors or actuators that have not been used in the nominal case. In the latter case, vanishing rows in the nominal controller $\boldsymbol{K}$ are replaced by non-zero elements, which means that new actuators are used and, hence, a new control configuration results.

9.1.3 Model-Matching Control for Sensor Failures

This section considers the case of complete sensor failures. If the ith sensor fails, the output y_i is set to zero. In the plant model the matrix $\boldsymbol{C}$ changes to $\boldsymbol{C}_f$, whose ith row is zero, but the other matrices remain the same as in the nominal case. The corresponding reconfiguration problem will be investigated here for output feedback

$$\boldsymbol{u}(t) = -\boldsymbol{K}\,\boldsymbol{y}(t),$$

for which the nominal closed-loop system is described by

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{C})\,\boldsymbol{x}(t)\\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t).\end{aligned}$$

For the faulty plant, the new controller

$$\boldsymbol{u}(t) = -\boldsymbol{K}_f\,\boldsymbol{y}_f(t)$$

should be found such that the closed loop

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_f\,\boldsymbol{C}_f)\,\boldsymbol{x}(t)\\ \boldsymbol{y}_f(t) &= \boldsymbol{C}_f\,\boldsymbol{x}(t)\end{aligned}$$

has the same dynamics as the nominal loop.

The controller has to satisfy the simplified version of Eq. (9.7)

$$\boldsymbol{K}_f\,\boldsymbol{C}_f = \boldsymbol{K}\boldsymbol{C}. \tag{9.10}$$

To find an appropriate matrix $\boldsymbol{K}_f$ is possible only if the condition

$$\text{Kern}(\boldsymbol{C}_f) \subseteq \text{Kern}(\boldsymbol{C}) \tag{9.11}$$

is satisfied, where Kern denotes the kernel[1] of a matrix. The condition means that the measurement information obtained by the full output vector $\boldsymbol{y}$ is the same as the information obtained by the remaining sensors through $\boldsymbol{y}_f$. The condition (9.11) can be written in an equivalent form as

[1] The kernel of $\boldsymbol{C}$ is the set of vectors $\boldsymbol{x}$ for which $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{0}$ holds.

$$\text{rank}\ \boldsymbol{C}_f = \text{rank}\begin{pmatrix}\boldsymbol{C}\\ \boldsymbol{C}_f\end{pmatrix}.$$

Lemma 9.1 *In case of sensor failures, exact model-matching can be reached if the relation (9.11) holds. Then, the controller*

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{P}\boldsymbol{y}(t) \tag{9.12}$$

solves the reconfiguration problem where

$$\boldsymbol{P} = \boldsymbol{C}\boldsymbol{C}_f^{+} = \boldsymbol{C}\boldsymbol{C}_f^{\mathrm{T}}\left(\boldsymbol{C}_f\boldsymbol{C}_f^{\mathrm{T}}\right)^{-1} \tag{9.13}$$

satisfies the relation

$$\boldsymbol{C} = \boldsymbol{P}\boldsymbol{C}_f. \tag{9.14}$$

The reconfigured controller $\boldsymbol{K}_f = \boldsymbol{K}\boldsymbol{P}$ produces a closed-loop system that has exactly the same properties as the faultless closed-loop system.

Situations where the requirement (9.11) is satisfied include the following:

- The fault has changed the sensitivity of the sensor, but the signal is not completely lost. Hence, $\boldsymbol{y}_f = a\ \boldsymbol{y}$ holds for some scalar a.
- A sensors is at fault which has at least one parallel redundant sensor. The matrix $\boldsymbol{P}$ switches the output to the redundant sensor.
- An analytic relation between the faulty output and several other output values exists, which can be reformulated by using the matrix $\boldsymbol{P}$.

The later two cases are only possible if $\boldsymbol{C}$ does not have full rank, which is likely in special applications only.

9.1.4 Model-Matching Control for Actuator Failures

In case of an actuator failure, the matrix $\boldsymbol{B}$ is replaced by the matrix $\boldsymbol{B}_f$ with zero column for the failing actuator. The output feedback

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{y}(t)$$

which leads to the closed-loop system

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \left(\boldsymbol{A} - \boldsymbol{B}_f\,\boldsymbol{K}\boldsymbol{C}\right)\,\boldsymbol{x}(t)\\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t).\end{aligned}$$

should be replaced by a new controller

$$\boldsymbol{u}_f(t) = -\boldsymbol{K}_f\, \boldsymbol{y}(t)$$

such that the closed loop

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= (\boldsymbol{A} - \boldsymbol{B}_f\, \boldsymbol{K}_f\, \boldsymbol{C})\, \boldsymbol{x}(t)\\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t)\end{aligned}$$

has the same dynamics as the nominal loop.

The controller has to satisfy the simplified version of Eq. (9.7)

$$\boldsymbol{B}_f\, \boldsymbol{K}_f = \boldsymbol{B}\boldsymbol{K}. \qquad (9.15)$$

A solution $\boldsymbol{K}_f$ to this equation exists only if the condition

$$\text{Im}\ (\boldsymbol{B}_f) \supseteq\ \text{Im}\ (\boldsymbol{B}) \qquad (9.16)$$

holds, where Im denotes the image[2] of a matrix. An equivalent formulation of the condition (9.16) is given by

$$\text{rank}\ \boldsymbol{B}_f = \text{rank}\ \left(\boldsymbol{B}\ \ \boldsymbol{B}_f\right).$$

Lemma 9.2 *In case of actuator failures, exact model-matching is possible if Eq. (9.16) holds. Then, the reconfigured controller is given by*

$$\boldsymbol{u}(t) = -\boldsymbol{N}\boldsymbol{K}\boldsymbol{y}(t), \qquad (9.17)$$

where

$$\boldsymbol{N} = \boldsymbol{B}_f^{+}\boldsymbol{B} = \left(\boldsymbol{B}_f^{\mathrm{T}}\boldsymbol{B}_f\right)^{-1}\boldsymbol{B}_f^{\mathrm{T}}\boldsymbol{B} \qquad (9.18)$$

is a matrix satisfying the relation

$$\boldsymbol{B}_f\boldsymbol{N} = \boldsymbol{B}. \qquad (9.19)$$

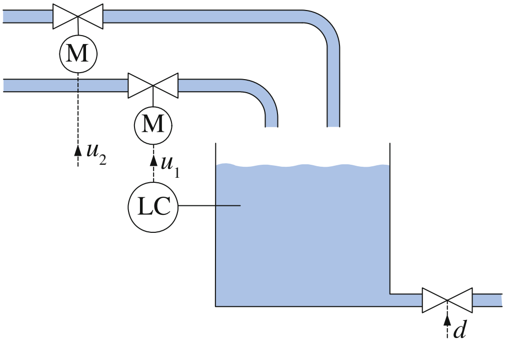
The new controller $\boldsymbol{K}_f = \boldsymbol{N}\boldsymbol{K}$ yields a closed-loop system with exactly the same properties as the nominal loop.

Example 9.1 Model-matching for actuator failures

This example demonstrates the model-matching approach for actuator failures and shows the main idea and a situation in which this approach fails.

[2] The image of $\boldsymbol{C}$ is the set of vectors $\boldsymbol{y}$, for which a vector $\boldsymbol{x}$ exists such that $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$ holds.

Fig. 9.3 Example demonstrating the model-matching reconfiguration strategy



Consider the tank system shown in Fig. 9.3 which has two input pipes. Obviously, for level control, only one pipe is necessary as control input and the redundant input can be used in case of an actuator failure.

Assume first, that the valve positions are used directly as the control inputs. Then the system can be described by a state-space model (9.3), (9.4) where the matrix

$$\boldsymbol{B} = (\boldsymbol{b} \;\; k\boldsymbol{b})$$

has two linearly depending columns because the two inputs influence the system in the same way and the effects of the two actuators distinguish only with respect to some constant factor k. In the nominal system, the first control input is used:

$$u_1(t) = u_C(t) = -\boldsymbol{K}\boldsymbol{y}(t)$$

for some controller $\boldsymbol{K}$ and some output $\boldsymbol{y}$ of the tank system.

If the corresponding actuator fails, the controller should be switched to the second input, where

$$\boldsymbol{B}_f = (\boldsymbol{0} \;\; k\boldsymbol{b})$$

holds. The model-matching solutions yields the (2, 1)-element of the matrix $\boldsymbol{N}$

$$\boldsymbol{N}_{21} = (k^2\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b})^{-1}k\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b} = \frac{1}{k},$$

which means that the output $u_C(t)$ of the nominal controller is transformed into the input

$$u_2(t) = \frac{1}{k}u_C(t)$$

to the second actuator. This is an obvious solution: As the gain of the new actuator is k-times the gain of the old one, the old input u_C is multiplied by $\frac{1}{k}$. A perfect reconfiguration results.

Now change the situation by including the motors for the valves as shown in Fig. 9.3. As these motors have integral dynamics, two additional states have to be added to the state

$$\tilde{\boldsymbol{x}} = \begin{pmatrix} x_{a1} \\ x_{a2} \\ \boldsymbol{x} \end{pmatrix}$$

such that the model now reads as

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}}(t) &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \boldsymbol{b} & k\boldsymbol{b} & \boldsymbol{A} \end{pmatrix} \tilde{\boldsymbol{x}}(t) + \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} \\ \boldsymbol{y}(t) &= (\boldsymbol{O} \quad \boldsymbol{O} \quad \boldsymbol{C})\, \tilde{\boldsymbol{x}}(t) \end{aligned}$$

In principle, the same solution as before is possible. However, the model-matching approach yields for

$$\boldsymbol{B} = \begin{pmatrix} 1 \\ 0 \\ \boldsymbol{0} \end{pmatrix} \quad \text{and} \quad \boldsymbol{B}_f = \begin{pmatrix} 0 \\ 1 \\ \boldsymbol{0} \end{pmatrix}$$

the solution

$$\boldsymbol{N}_{21} = \left(\begin{pmatrix} 0 \\ 1 \\ \boldsymbol{0} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 0 \\ 1 \\ \boldsymbol{0} \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \\ \boldsymbol{0} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 1 \\ 0 \\ \boldsymbol{0} \end{pmatrix} = \boldsymbol{O},$$

where the pseudo-inverse matrix has been built after the zero columns for the no longer available inputs have been deleted. Hence, there is no control input at all. The model-matching approach fails.

The reason for this result lies in the fact that the model-matching idea tries to reproduce the effect $\boldsymbol{B}\boldsymbol{u}$ of the nominal controller by the reconfigured controller $\boldsymbol{B}_f \boldsymbol{N}\boldsymbol{u}$. This is impossible in this example, because the nominal controller has a direct effect only on the state variable x_{a1} an no effect at all on the state variable x_{a2} whereas the redundant input leads to the reverse situation. Hence, no choice of $\boldsymbol{N}$ can reproduce any of the effects of the nominal input. The failure of the model-matching approach lies in this idea and can be circumvented by extending the model-matching aim to the whole plant as described below. □

9.1.5 Markov Parameter Approach to Control Reconfiguration for Actuator Failures

The model-matching approach using the pseudo-inverse of the input matrix fails because if concentrates on the forcing action at point Ⓟ in Fig. 9.4. In the approach shown in this section, the goal refers to the I/O-behaviour of the plant. By this formulation, analytical redundancies become amenable which are based on internal couplings via the system matrix on the one hand and the selection of relevant states via the output matrix on the other hand, see point Ⓠ in the figure. Such redundancies are hidden from a forcing action perspective.

The *Markov parameters*

$$\boldsymbol{G}_i = \boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{B}, \quad i = 1, \ldots, n \tag{9.20}$$

completely describe the I/O-behaviour of a linear system (9.3), (9.4) in terms of its transfer function

$$\boldsymbol{P}(s) = \sum_{i=0}^{\infty} \boldsymbol{G}_i s^{-i}. \tag{9.21}$$

The Markov parameter-based approach to control reconfiguration tries to recover the nominal plant Markov parameters after an actuator failure by using the static reconfiguration block

$$\boldsymbol{u}_{\mathrm{c}}(t) = \boldsymbol{N}\boldsymbol{u}_{f}(t). \tag{9.22}$$

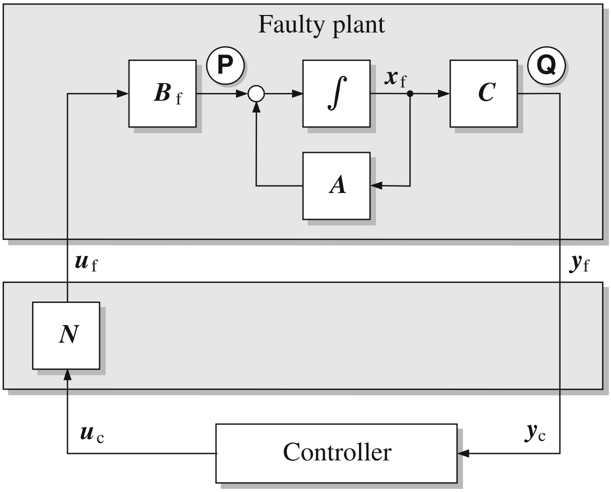

Faulty plant
$\boldsymbol{B}_{\mathrm{f}}$
P
$\int$
$\boldsymbol{x}_{\mathrm{f}}$
$\boldsymbol{C}$
Q
$\boldsymbol{A}$
$\boldsymbol{u}_{\mathrm{f}}$
$\boldsymbol{y}_{\mathrm{f}}$
$\boldsymbol{N}$
$\boldsymbol{u}_{\mathrm{c}}$
$\boldsymbol{y}_{\mathrm{c}}$
Controller

Fig. 9.4 Input/output-based reconfiguration after actuator failures

If the Markov parameters of a reconfigured plant match those of the nominal plant (9.3), (9.4) exactly, the dynamical I/O-behaviour is recovered exactly, which is both necessary and sufficient for successful static I/O-reconfiguration.

With the observability matrix

$$\boldsymbol{S}_O = \begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{n-1} \end{pmatrix} \in |\mathcal{R}^{n\cdot m\times n} \tag{9.23}$$

the design problem to Markov parameter recovery can be posed as

$$\boldsymbol{N} = \arg\min_{\boldsymbol{N}} \|\boldsymbol{S}_O\boldsymbol{B}_f\boldsymbol{N} - \boldsymbol{S}_O\boldsymbol{B}\| \tag{9.24}$$

with the solution

$$\boldsymbol{N} = (\boldsymbol{S}_O\boldsymbol{B}_f)^+\boldsymbol{S}_O\boldsymbol{B}. \tag{9.25}$$

If the condition

$$\text{Im}\,(\boldsymbol{S}_O\boldsymbol{B}_f) \supseteq \text{Im}\,(\boldsymbol{S}_O\boldsymbol{B}) \tag{9.26}$$

holds, then perfect I/O-reconfiguration results in the sense that all Markov parameters are exactly recovered. This condition is equivalent to

$$\text{rank}\,(\boldsymbol{S}_O\boldsymbol{B}_f) = \text{rank}\,(\boldsymbol{S}_O\boldsymbol{B}_f \;\; \boldsymbol{S}_O\boldsymbol{B}). \tag{9.27}$$

If this condition is violated, an approximate solution is obtained in this way which matches the original Markov parameters as closely as possible.

Lemma 9.3 *In case of actuator failures, exact model-matching with respect to the I/O-behaviour can be reached if the condition (9.26) holds. Then the reconfigured controller is given by*

$$\boldsymbol{u}(t) = -\boldsymbol{N}\boldsymbol{K}\boldsymbol{y}(t), \tag{9.28}$$

where

$$\boldsymbol{N} = (\boldsymbol{S}_O\boldsymbol{B}_f)^+\boldsymbol{S}_O\boldsymbol{B} \tag{9.29}$$

is a matrix satisfying the relation

$$\boldsymbol{C}\boldsymbol{A}^{(i-1)}\boldsymbol{B}_f\boldsymbol{N} = \boldsymbol{C}\boldsymbol{A}^{(i-1)}\boldsymbol{B}, \quad i = 1, \ldots, n. \tag{9.30}$$

The new controller yields a closed-loop system with exactly the same I/O-behaviour as the nominal loop.

Remark 9.1 (*Generality of the method*) The approach is valid in connection with any controller, since the plant I/O-response is recovered and the fault is hidden from the controller. If the nominal loop was internally stable, this property is preserved under reconfiguration if condition (9.26) holds, as an analysis using the Kalman decomposition reveals. □

Example 9.1 (cont.) Model-matching for actuator failures: Markov approach
The example is now solved using the Markov parameter approach. It is shown that the problems of the pseudo-inverse method are overcome.

The construction of the observability matrix (9.23) yields

$$\boldsymbol{S}_O \boldsymbol{B} = (\gamma \;\; k\gamma)\boldsymbol{b} \;\; \text{with} \;\; \gamma = (\boldsymbol{C} \;\; \boldsymbol{CA} ...)^{\mathrm{T}}, \tag{9.31}$$

whereas after the fault the relation

$$\boldsymbol{S}_O \boldsymbol{B}_f = (\boldsymbol{0} \;\; k\gamma)\boldsymbol{b} \tag{9.32}$$

holds. Condition (9.26) is met and the admissible solution to the problem

$$\boldsymbol{S}_O \boldsymbol{B}_f \boldsymbol{N} = \boldsymbol{S}_O \boldsymbol{B} \tag{9.33}$$

is found using Eq. (9.25) as

$$\boldsymbol{N} = \begin{pmatrix} 0 & 0 \\ \frac{1}{k} & 1 \end{pmatrix}. \tag{9.34}$$

As expected, the control input meant for the first valve is redirected to the second valve with the correct gain adjustment. □

Example 9.2 Markov parameter approach applied to the two-tank example
The plant consists of the two tanks T_1 and T_2 interconnected by valves u_L, u_H, where T_1 is filled via pump u_{P} as shown in Fig. 9.5. Valves are electromechanically driven with the motor states v_L, v_H. The controlled quantities are the levels h_1 and h_2. With the state $\boldsymbol{x} = (v_L, \; v_H, \; h_1, \; h_2)^{\mathrm{T}}$, the tank system is described by the linear model (9.3), (9.4) with

$$\boldsymbol{A} = 10^3 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -3.2 & -3.4 & -7.1 & 3.6 \\ 3.2 & 3.4 & 7.1 & -18 \end{pmatrix}$$

$$\boldsymbol{B} = 10^3 \begin{pmatrix} 0 & 10^{-3} & 0 \\ 0 & 0 & 10^{-3} \\ 8.1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \boldsymbol{B}_f = 10^3 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 10^{-3} \\ 8.1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{C} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and controlled by two decentralised proportional controllers *LC*.

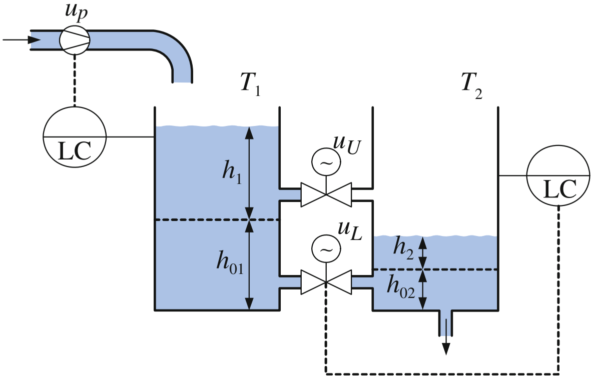


Fig. 9.5 Reconfiguration of a two-tank system

After a blocking of the lower valve at fault time t_f, which yields $u_L(t) = 0$ for $t \geq t_\mathrm{f}$, the plant is statically I/O-reconfigurable according to the condition (9.27). The reconfiguration (9.29) yields

$$\boldsymbol{N} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.9167 & 1 \end{pmatrix}.$$

The behaviour of the successfully reconfigured plant with fault f occurring at $t_f = 250$ s is shown in Fig. 9.6. After the fault appearing at $t_f = 250$ s and the reconfiguration accomplished at $t = 260$ s, the control action is redirected from the lower to the upper valve. This action appears logical, but it cannot be found by the pseudo-inverse method. □

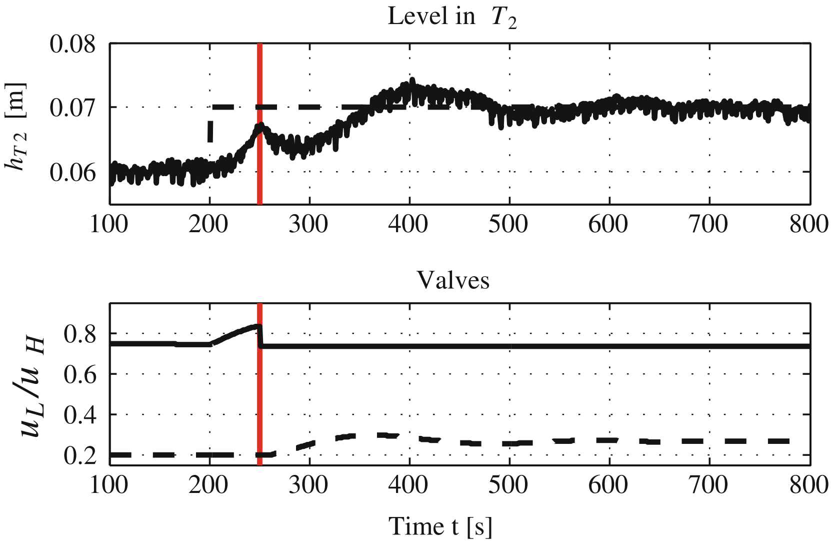


Fig. 9.6 Experimental results with the reconfigured tank system: After the failure of the lower valve (u_L, *solid line*) the controller acts at the upper valve (u_H, *dashed line*)

9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.1 *The Idea of Virtual Sensors and Virtual Actuators*

Severe faults such as the complete failure of actuators or sensors open the control loop with the nominal controller. In order to hold the system in operation, it is necessary to use a different set of input or output signals to accomplish the control task. Once the new control configuration is selected, new controller parameters have to be found. The goal of the reconfiguration is to stabilise the faulty process and to keep it operational with sufficient performance.

Figure 9.7 shows the main idea of the methods explained in this section. Instead of adapting the controller to the faulty plant, a reconfiguration block is used to adapt the faulty plant to the nominal controller. The faulty plant together with the reconfiguration block should produce, for a given input $\boldsymbol{u}_{\mathrm{c}}$, the same (or approximately the same) output $\boldsymbol{y}_{\mathrm{c}}$ as the nominal plant. Hence, the controller "sees" the same plant as before and reacts in the same way as before.

This solution of the reconfiguration problem tries to apply a minimal change to the control loop. In particular, the nominal controller remains an unchanged block of the control loop. The rationale for keeping the controller as before is given by the fact that the existing control law includes valuable implicit knowledge about the process and the possible performance of the closed-loop system. This knowledge was acquired during the design cycle and is not represented in the process model. For example, during the design it became obvious, which control objectives (like overshoot, bandwidth, settling time) can be met with reasonable control effort and which not. The trade-off between the different control objectives is represented by the nominal controller.

In case of a sensor breakdown, the reconfiguration block results from the application of a Luenberger observer to reconstruct the immeasurable output. It is called

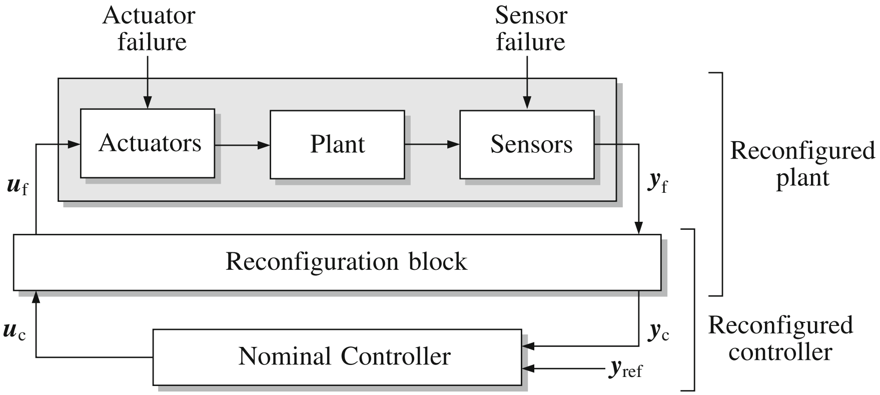

Actuator failure
Sensor failure
Actuators
Plant
Sensors
$\boldsymbol{u}_{\mathrm{f}}$
$\boldsymbol{y}_{\mathrm{f}}$
Reconfigured plant
Reconfiguration block
$\boldsymbol{u}_{\mathrm{c}}$
$\boldsymbol{y}_{\mathrm{c}}$
Nominal Controller
$\boldsymbol{y}_{\mathrm{ref}}$
Reconfigured controller

Fig. 9.7 Principle of control reconfiguration for actuator or sensor failures

a “virtual sensor”, because it reconstructs that element y_i of the output vector $\boldsymbol{y}_{\mathrm{c}}$ from the other measured output signals that the faulty sensor does no longer measure. If an actuator becomes faulty, the reconfiguration block is obtained in a dual way. The reconfiguration block is called a “virtual actuator”, because it acts like the faulty actuator but replaces the effect of this actuator by using the control input of the other actuators appropriately. The reconfigured controller, which is to be applied to the faulty plant, consists of the nominal controller and the reconfiguration block (Fig. 9.7).

The way to find appropriate reconfiguration blocks, which will be described in this section, uses an alternative interpretation of the reconfigured control loop: The faulty process and the reconfiguration block together are called the reconfigured plant, which is connected to the nominal controller. If the reconfigured plant behaves like the nominal plant, the loop consisting of the reconfigured plant and the controller behaves like the nominal closed-loop system. This is true for an arbitrary nominal controller.

Example 9.3 Two-tank reconfiguration problem

The reconfiguration problem and a way of its solution are illustrated by the two coupled tanks depicted in Fig. 9.8.

The main mission of the system is to store water at a certain level in the right tank for some consumer. During the nominal operation there exists two level controllers, with the set-points $y_{1\mathrm{ref}}$ and $y_{2\mathrm{ref}}$. The right controller uses the upper valve, whose position is given by the input u_2. A redundant control input is provided by the lower valve with input signal u_3. In the nominal case, the valve V_{12} is closed. The right controller has to attenuate the disturbance d and to hold the tank level at a given value $y_{2\mathrm{ref}}$. The control specifications include the stability, the set-point following requirement and the specification that the command step response should not have a large overshoot.

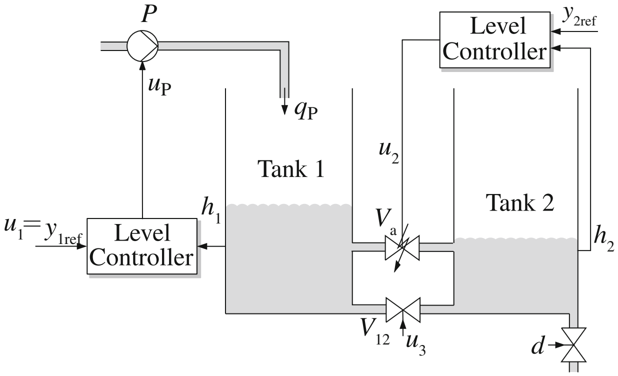


Fig. 9.8 Reconfiguration problem for the tank example

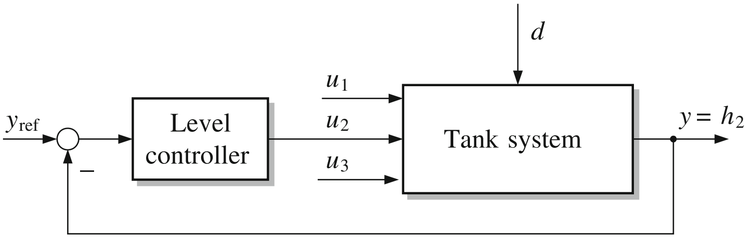


Fig. 9.9 Block diagram of the reconfiguration problem

For the reconfiguration problem, three actuator faults are considered:

- Valve V_a is closed and blocked.
- Valve V_a is open and blocked.
- A level sensor is faulty.

In these cases, one of the two control loops does no longer work. The reconfiguration task consists in finding a new control structure by selecting appropriate actuators, new control laws and new set-points for the control loops such that the control aims described above are obtained (Fig. 9.9).

Obviously, the reconfiguration task cannot be solved by simply changing the parameters of the given controllers, but a structural change of the control configuration is necessary:

- If Valve V_a is closed and blocked, the level controller of the right tank has to use the lower valve V_{12} as control input. In this case, the controller of the left tank can remain unchanged.
- If Valve V_a is open and blocked, in addition to the change of the level controller of the right tank as before, the set-point of the level controller of the left tank has to be set to a value which is lower then the position of Valve V_a. Another possibility is to use the set-point of the level controller of the left tank as control input of the level controller of the right tank.
- In case of the sensor fault, the missing sensor reading has to be reconstructed by means of the remaining output measurements.

All these solutions, which for this simple example seem to be obvious, have to be found automatically by a fault-tolerant control algorithm. □

9.2.2 Reconfiguration Problem

Before explaining the reconfiguration method, the problem to be solved is formally stated. The model of the nominal process is given in state-space form:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{E}\boldsymbol{d}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{9.35}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t). \tag{9.36}$$

The standard model is extended by the disturbance $\boldsymbol{d} \in |\mathcal{R}^p$.

It is important that the process model includes all available input and output signals including those that are not used by the nominal controller. Unlike in the traditional design problem, $\boldsymbol{B}$ and $\boldsymbol{C}$ may not have full rank.

The nominal process is stabilised by a nominal controller with output $\boldsymbol{u}(t)$ and inputs $\boldsymbol{y}(t)$ and $\boldsymbol{y}_{\mathrm{ref}}(t)$. The reconfiguration method explained here can be applied without further assumptions on the controller, which may have arbitrary dynamics and even be nonlinear. However, to demonstrate the properties of the resulting control loop a linear feedback controller

$$\boldsymbol{u}_{\mathrm{c}}(t) = -\boldsymbol{K}\boldsymbol{y}_{\mathrm{c}}(t) + \boldsymbol{V}\boldsymbol{y}_{\mathrm{ref}}(t) \tag{9.37}$$

is used.

Process and controller form the nominal control loop for $\boldsymbol{u}(t) = \boldsymbol{u}_{\mathrm{c}}(t)$ and $\boldsymbol{y}(t) = \boldsymbol{y}_{\mathrm{c}}(t)$:

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{C})\,\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{V}\boldsymbol{y}_{\mathrm{ref}}(t) + \boldsymbol{E}\boldsymbol{d}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{9.38}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t). \tag{9.39}$$

This control loop is assumed to be stable and to satisfy the performance requirements concerning set-point tracking and disturbance rejection.

Fault cases. In the case that the fault f indicates a loss of sensor i, the ith row of the matrix $\boldsymbol{C}$ is changed into zeros and the new matrix is denoted by $\boldsymbol{C}_f$. If the jth actuator fails, the jth column of the matrix $\boldsymbol{B}$ is set to zero and the resulting matrix denoted by $\boldsymbol{B}_f$. In this way, the number of input signals, output signals and state variables is not changed in the model, though some of them may have lost their function. It is assumed that the faulty process is still controllable and observable. This assumption implies that a stabilising controller exists. The input and the output of the faulty plant are denoted by $\boldsymbol{u}_f$ or $\boldsymbol{y}_f$, respectively.

Reconfiguration task. The aim is to find a reconfigured controller that makes the closed-loop system satisfy the following conditions, which, depending on the control task, refer to the autonomous behaviour, reference tracking and disturbance rejection:

- **Strong reconfiguration goal**:
  The controller should make the reconfigured control loop behave in exactly the same way as the nominal control loop, i.e. the relation

  $$\boldsymbol{y}_f(t) = \boldsymbol{y}(t)$$

  should hold for any $\boldsymbol{d}(t)$, $\boldsymbol{y}_{\mathrm{ref}}(t)$ and $\boldsymbol{x}_0$.

It will be demonstrated that this strong goal is only feasible in very special cases. Therefore, a weaker goal is defined in terms of the dynamical and the static behaviour of the reconfigured loop.

- **Weak reconfiguration goal**:
  The weak goal consists of a static and a dynamical part. Considering the static behaviour, the output $\boldsymbol{y}_f$ of the reconfigured loop should have the same value as for the nominal system. This means that for constant values of $\boldsymbol{y}_{\text{ref}}$ and $\boldsymbol{d}$, the relation
  $$\boldsymbol{y}_f(t) \rightarrow \boldsymbol{y}(t) \text{ for } t \rightarrow \infty$$
  should hold. The transient behaviour is determined by the poles and zeros of the system which should not differ significantly in the nominal and the reconfigured control loop. This requirement applies for the autonomous, the disturbance, and the command following behaviour of the reconfigured loop. Additional poles (and zeros) are allowed only if they are fast enough not to dominate the system behaviour.

9.2.3 Virtual Sensor

This section describes a reconfiguration block that reconstructs a measurement y_i from the remaining sensor signals after the ith sensor is no longer available. The main idea is to use an observer for the faulty system, which represents the main part of the reconfiguration block to be built. This block is called *virtual sensor* due to its function of replacing a broken sensor.

The plant with faulty sensor is described by the state-space model

$$\dot{\boldsymbol{x}}_f(t) = \boldsymbol{A}\boldsymbol{x}_f(t) + \boldsymbol{B}\boldsymbol{u}_f(t) + \boldsymbol{E}\boldsymbol{d}(t), \quad \boldsymbol{x}_f(0) = \boldsymbol{x}_{f0} \tag{9.40}$$
$$\boldsymbol{y}_f(t) = \boldsymbol{C}_f\boldsymbol{x}_f(t), \tag{9.41}$$

where the sensor failure is reflected by the matrix $\boldsymbol{C}_f$. If the condition (9.11) is satisfied, the complete output vector $\boldsymbol{y}$ can be reconstructed from $\boldsymbol{y}_f$ and the reconfigured controller (9.12) can be used. This new control structure can be interpreted as consisting of a reconfiguration block

$$\begin{aligned}\boldsymbol{y}_{\text{c}}(t) &= \boldsymbol{P}\boldsymbol{y}_f(t) + \boldsymbol{y}_{\Delta}\\ \boldsymbol{u}_f(t) &= \boldsymbol{u}_{\text{c}}(t)\end{aligned}$$

and the nominal controller. That is, under the condition (9.11) the virtual sensor is a static reconfiguration block.

In the following, the general case is considered, where the condition (9.11) is violated. Then, the reconfiguration block includes a state observer and a direct feedthrough:

Definition 9.1 (*Virtual sensor*) Consider the plant (9.40), (9.41) with faulty sensor. The virtual sensor is defined as the system

$$\dot{\boldsymbol{x}}_{\mathrm{V}}(t) = \boldsymbol{A}_{\mathrm{V}}\boldsymbol{x}_{\mathrm{V}}(t) + \boldsymbol{B}_{\mathrm{V}}\boldsymbol{u}_{\mathrm{c}}(t) + \boldsymbol{L}\boldsymbol{y}_f(t), \quad \boldsymbol{x}_{\mathrm{V}}(0) = \boldsymbol{x}_{\mathrm{V}0} \tag{9.42}$$

$$\boldsymbol{u}_f(t) = \boldsymbol{u}_{\mathrm{c}}(t) \tag{9.43}$$

$$\boldsymbol{y}_{\mathrm{c}}(t) = \boldsymbol{C}_{\Delta}\boldsymbol{x}_{\mathrm{V}}(t) + \boldsymbol{P}\boldsymbol{y}_f(t) \tag{9.44}$$

with the state $\boldsymbol{x}_{\mathrm{V}} \in |\mathcal{R}^n$ and with matrices

$$\boldsymbol{A}_{\mathrm{V}} = \boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}_f \tag{9.45}$$

$$\boldsymbol{B}_{\mathrm{V}} = \boldsymbol{B} \tag{9.46}$$

$$\boldsymbol{C}_{\mathrm{V}} = \boldsymbol{C} - \boldsymbol{P}\boldsymbol{C}_f. \tag{9.47}$$

$\boldsymbol{P}$ and $\boldsymbol{L}$ denote matrices that can be freely chosen.

The main part of the virtual sensor is the state observer with the state vector $\boldsymbol{x}_{\mathrm{V}}(t)$. The complete output $\boldsymbol{y}_{\mathrm{c}}(t)$ of the plant can be approximately determined: $\boldsymbol{y}_{\mathrm{c}}(t) \approx \boldsymbol{C}\boldsymbol{x}_{\mathrm{V}}(t)$. This observation result is improved by using the available sensor values and by observing only the difference between the nominal and the faulty output. In a generalised form, this approach is represented by Eq. (9.44), where the matrix $\boldsymbol{P}$ is a design parameter. For $\boldsymbol{P} = \boldsymbol{O}$ only observed values are used.

Model of the reconfigured plant. The plant together with the virtual sensor is described by Eqs. (9.40)–(9.47):

$$\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_{\mathrm{V}}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{L}\boldsymbol{C}_f & \boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}_f \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_{\mathrm{V}}(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{pmatrix} \boldsymbol{u}_{\mathrm{c}}(t) + \begin{pmatrix} \boldsymbol{E} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{d}(t) \tag{9.48}$$

$$\boldsymbol{y}_{\mathrm{c}}(t) = \begin{pmatrix} \boldsymbol{P}\boldsymbol{C}_f & \boldsymbol{C}_{\mathrm{V}} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_{\mathrm{V}}(t) \end{pmatrix}. \tag{9.49}$$

A state transformation is performed in order to introduce the observation error $\boldsymbol{x}_{\Delta}(t) = \boldsymbol{x}_{\mathrm{V}}(t) - \boldsymbol{x}_f(t)$: Eqs. (9.48), (9.49) are equivalent to

$$\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_{\Delta}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}_f \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_{\Delta}(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{u}_{\mathrm{c}}(t) + \begin{pmatrix} \boldsymbol{E} \\ -\boldsymbol{E} \end{pmatrix} \boldsymbol{d}(t) \tag{9.50}$$

$$\boldsymbol{y}_{\mathrm{c}}(t) = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{C}_{\mathrm{V}} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_{\Delta}(t) \end{pmatrix} \tag{9.51}$$

$$\begin{pmatrix} \boldsymbol{x}_f(0) \\ \boldsymbol{x}_{\Delta}(0) \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{f}0} \\ \boldsymbol{x}_{\mathrm{V}0} - \boldsymbol{x}_{\mathrm{f}0} \end{pmatrix}.$$

Model of the reconfigured loop. For the analysis of the closed-loop behaviour the model of the reconfigured plant is combined with the linear feedback controller (9.37):

$$\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_\Delta(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} - \boldsymbol{BKC} & -\boldsymbol{BKC}_{\rm V} \\ \boldsymbol{O} & \boldsymbol{A} - \boldsymbol{LC}_f \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} + \\ + \begin{pmatrix} \boldsymbol{E} \\ -\boldsymbol{E} \end{pmatrix} \boldsymbol{d}(t) + \begin{pmatrix} \boldsymbol{BV} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{y}_{\rm ref}(t) \qquad (9.52)$$

$$\boldsymbol{y}_f(t) = \begin{pmatrix} \boldsymbol{C}_f & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix}. \qquad (9.53)$$

The trajectory of this system depends on the initial state, the reference input $\boldsymbol{y}_{\rm ref}$ and the disturbance $\boldsymbol{d}$ (Fig. 9.10). As the system is linear, the behaviour can be analysed separately for these three excitations.

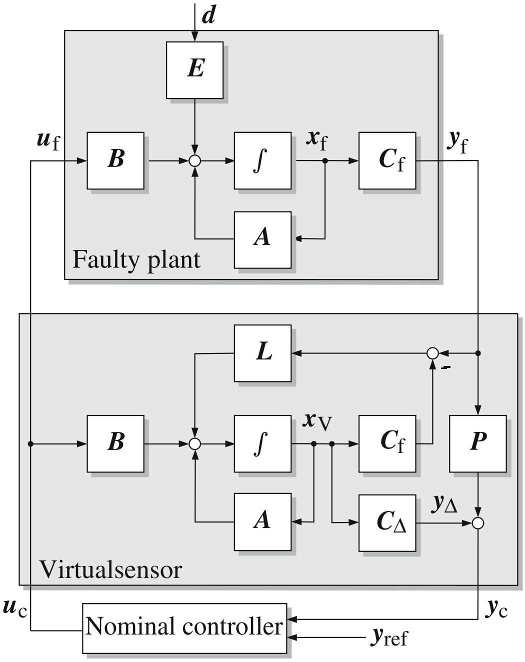
Autonomous behaviour. For $\boldsymbol{y}_{\rm ref}(t) = \boldsymbol{0}$ and $\boldsymbol{d}(t) = \boldsymbol{0}$ the system (9.52), (9.53) simplifies to

$$\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_\Delta(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} - \boldsymbol{BKC} & -\boldsymbol{BKC}_{\rm V} \\ \boldsymbol{O} & \boldsymbol{A} - \boldsymbol{LC}_f \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \qquad (9.54)$$

$$\boldsymbol{y}_f(t) = \begin{pmatrix} \boldsymbol{C}_f & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \qquad (9.55)$$

$$\begin{pmatrix} \boldsymbol{x}_f(0) \\ \boldsymbol{x}_\Delta(0) \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{\rm f0} \\ \boldsymbol{x}_{\rm V0} - \boldsymbol{x}_{\rm f0} \end{pmatrix}.$$

Fig. 9.10 Reconfiguration by using a virtual sensor



The separation principle of state observers applies: The matrix $\boldsymbol{K}$ influences the behaviour of the process state $\boldsymbol{x}_f(t)$ through the submatrix $\boldsymbol{A}-\boldsymbol{BKC}$ (controller design), while $\boldsymbol{L}$ affects the behaviour of the observation error $\boldsymbol{x}_\Delta$ through the submatrix $\boldsymbol{A}-\boldsymbol{LC}_f$ (observer design). There are cross-couplings in one direction only from $\boldsymbol{x}_\Delta(t)$ to $\boldsymbol{x}_f(t)$. The strength of the couplings and the influence of $\boldsymbol{x}_\Delta(t)$ on the output can be reduced by a suitable choice of the matrix $\boldsymbol{P}$.

Theorem 9.1 (Separation principle for the virtual sensor) *The set σ of eigenvalues of the reconfigured closed-loop system (9.54), (9.55) consists of the set of eigenvalues of the nominal closed-loop system (9.38), (9.39) and the set of eigenvalues of the virtual sensor (9.42):*

$$\sigma=\sigma\{\boldsymbol{A}-\boldsymbol{BKC}\}\cup\sigma\{\boldsymbol{A}-\boldsymbol{LC}_f\}.$$

The stability of the closed-loop is guaranteed if the nominal control loop is stable (depending on $\boldsymbol{K}$) and if the observer is stable (depending on $\boldsymbol{L}$). The second condition can be satisfied by an appropriate choice of $\boldsymbol{L}$ because the pair $(\boldsymbol{A},\boldsymbol{C}_f)$ is assumed to be observable. The equilibrium state is zero for both the faulty and the nominal system.

Tracking behaviour. For $\boldsymbol{x}_{\mathrm{f}0}=\boldsymbol{x}_{\mathrm{V}0}=\boldsymbol{0}$ and $\boldsymbol{d}=\boldsymbol{0}$, the system (9.52), (9.53) simplifies to

$$\begin{aligned}\dot{\boldsymbol{x}}_f(t)&=(\boldsymbol{A}-\boldsymbol{BKC})\boldsymbol{x}_f(t)+\boldsymbol{BV}\boldsymbol{y}_{\mathrm{ref}}(t),\quad \boldsymbol{x}_f(0)=\boldsymbol{O}\\ \boldsymbol{y}_f(t)&=\boldsymbol{C}_f\boldsymbol{x}_f(t),\end{aligned}$$

which is identical to the behaviour of the nominal closed-loop system (9.38), (9.39). Hence, the reference tracking behaviour of the reconfigured control loop is identical to that of the nominal control loop.

Disturbance behaviour. For the disturbance behaviour it is assumed that the initial state and the reference input are zero. This leads to the following closed-loop system:

$$\begin{aligned}\begin{pmatrix}\dot{\boldsymbol{x}}_f(t)\\ \dot{\boldsymbol{x}}_\Delta(t)\end{pmatrix}&=\begin{pmatrix}\boldsymbol{A}-\boldsymbol{BKC} & -\boldsymbol{BKC}_{\mathrm{V}}\\ \boldsymbol{O} & \boldsymbol{A}-\boldsymbol{LC}_f\end{pmatrix}\begin{pmatrix}\boldsymbol{x}_f(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix}+\begin{pmatrix}\boldsymbol{E}\\ -\boldsymbol{E}\end{pmatrix}\boldsymbol{d}(t)\\ \boldsymbol{y}_f(t)&=\begin{pmatrix}\boldsymbol{C}_f & \boldsymbol{O}\end{pmatrix}\begin{pmatrix}\boldsymbol{x}_f(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix}\\ \begin{pmatrix}\boldsymbol{x}_f(0)\\ \boldsymbol{x}_\Delta(0)\end{pmatrix}&=\begin{pmatrix}\boldsymbol{O}\\ \boldsymbol{O}\end{pmatrix}.\end{aligned}$$

It is obvious that the output $\boldsymbol{y}_f$ is different from the output $\boldsymbol{y}$ of the nominal control loop. The dynamical disturbance behaviour is much more complex because the number of states of the reconfigured process is $2n$ instead of n for the nominal process. The poles of the disturbance rejection behaviour depend on $\boldsymbol{K}$ and $\boldsymbol{L}$, while the zeros are affected by $\boldsymbol{P}$.

These results are summarised in the following theorem.

Theorem 9.2 (Virtual sensor) *For sensor faults, the virtual sensor (9.42)–(9.44) solves the reconfiguration problem such that the weak reconfiguration goal is reached provided that the faulty process is observable. The strong goal is reached for the reference tracking behaviour.*

The analysis has shown how the virtual sensor works. The direct feedthrough $\boldsymbol{P}$ reconstructs or at least approximates the output $\boldsymbol{y}_{\mathrm{c}}$ of the faultless plant from the remaining output $\boldsymbol{y}_f$. If the condition (9.11) is satisfied and $\boldsymbol{P}$ is chosen according to Eq. (9.13), the virtual sensor shrinks to a static reconfiguration block

$$\begin{aligned}\boldsymbol{y}_{\mathrm{c}}(t) &= \boldsymbol{C}\boldsymbol{C}_f^{\mathrm{T}}\left(\boldsymbol{C}_f^{\mathrm{T}}\boldsymbol{C}_f\right)^{-1}\boldsymbol{y}_f(t)\\ \boldsymbol{u}_f(t) &= \boldsymbol{u}_{\mathrm{c}}(t),\end{aligned}$$

because $\boldsymbol{C}_{\mathrm{V}} = \boldsymbol{O}$ results. This solution to the reconfiguration problem coincides with the solution obtained by the model-matching approach. The strong reconfiguration goal is satisfied.

If the condition (9.11) is not satisfied, the virtual sensor reconstructs the missing sensor information. Its state $\boldsymbol{x}_{\mathrm{V}}(t)$ approximates the plant state $\boldsymbol{x}_f(t)$. The strong reconfiguration goal is satisfied only for the reference tracking behaviour. The disturbance behaviour of the reconfigured closed-loop system is typically slower compared with the nominal behaviour. The smaller the state $\boldsymbol{x}_{\Delta}(t)$ in the model of the disturbance behaviour is, the better approximates the reconfigured loop the nominal behaviour.

9.2.4 Virtual Actuator

This section develops a solution to the reconfiguration problem for actuator failures. The notion of a virtual actuator is introduced as the dual system to the virtual sensor.

The system under consideration is described by

$$\dot{\boldsymbol{x}}_f(t) = \boldsymbol{A}\boldsymbol{x}_f(t) + \boldsymbol{B}_f\boldsymbol{u}_f(t) + \boldsymbol{E}\boldsymbol{d}(t), \quad \boldsymbol{x}_f(0) = \boldsymbol{x}_{\mathrm{f}0} \tag{9.56}$$

$$\boldsymbol{y}_f(t) = \boldsymbol{C}\boldsymbol{x}_f(t), \tag{9.57}$$

where zero columns in the matrix $\boldsymbol{B}_f$ reflect the failing actuators. If the condition (9.16) is satisfied, the static reconfiguration block

$$\begin{aligned}\boldsymbol{u}_f(t) &= \boldsymbol{N}\boldsymbol{u}_{\mathrm{c}}(t)\\ \boldsymbol{y}_{\mathrm{c}}(t) &= \boldsymbol{y}_f(t)\end{aligned}$$

can be used. In the following, the more general case is investigated, where this condition is not satisfied.

To explain the structure of the virtual actuator, the dual system of the reconfigured control loop for sensor faults shown in Fig. 9.11 is constructed. The result is shown in Fig. 9.12.

Definition 9.2 (*Virtual actuator*) Consider the plant (9.56), (9.57) with faulty actuator. The virtual actuator is defined as the system

$$\dot{\boldsymbol{x}}_\Delta(t) = \boldsymbol{A}_\Delta \boldsymbol{x}_\Delta(t) + \boldsymbol{B}_\Delta \boldsymbol{u}_\mathrm{c}(t), \quad \boldsymbol{x}_\Delta(0) = \boldsymbol{x}_{\Delta 0} \tag{9.58}$$

$$\boldsymbol{u}_f(t) = \boldsymbol{C}_\Delta \boldsymbol{x}_\Delta(t) + \boldsymbol{D}_\Delta \boldsymbol{u}_\mathrm{c}(t) \tag{9.59}$$

$$\boldsymbol{y}_\mathrm{c}(t) = \boldsymbol{C}\boldsymbol{x}_\Delta(t) + \boldsymbol{y}_f(t) \tag{9.60}$$

with the state $\boldsymbol{x}_\Delta \in |\mathcal{R}^n$ and the matrices

$$\boldsymbol{A}_\Delta = \boldsymbol{A} - \boldsymbol{B}_f \boldsymbol{M} \tag{9.61}$$

$$\boldsymbol{B}_\Delta = \boldsymbol{B} - \boldsymbol{B}_f \boldsymbol{N} \tag{9.62}$$

$$\boldsymbol{C}_\Delta = \boldsymbol{M} \tag{9.63}$$

$$\boldsymbol{D}_\Delta = \boldsymbol{N}. \tag{9.64}$$

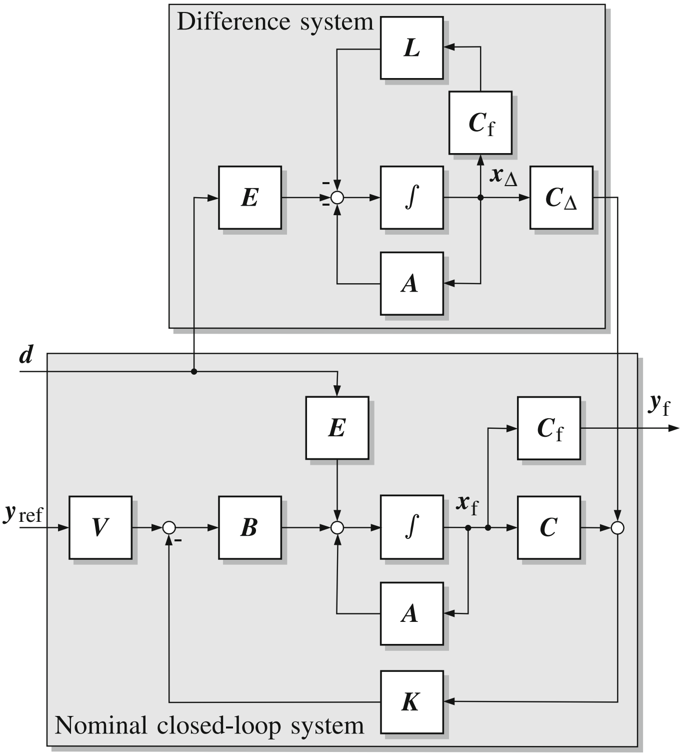


Fig. 9.11 Analysis of the closed-loop system with virtual sensor

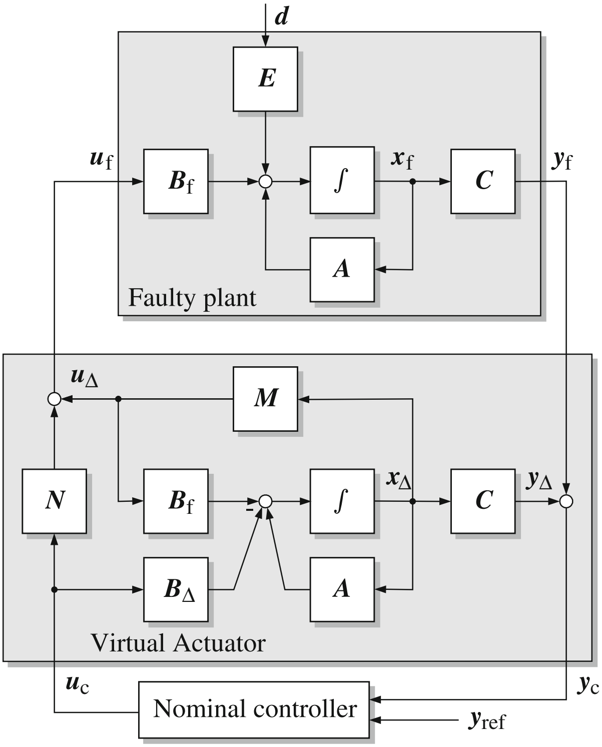


Fig. 9.12 Reconfiguration by means of a virtual actuator

M and ***N*** denote matrices that can be freely chosen.

Analysis of the reconfigured plant. The plant together with the virtual actuator leads to the following model of the reconfigured plant:

$$\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_\Delta(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B}_f \boldsymbol{M} \\ \boldsymbol{O} & \boldsymbol{A} - \boldsymbol{B}_f \boldsymbol{M} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{B}_f \boldsymbol{N} \\ \boldsymbol{B} - \boldsymbol{B}_f \boldsymbol{N} \end{pmatrix} \boldsymbol{u}_{\mathrm{c}}(t) + \begin{pmatrix} \boldsymbol{E} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{d}(t) \tag{9.65}$$

$$\boldsymbol{y}_{\mathrm{c}}(t) = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{C} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix}. \tag{9.66}$$

The introduction of the new state $\hat{\boldsymbol{x}}(t) = \boldsymbol{x}_f(t) + \boldsymbol{x}_\Delta(t)$ leads to the following equivalent model:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\hat{\boldsymbol{x}}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix} &= \begin{pmatrix}\boldsymbol{A} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{A}-\boldsymbol{B}_f\boldsymbol{M}\end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{x}}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix}\\
&\quad + \begin{pmatrix}\boldsymbol{B}\\ \boldsymbol{B}-\boldsymbol{B}_f\boldsymbol{N}\end{pmatrix}\boldsymbol{u}_\mathrm{c}(t) + \begin{pmatrix}\boldsymbol{E}\\ \boldsymbol{O}\end{pmatrix}\boldsymbol{d}(t)\\
\boldsymbol{y}_\mathrm{c}(t) &= \begin{pmatrix}\boldsymbol{C} & \boldsymbol{O}\end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{x}}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix}\\
\begin{pmatrix}\hat{\boldsymbol{x}}(0)\\ \boldsymbol{x}_\Delta(0)\end{pmatrix} &= \begin{pmatrix}\boldsymbol{x}_0+\boldsymbol{x}_{\Delta 0}\\ \boldsymbol{x}_{\Delta 0}\end{pmatrix}.
\end{aligned}$$

Note that the state $\boldsymbol{x}_\Delta$ of the second subsystem is not observable by $\boldsymbol{y}_\mathrm{c}$. Hence, this state does not influence the I/O-behaviour of the reconfigured plant, whose model can be reduced to

$$\begin{aligned}
\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}_\mathrm{c}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 + \boldsymbol{x}_{\Delta 0}\\
\boldsymbol{y}_\mathrm{c}(t) &= \boldsymbol{C}\boldsymbol{x}(t).
\end{aligned}$$

This model is identical to the nominal plant provided that $\boldsymbol{x}_{\Delta 0} = \boldsymbol{0}$ holds.

Theorem 9.3 *The reconfigured plant (9.56)–(9.64) has the same I/O-behaviour as the nominal plant (9.38), (9.39) for arbitrary parameter matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ of the virtual actuator.*

Hence, the virtual actuator yields a reconfigured plant that satisfies the fault-hiding goal for arbitrary matrices $\boldsymbol{M}$ and $\boldsymbol{N}$.

Separation principle for the virtual actuator. The reconfigured closed-loop system consists of the reconfigured plant and the controller (9.37), both of which are considered for vanishing disturbance $\boldsymbol{d}$ and command input $\boldsymbol{y}_\mathrm{ref}$. If the transformed model is used, the reconfigured closed-loop system is described by

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\hat{\boldsymbol{x}}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix} &= \begin{pmatrix}\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}\boldsymbol{C} & \boldsymbol{O}\\ -\boldsymbol{B}_\Delta\boldsymbol{K}\boldsymbol{C} & \boldsymbol{A}-\boldsymbol{B}_f\boldsymbol{M}\end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{x}}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix}\\
\begin{pmatrix}\hat{\boldsymbol{x}}(0)\\ \boldsymbol{x}_\Delta(0)\end{pmatrix} &= \begin{pmatrix}\boldsymbol{x}_0+\boldsymbol{x}_{\Delta 0}\\ \boldsymbol{x}_{\Delta 0}\end{pmatrix}.
\end{aligned}$$

As the system matrix is a block triangular matrix, the following result is obtained:

Theorem 9.4 (Separation principle for the virtual actuator) *The set σ of eigenvalues of the reconfigured closed-loop system (9.37), (9.56)–(9.64) consists of the set of eigenvalues of the nominal closed-loop system (9.37)–(9.39) and the set of eigenvalues of the virtual actuator (9.58):*

$$\sigma = \sigma\{\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}\boldsymbol{C}\} \cup \sigma\{\boldsymbol{A}-\boldsymbol{B}_f\boldsymbol{M}\}.$$

This theorem holds true for arbitrary matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ of the virtual actuator. Clearly, a corollary of this theorem is that the matrix $\boldsymbol{M}$ has to be chosen so that the matrix $\boldsymbol{A} - \boldsymbol{B}_f\boldsymbol{M}$ has eigenvalues with negative real parts in order to ensure the stability of the reconfigured closed-loop system.

Corollary 9.1 *The stability of the reconfigured closed-loop system can be ensured by appropriately choosing the matrix $\boldsymbol{M}$ of the virtual actuator if and only if the pair $(\boldsymbol{A}, \boldsymbol{B}_f)$ is stabilisable.*

This corollary shows that the stabilisation goal can be satisfied by using the generalised virtual actuator as long as the faulty plant is stabilisable.

I/O-behaviour of the reconfigured closed-loop system. The following investigates the I/O-behaviour of the reconfigured closed-loop system and derives guidelines for choosing the parameter matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ of the virtual actuator. If the models of the faulty plant (9.56), (9.57) is combined with the virtual actuator (9.58)–(9.60) and the controller (9.37), the following model is obtained after the state $\hat{\boldsymbol{x}}$ has been introduced as before:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\hat{\boldsymbol{x}}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix} = \begin{pmatrix}\boldsymbol{A}-\boldsymbol{BKC} & \boldsymbol{O}\\ -\boldsymbol{B}_\Delta\boldsymbol{KC} & \boldsymbol{A}-\boldsymbol{B}_f\boldsymbol{M}\end{pmatrix}\begin{pmatrix}\boldsymbol{x}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix} + \begin{pmatrix}\boldsymbol{BV}\\ \boldsymbol{B}_\Delta\boldsymbol{V}\end{pmatrix}\boldsymbol{y}_{\mathrm{ref}}(t) + \begin{pmatrix}\boldsymbol{E}\\ \boldsymbol{O}\end{pmatrix}\boldsymbol{d}(t) \tag{9.67}$$

$$\begin{pmatrix}\hat{\boldsymbol{x}}(0)\\ \boldsymbol{x}_\Delta(0)\end{pmatrix} = \begin{pmatrix}\boldsymbol{x}_0+\boldsymbol{x}_{\Delta 0}\\ \boldsymbol{x}_{\Delta 0}\end{pmatrix}$$

$$\boldsymbol{y}_{\mathrm{c}}(t) = (\boldsymbol{C} \quad \boldsymbol{O})\begin{pmatrix}\hat{\boldsymbol{x}}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix} \tag{9.68}$$

$$\boldsymbol{y}_f(t) = (\boldsymbol{C} \quad -\boldsymbol{C})\begin{pmatrix}\hat{\boldsymbol{x}}(t)\\ \boldsymbol{x}_\Delta(t)\end{pmatrix}. \tag{9.69}$$

The block diagram that illustrates this model is shown in Fig. 9.13. The lower block represents the nominal closed-loop system. The control error $\boldsymbol{e} = \boldsymbol{V}\boldsymbol{y}_{\mathrm{ref}} - \boldsymbol{y}_{\mathrm{c}}$ is fed into the "difference system"

$$\dot{\boldsymbol{x}}_\Delta(t) = (\boldsymbol{A}-\boldsymbol{B}_f\boldsymbol{M})\boldsymbol{x}_\Delta(t) + \boldsymbol{B}_\Delta\boldsymbol{e}(t), \quad \boldsymbol{x}_\Delta(0) = \boldsymbol{x}_{\Delta 0} \tag{9.70}$$

$$\boldsymbol{y}_\Delta(t) = \boldsymbol{C}\boldsymbol{x}_\Delta(t), \tag{9.71}$$

whose name results from its output $\boldsymbol{y}_\Delta$, which is the difference between the output $\boldsymbol{y}_{\mathrm{c}}$ of the nominal closed-loop system and the output $\boldsymbol{y}_f$ of the reconfigured closed-loop system. Hence, $\boldsymbol{y}_\Delta$ shows how the reconfigured closed-loop system differs from the nominal loop.

This model yields two corollaries:

- The I/O-behaviour with respect to the disturbance input $\boldsymbol{d}$ or the command input $\boldsymbol{y}_{\mathrm{ref}}$, respectively, and to the output $\boldsymbol{y}_{\mathrm{c}}$ is identical to the corresponding I/O-behaviour of the nominal closed-loop system.

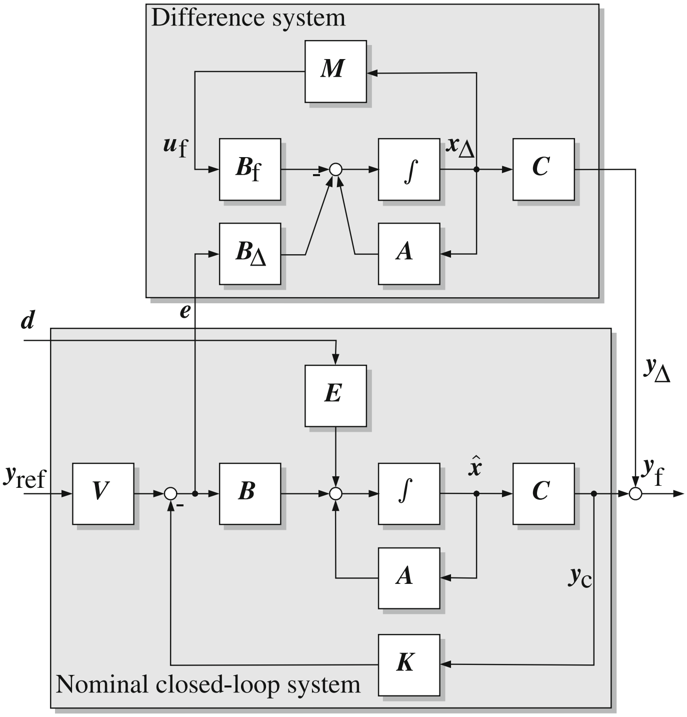


Fig. 9.13 Transformed closed-loop system showing the separation principle

- The I/O-behaviour with respect to the disturbance input $\boldsymbol{d}$ or the command input $\boldsymbol{y}_{\rm ref}$, respectively, and to the output $\boldsymbol{y}_f$ differs from that of the nominal closed-loop system due to the influence of the difference system (9.70), (9.71).

To summarise these results, the virtual actuator presents a successful reconfiguration block in case of actuator failures. It creates a stable control loop with n placeable additional poles. However, it does not restore the original equilibrium unless the equilibrium is zero.

Theorem 9.5 (Virtual actuator) *For actuator failures, the virtual actuator (9.58)–(9.64) is a solution to the reconfiguration problem such that the weak reconfiguration goal is reached provided that the faulty process is controllable.*

The following part of this section concerns the question how to choose the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ of the virtual actuator in order to get a small difference $\boldsymbol{y}_\Delta$ between the behaviour of the nominal and the reconfigured closed-loop system.

Complete reconfiguration. As Fig. 9.13 and Eqs. (9.70), (9.71) show, a complete reconfiguration is possible if the matrix $\boldsymbol{N}$ can be chosen such that the matrix $\boldsymbol{B}_\Delta$ vanishes.

Corollary 9.2 *If the matrix $\boldsymbol{N}$ can be chosen such that*

$$\boldsymbol{B}_{\Delta} = \boldsymbol{B} - \boldsymbol{B}_f \boldsymbol{N} = \boldsymbol{O} \tag{9.72}$$

holds, the I/O-behaviour of the reconfigured closed-loop system is identical to that of the nominal control loop for both the disturbance input $\boldsymbol{d}$ and the command input $\boldsymbol{y}_{\mathrm{ref}}$. Furthermore, if

$$\boldsymbol{x}_{\Delta}(0) = \boldsymbol{0} \tag{9.73}$$

holds, the reconfigured loop has the same free motion as the nominal loop.

The condition (9.72) can be satisfied for an arbitrary controller (9.37) if and only if the relation (9.16) holds. Then the virtual actuator (9.58), (9.60) reduces to the static reconfiguration block

$$\boldsymbol{u}_f(t) = (\boldsymbol{B}_f^{\mathrm{T}} \boldsymbol{B}_f)^{-1} \boldsymbol{B}_f^{\mathrm{T}} \boldsymbol{B} \boldsymbol{u}_{\mathrm{c}}(t) \tag{9.74}$$

$$\boldsymbol{y}_{\mathrm{c}}(t) = \boldsymbol{y}_f(t), \tag{9.75}$$

which is identical to the reconfiguration solution described in Sect. 9.1.4.

If the condition (9.16) is violated, this static reconfiguration block does not solve the reconfiguration problem, the inequality $\boldsymbol{B}_{\Delta} \neq \boldsymbol{O}$ holds and the dynamical part of the virtual actuator becomes active.

Design of the virtual actuator by disturbance decoupling methods. If the transfer function matrix of the difference system (9.70), (9.71) vanishes

$$\boldsymbol{G}(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{B}_f \boldsymbol{M})^{-1}(\boldsymbol{B} - \boldsymbol{B}_f \boldsymbol{N}) = \boldsymbol{O}, \tag{9.76}$$

the reconfiguration is complete as well. Then the difference model (9.70), (9.71), which can be equivalently written as

$$\dot{\boldsymbol{x}}_{\Delta}(t) = \boldsymbol{A}\boldsymbol{x}_{\Delta}(t) + \boldsymbol{B}\boldsymbol{u}_{\mathrm{c}}(t) + \boldsymbol{B}_f \boldsymbol{u}_f(t), \quad \boldsymbol{x}_{\Delta}(0) = \boldsymbol{x}_{\Delta 0} \tag{9.77}$$

$$\boldsymbol{u}_{\Delta}(t) = \boldsymbol{M}\boldsymbol{x}_{\Delta}(t) + \boldsymbol{N}\boldsymbol{u}_{\mathrm{c}}(t) + \boldsymbol{Q}\tilde{\boldsymbol{u}}(t) \tag{9.78}$$

has a vanishing output. To select the matrices $\boldsymbol{N}$ and $\boldsymbol{M}$ such that the condition (9.76) holds is a disturbance decoupling problem for known disturbance $\boldsymbol{u}_{\mathrm{c}}$. It has been shown in [340] that the solution to this problem yields a complete reconfiguration. This solution exist, however, only under restrictive conditions.

Restoration of the static behaviour. The static behaviour is completely reconstructed if the gain of the difference system vanishes:

$$\boldsymbol{G}(0) = -\boldsymbol{C}(\boldsymbol{A} - \boldsymbol{B}_f \boldsymbol{M})^{-1}(\boldsymbol{B} - \boldsymbol{B}_f \boldsymbol{N}) = \boldsymbol{O}. \tag{9.79}$$

Approximate solution. The generalised virtual actuator has the property that the effect of the virtual actuator "disappears" if the matrix $\boldsymbol{B}_{\Delta}$ can be made very small by choosing the matrix $\boldsymbol{N}$ appropriately.

Corollary 9.3 *For* $\|\boldsymbol{B}_{\Delta}\| \to 0$*, the behaviour of the reconfigured closed-loop system approaches that of the nominal loop:*

$$\|\boldsymbol{y}_{\mathrm{c}}(t) - \boldsymbol{y}_f(t)\| \to 0.$$

Hence, if $\|\boldsymbol{B}_{\Delta}\|$ is sufficiently small it is reasonable to use the static reconfiguration block only.

Example 9.4 Reconfiguration of the two-tank system

To illustrate the reconfiguration by means of the virtual actuator, the problem posed in Example 9.3 is considered. The tank system is described by the nonlinear state-space model

$$\begin{aligned}
\dot{h}_1(t) &= \frac{Q_{1\max}}{A_1}(-k_I x_{\mathrm{r}}(t) - k_P(h_1(t) - u_1(t))) \\
&\qquad - \frac{Q_{1\max}}{S}\sqrt{2g(h_1(t) - h_v)}u_2(t) - \frac{Q_{1\max}}{S}\sqrt{2gh_1(t)}u_3(t) \\
\dot{x}_{\mathrm{r}}(t) &= h_1(t) - u_1(t) \\
\dot{h}_2(t) &= \frac{1}{A_2}\left(S\sqrt{2g(h_1(t) - h_v)}u_2(t) + S\sqrt{2gh_1(t)}u_3(t) - S\sqrt{2gh_2(t)}d(t)\right) \\
y_{\mathrm{c}}(t) &= h_2(t)
\end{aligned}$$

that includes the controller of the left tank, which is a PI controller

$$\begin{aligned}
\dot{x}_{\mathrm{r}}(t) &= h_1(t) - u_1(t) \\
\tilde{u}_1(t) &= -k_I x_{\mathrm{r}}(t) - k_P(h_1(t) - u_1(t)).
\end{aligned}$$

This model uses the following parameters:

Symbol	Physical meaning
A_1, A_2	Cross section areas of the two tanks
$Q_{1\max}$	Maximum flow through the pump
h_v	Height of the upper pipe above the tank bottom
S	Constant of the valves
g	gravity constant
k_I, k_P	Controller parameters

After the linearisation of the model around the operation point described by $\bar{h}_1$, $\bar{h}_2$, $\bar{u}_1$, $\bar{u}_2$, $\bar{u}_3$, the linear model (9.35), (9.36) with

$$
\begin{aligned}
\boldsymbol{A} &= \begin{pmatrix} -0.0478 & -0.0004 & 0 \\ 1.0000 & 0 & 0 \\ 0.0058 & 0 & -0.0058 \end{pmatrix} \\
\boldsymbol{B} &= \begin{pmatrix} 0.0406 & -0.0058 & -0.0092 \\ -1.0000 & 0 & 0 \\ 0 & 0.0046 & 0.0073 \end{pmatrix} \\
\boldsymbol{C} &= (0 \quad 0 \quad 1) \\
\boldsymbol{E} &= \begin{pmatrix} 0 \\ 0 \\ -0.0454 \end{pmatrix}
\end{aligned}
$$

is obtained. Is is assumed that the upper valve fails and is, therefore, completely closed and no longer used as actuator of the right level controller. Then, the second column in the matrix $\boldsymbol{B}$ has to be set to zero to obtain the matrix $\boldsymbol{B}_f$:

$$
\boldsymbol{B} = \begin{pmatrix} 0.0406 & 0 & -0.0092 \\ -1.0000 & 0 & 0 \\ 0 & 0 & 0.0073 \end{pmatrix}.
$$

Static reconfiguration. A complete reconfiguration of the controller is possible, because the condition (9.16) is satisfied due to the lower valve, which represents a redundant control input with similar effects on the tank system as the upper valve. In fact, the last column of $\boldsymbol{B}$ is linearly dependent upon the second column:

$$
0.6325 \begin{pmatrix} -0.0092 \\ 0 \\ 0.0073 \end{pmatrix} = \begin{pmatrix} -0.0058 \\ 0 \\ 0.0046 \end{pmatrix}.
$$

Hence, the reconfiguration is possible with a static reconfiguration block (9.74), for which the following parameters are obtained (Fig. 9.14):

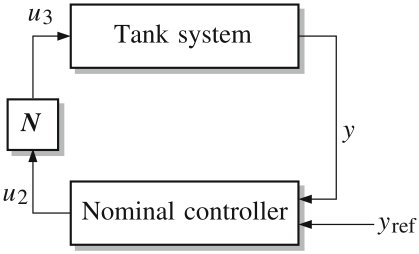


Fig. 9.14 Static reconfiguration of the tank system

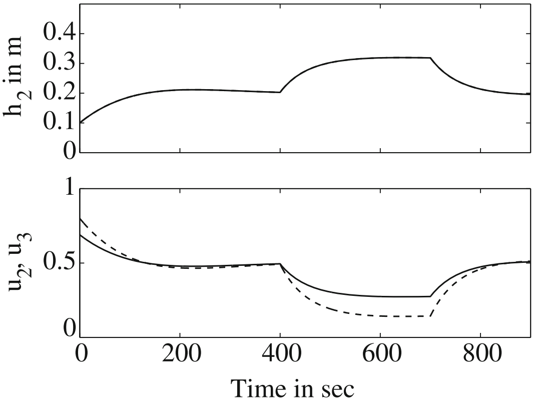


Fig. 9.15 Behaviour of the reconfigured closed-loop system where the reconfigured controller uses the input u_3

$$u_f(t) = \begin{pmatrix} 0 & 0 & -2.7039 \\ 0 & 0 & 0.6325 \\ 0 & 0 & 1 \end{pmatrix} u_{\text{c}}(t).$$

Figure 9.15 shows the reference tracking behaviour of the right tank for changing level set-point. The right tank has the same behaviour with the reconfigured controller as in the nominal case. In the lower subplot the control input u_3 used by the reconfigured controller is compared to the input u_2 of the nominal controller, which is shown by the dashed lines. Clearly, the new input has to be smaller than the nominal one, because the lower valve between the tanks has a higher effectiveness than the upper one, which can be seen by comparing the corresponding columns in the matrix $\boldsymbol{B}$.

Reconfiguration by means of the virtual actuator. If the lower valve is not available for the reconfiguration, the right controller has only the input u_1, which is the command signal of the left controller, as its disposal. With the third columns deleted, the matrices $\boldsymbol{B}$ and $\boldsymbol{B}_f$ do no longer satisfy the condition (9.16). Hence, a dynamical reconfiguration block has to be used. The matrix $\boldsymbol{N}$ of the virtual actuator is chosen according to Eq. (9.18):

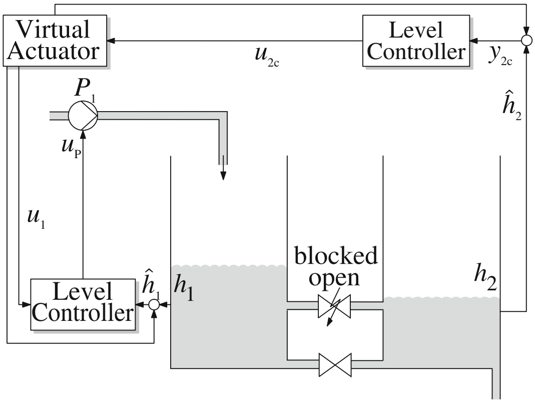


Fig. 9.16 Reconfigured system with virtual actuator

$$\boldsymbol{N} = \begin{pmatrix} 1 & -0.0002 & -0.0004 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The nominal closed-loop system has the eigenvalues -0.0427, $-0.0124 \pm 0.0058i$. Therefore, the matrix $\boldsymbol{M}$ of the virtual actuator is chosen so as to place the eigenvalues of the matrix $\boldsymbol{A} - \boldsymbol{B}_f \boldsymbol{M}$ to the left of these eigenvalues, namely at -0.05, -0.06 and -0.07. As $\boldsymbol{M}$ should use only the first input, its non-zero elements are restricted to the first row:

$$\boldsymbol{M} = \begin{pmatrix} -0.9968 & -0.0048 & -0.0002 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

In summary, the virtual actuator (9.58), (9.59) results

$$\begin{aligned} \dot{\boldsymbol{x}}_\Delta(t) &= \begin{pmatrix} -0.0074 & -0.0002 & 0 \\ 0.0032 & -0.0048 & -0.0002 \\ 0.0058 & 0 & -0.0058 \end{pmatrix} \boldsymbol{x}_\Delta(t) \; + \\ &\quad + \begin{pmatrix} 0 & -0.0058 & -0.0091 \\ 0 & -0.0002 & -0.0004 \\ 0 & 0.0046 & 0.0073 \end{pmatrix} u_{2c}(t) \\ u_1(t) &= (-0.9968 \quad -0.0048 \quad -0.0002)\, \boldsymbol{x}_\Delta(t) \; + \\ &\quad + (1 \quad -0.0002 \quad 0.0004)\, \boldsymbol{y}_{\mathrm{c}}(t), \end{aligned}$$

where $u_{2c}(t)$ is the control input generated by the nominal level controller of the right tank. This signal is used now as an input of the virtual actuator (Fig. 9.16).

Figure 9.17 shows the disturbance behaviour of the tank system after the controller has been extended by a virtual actuator shown above. The response is slower than the nominal

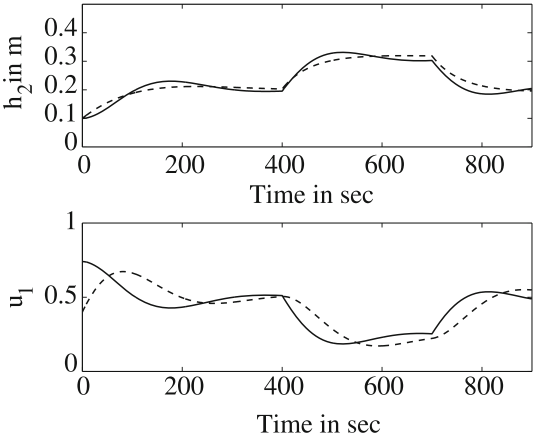


Fig. 9.17 Behaviour of the reconfigured closed-loop system where the reconfigured controller uses the input u_1

response, which is drawn by dashed lines to make a comparison possible. The slower response results from the fact that the controller of the right tank uses now the command input of the controller of the left tank as control input. □

9.2.5 Duality Between Virtual Sensors and Virtual Actuators

The comparison of the reconfiguration blocks developed in the preceding sections clearly shows the duality of the approaches for sensor and actuator failures. The variables correspond to each other in the following way:

Note that the duality involves more than just swapping input and output and transposing the matrices. It requires that the directions of the signals be reversed. Summation points become signal knots and vice versa. The diagrams also require mirroring to preserve the clockwise signal direction of the control loop.

A short mathematical demonstration of the duality is given here. If the system (9.48) is transposed, the input and output matrices are exchanged, and all system matrices are transposed, the following model results:

$$\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\hat{\boldsymbol{x}}}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^{\mathrm{T}} & \boldsymbol{C}_f^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}} \\ \boldsymbol{O} & \boldsymbol{A}^{\mathrm{T}} - \boldsymbol{C}_f^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \hat{\boldsymbol{x}}(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{C}_f^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} \\ \boldsymbol{C}^{\mathrm{T}} - \boldsymbol{C}_f^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} \end{pmatrix} \boldsymbol{u}_f(t)$$

$$\boldsymbol{y}_{\mathrm{c}}(t) = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{B} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \hat{\boldsymbol{x}}(t) \end{pmatrix}.$$

Apart from the different variable names according to Table 9.1, the result is identical to (9.65)–(9.66). The duality holds for most properties, but not for the reference tracking. The reason is that $\boldsymbol{y}_{\mathrm{ref}}$ and $\boldsymbol{y}$ do not have symmetric positions in the system.

9.2.6 Experimental Evaluation: Level and Temperature Control

Reconfiguration of a level and temperature control loop. For a demonstration of the control reconfiguration in case of an actuator failure the part of the chemical process shown in Fig. 9.18 is considered. The control objectives are to maintain a constant liquid level and a constant temperature in the reactor tank B_1 and, thus, producing a constant product outflow. To achieve this, hot and cold liquid can be

Table 9.1 Duality of the system variables

Virtual sensor	A	B	C	K	L	P	$\hat{x}$	u	y
Virtual actuator	A^{T}	C^{T}	B^{T}	K^{T}	M^{T}	N^{T}	$\tilde{x}$	y	u

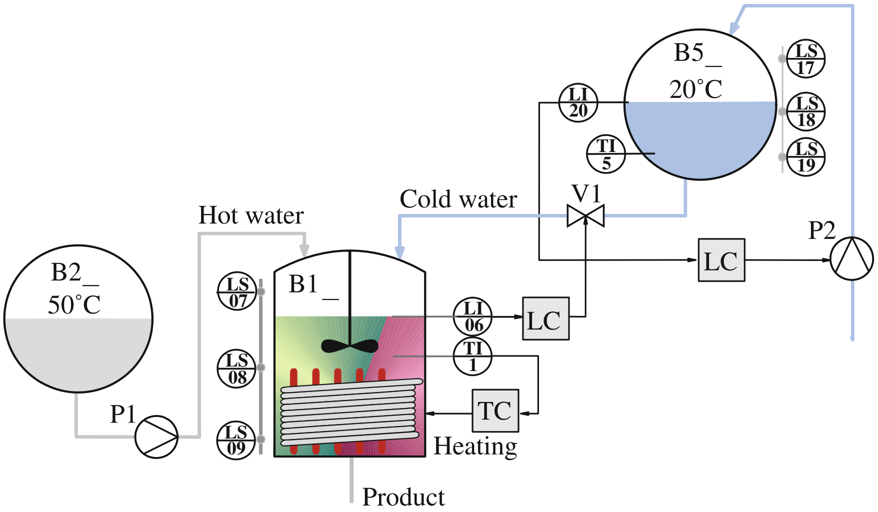


Fig. 9.18 Plant used for control reconfiguration (LC - level control, TC - temperature control)

brought into the reactor from Tanks B_2 and B_5. The main reactor B_1 can be heated and cooled.

In the nominal case, the liquid level is controlled by adjusting the cold liquid inflow from Tank B_5 and the temperature by means of the heating.

Plant model. The plant model contains three states: the reactor content V_{B1}, the reactor temperature ϑ_{B1} and the content of the cold liquid tank V_{B5}. From a mass balance, the following equations are obtained

$$\begin{aligned}
\dot{V}_{\mathrm{B5}}(t) &= k_{\mathrm{P2}} u_{\mathrm{P2}}(t) - q_{51}(t) \\
\dot{V}_{\mathrm{B1}}(t) &= q_{21}(t) + q_{51}(t) - q_{1\mathrm{out}}(t) \\
\dot{\vartheta}_{\mathrm{B1}}(t) &= (\vartheta_{B2}(t) - \vartheta_{\mathrm{B1}}(t)) \frac{q_{21}(t)}{V_{\mathrm{B1}}(t)} + (\vartheta_{\mathrm{B5}}(t) - \vartheta_{\mathrm{B1}}(t)) \frac{q_{51}(t)}{V_{\mathrm{B1}}(t)} \\
&\quad + \frac{u_{\mathrm{heat}}(t) k_{\mathrm{heat}}}{V_{\mathrm{B1}}(t)},
\end{aligned}$$

where for the liquid flows the relations

$$\begin{aligned}
q_{21}(t) &= k_{\mathrm{P1}} u_{\mathrm{P1}}(t) \\
q_{51}(t) &= k_{\mathrm{V1}}\, 124.5^{u_{\mathrm{V1}}(t)} \sqrt{h_{\mathrm{B5}}(t) + 1.07} \\
q_{1\mathrm{out}}(t) &= k_{\mathrm{V2}} \sqrt{\frac{V_{\mathrm{B1}}(t)}{A_{\mathrm{B1}}} + 1.4}
\end{aligned}$$

hold. $h_{\mathrm{B5}}(t)$ is the liquid level in the spherical tank B_5, $u_{\mathrm{heat}}(t)$ the heating power, k_{heat} a heating coefficient, $u_{\mathrm{P1}}(t)$, $u_{\mathrm{P2}}(t)$ and $u_{\mathrm{V1}}(t)$ the control input to the two pumps and to the Valve V_1 and A_{B1} the cross-section area of the Tank B_1. After a linearisation of this nonlinear model around the operating point of $\vartheta_{\mathrm{B1}} = 40\,°\mathrm{C}$, the following linear model is obtained:

$$\begin{pmatrix} \dot{V}_{\mathrm{B5}}(t) \\ \dot{V}_{\mathrm{B1}}(t) \\ \dot{\vartheta}_{\mathrm{B1}}(t) \end{pmatrix} = 10^{-3} \begin{pmatrix} -0.46 & 0 & 0 \\ +0.46 & -0.33 & 0 \\ -0.48 & 0.008 & -1.1 \end{pmatrix} \begin{pmatrix} V_{\mathrm{B5}}(t) \\ V_{\mathrm{B1}}(t) \\ \vartheta_{\mathrm{B1}}(t) \end{pmatrix}$$
$$+ \begin{pmatrix} 0.09 & -0.023 & 0 & 0 \\ 0 & +0.023 & +0.05 & 0 \\ 0 & -0.024 & +0.02 & 0.223 \end{pmatrix} \begin{pmatrix} u_{\mathrm{P2}}(t) \\ u_{\mathrm{V1}}(t) \\ u_{\mathrm{P1}}(t) \\ u_{\mathrm{heat}}(t) \end{pmatrix}$$
$$\boldsymbol{y} = \begin{pmatrix} h_{\mathrm{B5}}(t) \\ h_{\mathrm{B1}}(t) \\ \vartheta_{\mathrm{B1}}(t) \end{pmatrix}.$$

The nominal proportional controllers are defined by:

$$\begin{aligned} u_{\mathrm{V1}}(t) &= -0.5\, V_{\mathrm{B1}}(t) \\ u_{\mathrm{heat}}(t) &= -0.5\, \vartheta_{\mathrm{B1}}(t) \\ u_{\mathrm{P2}}(t) &= -1\, V_{\mathrm{B5}}(t). \end{aligned}$$

They can be represented as

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{y}(t) \quad \text{with} \quad \boldsymbol{K} = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}.$$

Note that these controllers do not use the control input u_{P1}, because the matrix $\boldsymbol{K}$ has a vanishing third row.

Faults. Several severe faults can occur that open the control loops. For example, due to a heating failure, the reactor can no longer be heated, or clogging or blockage of Valve V_1 can bring the level controller out of operation. In the following, the heating failure and a blockage of Valve V_1 in its nominal position will be considered.

Controller reconfiguration after a heating failure. After a heating failure has occurred, the temperature controller

$$u_{\mathrm{heat}}(t) = -0.5\, \vartheta_{\mathrm{B1}}(t)$$

has no influence on the process. The system in the nominal and the faulty case has the matrices

$$\boldsymbol{B} = \begin{pmatrix} 0.09 & -0.023 & 0 & 0 \\ 0 & +0.023 & +0.05 & 0 \\ 0 & -0.024 & +0.02 & 0.223 \end{pmatrix}$$

$$\boldsymbol{B}_f = \begin{pmatrix} 0.09 & -0.023 & 0 & 0 \\ 0 & +0.023 & +0.05 & 0 \\ 0 & -0.024 & +0.02 & 0 \end{pmatrix},$$

which distinguish in the last column. Both matrices have the same rank and can be related to one another by the matrix

$$\boldsymbol{N} = \begin{pmatrix} 1 & 0 & 0 & -1.72 \\ 0 & 1 & 0 & -6.72 \\ 0 & 0 & 1 & 3.09 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

such that the equation

$$\boldsymbol{B}_f \boldsymbol{N} = \boldsymbol{B}$$

holds. Hence, a complete reconfiguration is possible by using the third control input, which is not used in the nominal case. The reconfigured controller

$$\boldsymbol{u}(t) = -\boldsymbol{N}\boldsymbol{K}\boldsymbol{y}(t)$$

has the controller matrix

$$\boldsymbol{N}\boldsymbol{K} = \begin{pmatrix} 0.5 & 0 & -0.86 \\ 0 & 1 & -3.36 \\ 0 & 0 & 1.55 \\ 0 & 0 & 0 \end{pmatrix}.$$

Obviously, the fourth actuator is no longer used. The effect of this actuator is distributed among the three remaining actuators, which can be seen in the last column of the new controller matrix. With the reconfigured controller, the behaviour of the nominal system is completely reproduced.

Controller reconfiguration by means of a virtual actuator. The loss of the actuator V_1 does not affect the operation point, but it breaks the level control loop for the reactor B_1. The use of a reduced virtual actuator allows to keep the nominal controller while changing the control structure as little as possible.

In the terminology of Sect. 9.2.3, the directly influenceable part $\boldsymbol{x}_{F1}$ of the plant state is defined by $V_{\rm B5}$ and $\vartheta_{\rm B1}$, while $\boldsymbol{x}_{F2}$ is the single state variable $V_{\rm B1}$:

$$\boldsymbol{x}_{\rm f1}(t) = \begin{pmatrix} B_{\rm B5}(t) \\ \vartheta_{\rm B5}(t) \end{pmatrix}, \qquad \boldsymbol{x}_{\rm f2}(t) = V_{\rm B1}(t).$$

The $(1, 2)$-parameter matrix $\boldsymbol{M}$ is determined by pole placement. The element of $\boldsymbol{M}$ that is acting on ϑ_{B1} has no influence on the actuator pole and is, therefore, set to 0. The other value is chosen so that the actuator pole lies at -0.004 in order to make the influence of the virtual actuator on the closed-loop dynamics as small as possible. The application of the method explained in Sect. 9.2.4 to this example leads to

$$\begin{aligned}
\dot{\hat{x}}_2(t) &= -0.004\,\hat{x}_2(t) + 0.0229\,u_{\mathrm{V2,R}}(t)\\
\hat{\boldsymbol{u}}(t) &= \begin{pmatrix} 0.015\\ -0.318\\ 0 \end{pmatrix} \hat{x}_2(t) + \begin{pmatrix} -0.107\\ 1.78\\ 0 \end{pmatrix} u_{\mathrm{V2,R}}(t)\\
\hat{\boldsymbol{y}}(t) &= \begin{pmatrix} -8\\ 0\\ 1 \end{pmatrix} \hat{x}_2(t).
\end{aligned}$$

The function of the reduced virtual actuator can be described as follows (Fig. 9.19). The input $u_{\mathrm{V1}}(t)$ is not available to control the inflow into the main reactor, but this inflow also depends on the level in Tank B_5 and, hence, on V_{B5}. In order to reach the same effect as the broken actuator, $V_{\mathrm{B5}}(t)$ is increased or decreased by influencing the Pump P_2 via the input $u_{\mathrm{P2}}(t)$. As $V_{\mathrm{B5}}(t)$ cannot be changed instantaneously, this "replacement action" is slower than the direct action of the nominal control loop on the valve V_1 and leads to a slower reaction of the system under the influence of the reconfigured controller.

In mathematical terms, the virtual actuator brings about an additional pole which yields the slower dynamics. The difference between the nominal and the new behaviour is determined by the virtual actuator and deducted from the measurements of

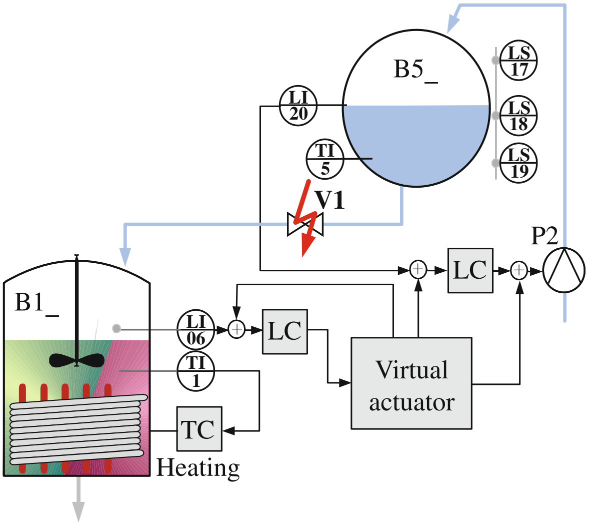


Fig. 9.19 Reconfigured controller including a virtual actuator

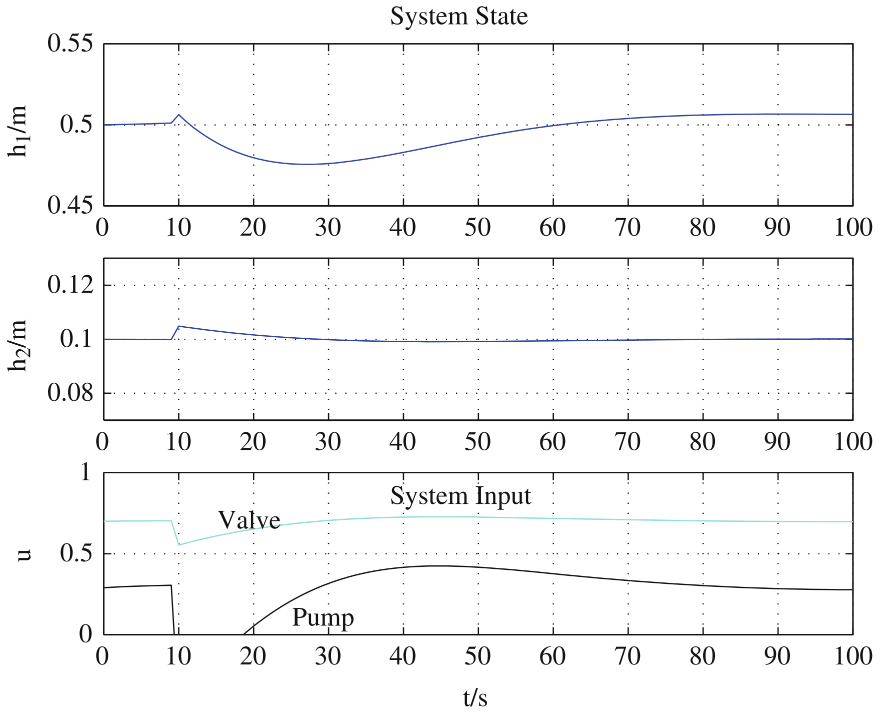


Fig. 9.20 Results of the reconfiguration experiment (Reactor temperature $\vartheta_{B1}(t)$ (*top*), reactor content $V_{B1}(t)$ (*middle*) and reactor content $V_{B5}(t)$

$V_{B1}(t)$ and $V_{B5}(t)$. In this way, the additional pole remains hidden from the level controller and this controller acts like in the nominal case.

The experimental results are shown in Fig. 9.20. The state $V_{B1}(t)$ is disturbed by withdrawing a considerable amount of liquid until time $t = 10$ s. The virtual actuator increases the level $V_{B5}(t)$ in Tank B_5 by increasing the pump input $u_{P1}(t)$. The effect of this manipulation and of the fault is "simulated" by the virtual actuator, subtracted from the sensor's data and, therefore, hidden from the nominal controller. After 180 s the tank level $V_{B5}(t)$ reaches its maximum and after another 800 s the state deviation has been reasonably compensated. A static deviation remains because of some modelling inaccuracies.

The dashed lines show the behaviour of the faultless closed-loop system. The slower reaction of the level controller results in the slower disturbance attenuation shown in the middle part of the figure, where the nominal system reaches the set-point of 19 dm^3 quicker than the reconfigured system. Hence, the operation of the main reactor can be restored with a minor performance degradation.

In the lower part of the figure the different behaviour of Tank B_5 can be seen. The difference is due to the different functions that this tank has in both situations. In the faultless case the level controller of this tank adjusts the liquid content to the set-point, whereas under faulty conditions this variable is used as a means to control the inflow into Tank B_1 and, thus, to control the contents of B_1.

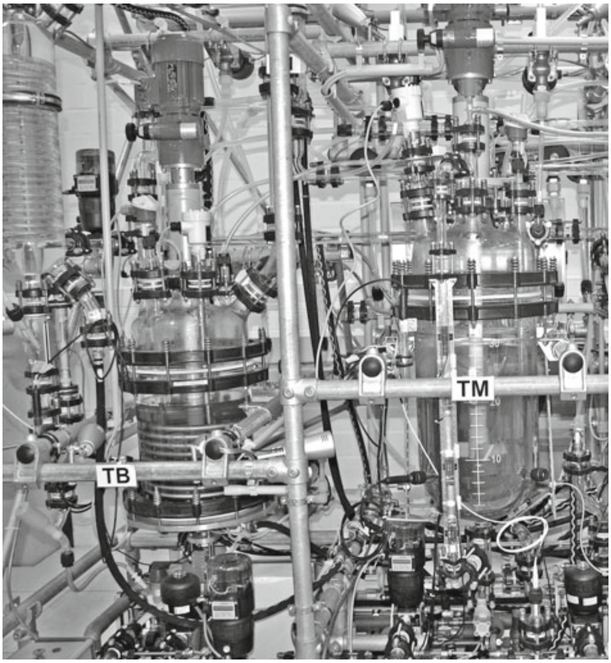


Fig. 9.21 Part of the chemical plant VERA used in the experiment

9.2.7 Experimental Evaluation: Conductivity Control Loop

The second application of the reconfiguration method that uses the virtual actuator is the fault-tolerant control of the conductivity of a liquid. Figure 9.21 shows the experimental set-up and Fig. 9.22 the schematic diagram of the three reactors involved in the control loop considered. The sequence of the two Reactors *TM* and *TB* with the Reactor *TS* is used to produce a liquid with prescribed temperature and conductivity. Several control loops have to be used, which are shown in the schematic diagram with the abbreviations *LC* for level controller, *TC* for temperature controller and *CC* for concentration controller. If actuator failures occur, these loops are brought out of operation. Typical failures concern the valve $V_{\rm CW}$ and the heating P_{el}.

The nominal controller uses the inputs $u_{\rm PS}$, $u_{\rm TS}$ and $u_{\rm TB}$, the three variables to be controlled are the temperature $\vartheta_{\rm TB}$, the liquid level $l_{\rm TS}$ in the Reactor *TS* and the conductivity $\lambda_{\rm TS}$ of the liquid in the Reactor *TS* (Fig. 9.23). The block diagram also shows the redundant inputs $u_{\rm CW}$ and $u_{\rm TM}$, which will be used for the reconfiguration.

Nonlinear model. The following nonlinear model is obtained from balance equations that concern the different components of the plant. To shorten the notation of the equations, the dependency of the signals from the time t is omitted:

- Change of the liquid temperature of Reactor *TS*:

$$\dot{\vartheta}_{\mathrm{TS}} = \frac{1}{A_{\mathrm{TS}}\rho l_{\mathrm{TS}}}\left\{\frac{P_{\mathrm{el,TS}} - \dot{Q}_{PL,TS}}{c_p} + \dot{m}_{\mathrm{TB}}(\vartheta_{\mathrm{TB}} - \vartheta_{\mathrm{TS}}) + \dot{m}_{\mathrm{TM}}(\vartheta_{\mathrm{TM}} - \vartheta_{\mathrm{TS}}) + \dot{m}_{\mathrm{CW}}(\vartheta_{\mathrm{CW}} - \vartheta_{\mathrm{TS}})\right\}$$

- Change of the liquid volume in Reactor *TS*:

$$\dot{l}_{\mathrm{TS}}(t) = \frac{\dot{m}_{\mathrm{TB}}(t) + \dot{m}_{\mathrm{TM}}(t) + \dot{m}_{\mathrm{CW}}(t) - \dot{m}_{\mathrm{TW}}(t)}{A_{\mathrm{TS}}\rho}$$

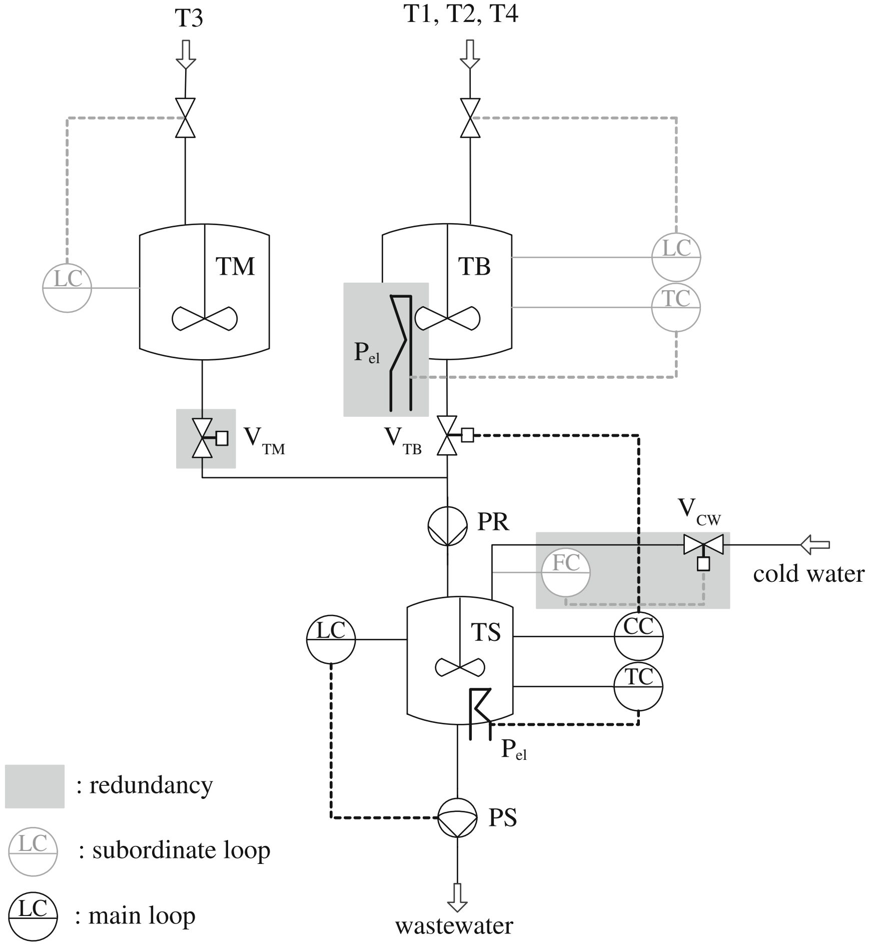


Fig. 9.22 Schematic diagram of the process

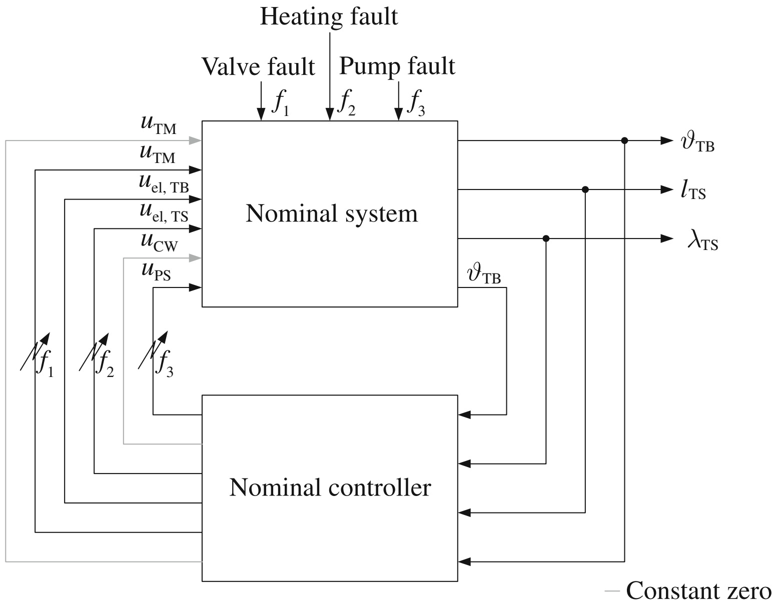


Fig. 9.23 Schematic diagram of the process

- Change of the concentration in Reactor *TS*:

$$\dot{c}_{\mathrm{TS}}(t) = \frac{\dot{m}_{\mathrm{TB}}(t)(c_{\mathrm{TB}} - c_{\mathrm{TS}}(t)) + \dot{m}_{\mathrm{TM}}(t)(c_{\mathrm{TM}} - c_{\mathrm{TS}}(t)) - \dot{m}_{\mathrm{CW}}(t)c_{\mathrm{TS}}(t)}{A_{\mathrm{TS}}\rho l_{\mathrm{TS}}(t)}$$

- Change of the liquid temperature in Reactor *TB*:

$$\dot{\vartheta}_{\mathrm{TB}}(t) = \frac{1}{A_{\mathrm{TB}}\rho l_{\mathrm{TB}}}\left\{\frac{P_{\mathrm{el,TB}}(t) - \dot{Q}_{PL,TB}(t)}{c_p} + \dot{m}_{T124}(t)(\vartheta_{T124} - \vartheta_{\mathrm{TB}}(t))\right\}$$

- Behaviour of the cold water Valve V_{CW}:

$$\dot{x}_{\mathrm{CW}}(t) = = -\frac{1}{T_{\mathrm{CW}}}x_{\mathrm{CW}}(t) + \frac{1}{T_{\mathrm{CW}}}u_{\mathrm{CW}}(t)$$
$$\dot{m}_{\mathrm{CW}}(t) = x_{\mathrm{CW}}(t) \quad \text{with} \;\; T_{\mathrm{CW}} = 3,7\,\mathrm{s}$$

- Actuator dynamics of the heating of the Reactor *TB*:

$$\begin{aligned}\dot{x}_{\mathrm{TB}}(t) &= = -\frac{1}{T_{\mathrm{el,TB}}} x_{\mathrm{TB}}(t) + \frac{1}{T_{\mathrm{el,TB}}} u_{\mathrm{TB}}(t)\\ P_{\mathrm{el,TB}}(t) &= k_{\mathrm{TB}} x_{\mathrm{TB}}(t),\\ &\text{with}\;\; T_{\mathrm{el,TB}} = 27\,\mathrm{s},\;\; k_{\mathrm{TB}} = 18\,\mathrm{kW}\end{aligned}$$

- Actuator dynamics of the heating of the Reactor *TS*:

$$\begin{aligned}\dot{x}_{\mathrm{TS}}(t) &= = -\frac{1}{T_{\mathrm{el,TS}}} x_{\mathrm{TS}}(t) + \frac{1}{T_{\mathrm{el,TS}}} u_{\mathrm{TS}}(t)\\ P_{\mathrm{el,TS}}(t) &= k_{\mathrm{TS}} x_{\mathrm{TS}}(t),\\ &\text{with}\;\; T_{\mathrm{el,TS}} = 65\,\mathrm{s},\;\; k_{\mathrm{TS}} = 4\,\mathrm{kW}\end{aligned}$$

Besides the state variables ϑ_{TB} and l_{TS}, the conductivity is the third variable to be controlled. This signal is obtained by the following relation:

$$\lambda_{\mathrm{TS}}(t) = 0{,}4469 \frac{\mathrm{mS}}{\mathrm{cm}} + 2047{,}7 \frac{\mathrm{mS}}{\mathrm{cm}} c_{\mathrm{TS}}(t).$$

All these equations use the following mass and heat flows:

- Mass flow from Rector *TB* towards Reactor *TS*:

$$\dot{m}_{\mathrm{TB}}(t) = \begin{cases} \left(0{,}019 \frac{\mathrm{kg}}{\mathrm{s}\sqrt{\mathrm{m}}} + 0{,}727 \frac{\mathrm{kg}}{\mathrm{s}\sqrt{\mathrm{m}}}(u_{\mathrm{TB}}(t) - 0{,}13)\right)\sqrt{l_{\mathrm{TB}} + 0{,}3\,\mathrm{m}}, \\ \hfill \text{if } u_{\mathrm{TB}} \geq 0{,}13 \\ 0 \frac{\mathrm{kg}}{\mathrm{s}}, \\ \text{else} \end{cases}$$

- Mass flow from Rector *TM* towards Reactor *TS*:

$$\dot{m}_{\mathrm{TM}}(t) = \begin{cases} \left(0{,}047 \frac{\mathrm{kg}}{\mathrm{s}\sqrt{\mathrm{m}}} + 0.605 \frac{\mathrm{kg}}{\mathrm{s}\sqrt{\mathrm{m}}}(u_{\mathrm{TM}}(t) - 0{,}04)\right)\sqrt{l_{\mathrm{TM}} + 0.3\,\mathrm{m}} \\ \hfill \text{if } u_{\mathrm{TM}} \geq 0.04 \\ 0 \frac{\mathrm{kg}}{\mathrm{s}} \hfill \text{else.} \end{cases}$$

- Mass flow out of the Reactor *TS*:

$$\dot{m}_{\mathrm{PS}}(t) = \dot{m}_{\mathrm{TW}}(t) = 0{,}1679 \frac{\mathrm{kg}}{\mathrm{s}\sqrt{\mathrm{m}}} u_{\mathrm{PS}}(t) \sqrt{l_{\mathrm{TS}}(t) + 0{,}36\,\mathrm{m}}$$

- Heat balance of the Reactor *TS*:

$$\dot{Q}_{\mathrm{PL,TS}}(\vartheta_{\mathrm{TS}}(t)) = \begin{cases} \dot{Q}_{\mathrm{PL,TS,on}}(\vartheta_{\mathrm{TS}}(t)), & \text{if heating is on} \\ \dot{Q}_{\mathrm{PL,TS,off}}(\vartheta_{\mathrm{TS}}(t)), & \text{if heating is off} \end{cases}$$

with

$$\dot{Q}_{\mathrm{PL,TS,on}}(\vartheta_{\mathrm{TS}}(t)) = \begin{cases} 46{,}9403\,\frac{\mathrm{W}}{^\circ\mathrm{C}}(\vartheta_{\mathrm{TS}}(t) - 22{,}5\,^\circ\mathrm{C}), & \text{if } \vartheta_{\mathrm{TS}} \geq 22{,}5\,^\circ\mathrm{C} \\ 0\,\mathrm{W}, & \text{if } \vartheta_{\mathrm{TS}} < 22{,}5\,^\circ\mathrm{C} \end{cases}$$

$$\dot{Q}_{\mathrm{PL,TS,off}}(\vartheta_{\mathrm{TS}}(t)) = \begin{cases} 4{,}8968\,\frac{\mathrm{W}}{^\circ\mathrm{C}}(\vartheta_{\mathrm{TS}}(t) - 22{,}5\,^\circ\mathrm{C}), & \text{if } \vartheta_{\mathrm{TS}} \geq 22{,}5\,^\circ\mathrm{C} \\ 0\,\mathrm{W}, & \text{if } \vartheta_{\mathrm{TS}} < 22{,}5\,^\circ\mathrm{C} \end{cases}$$

- Heat balance of the Reactor *TB*:

$$\dot{Q}_{PL,TB}(\vartheta_{\mathrm{TB}}(t)) = \begin{cases} \dot{Q}_{PL,TB,on}(\vartheta_{\mathrm{TB}}(t)), & \text{if heating is on} \\ \dot{Q}_{PL,TB,off}(\vartheta_{\mathrm{TB}}(t)), & \text{if heating is off} \end{cases}$$

$$\dot{Q}_{PL,TB,on}(\vartheta_{\mathrm{TB}}(t)) = \begin{cases} 135{,}468\,\frac{\mathrm{W}}{^\circ\mathrm{C}}(\vartheta_{\mathrm{TB}}(t) - 22{,}5\,^\circ\mathrm{C}), & \text{if } \vartheta_{\mathrm{TB}} \geq 22{,}5\,^\circ\mathrm{C} \\ 0\,\mathrm{W}, & \text{if } \vartheta_{\mathrm{TB}} < 22{,}5\,^\circ\mathrm{C} \end{cases}$$

$$\dot{Q}_{PL,TB,off}(\vartheta_{\mathrm{TB}}(t)) = \begin{cases} 4{,}8968\,\frac{\mathrm{W}}{^\circ\mathrm{C}}(\vartheta_{\mathrm{TB}}(t) - 22{,}5\,^\circ\mathrm{C}), & \text{if } \vartheta_{\mathrm{TB}} \geq 22{,}5\,^\circ\mathrm{C} \\ 0\,\mathrm{W}, & \text{if } \vartheta_{\mathrm{TB}} < 22{,}5\,^\circ\mathrm{C} \end{cases}$$

The given equations can be lumped together to get a nonlinear state-space model

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{g}(\boldsymbol{x}(k), \boldsymbol{u}(k)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \boldsymbol{y}(k) &= \boldsymbol{h}(\boldsymbol{x}(k), \boldsymbol{u}(k)) \end{aligned}$$

with the state, input and output vectors

$$\boldsymbol{x}(t) = \begin{pmatrix} \vartheta_{\mathrm{TS}}(t) \\ l_{\mathrm{TS}}(t) \\ c_{\mathrm{TS}}(t) \\ \vartheta_{\mathrm{TB}}(t) \\ x_{\mathrm{CW}}(t) \\ x_{\mathrm{TB}}(t) \\ x_{\mathrm{TS}}(t) \end{pmatrix}, \quad \boldsymbol{u}(t) = \begin{pmatrix} u_{\mathrm{TM}}(t) \\ u_{\mathrm{TB}}(t) \\ u_{\mathrm{TB}}(t) \\ u_{\mathrm{TS}}(t) \\ u_{\mathrm{CW}}(t) \\ u_{\mathrm{PS}}(t) \end{pmatrix}, \quad \boldsymbol{y}(t) = \begin{pmatrix} \vartheta_{\mathrm{TS}}(t) \\ l_{\mathrm{TS}}(t) \\ \lambda_{\mathrm{TS}}(t) \\ \vartheta_{\mathrm{TB}}(t) \end{pmatrix}.$$

Linearised model. A linearised state-space model

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{aligned}$$

is obtained from the nonlinear model with the following matrices:

$$A = 10^{-3} \cdot \begin{pmatrix} -3,46 & 0 & 0 & 1,46 & -59,12 & 0 & 39,36 \\ 0 & -0,76 & 0 & 0 & 1,41 & 0 & 0 \\ 0 & 0 & -3,15 & 0 & -0,0034 & 0 & 0 \\ 0 & 0 & 0 & -1,34 & 0 & 157,46 & 0 \\ 0 & 0 & 0 & 0 & -270,27 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -37,03 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -15,38 \end{pmatrix}$$

$$B = 10^{-3} \cdot \begin{pmatrix} -10,62 & 0 & 0 & 0 & 0 & 0 \\ 7,11 & 8,49 & 0 & 0 & 0 & -1,98 \\ 0,0249 & 0,0235 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 270,27 & 0 \\ 0 & 0 & 37,03 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15,38 & 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2047,7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$D = O.$$

The set of eigenvalues of the matrix A

$$\sigma = \{-0,2703; -0,0370; -0,0154; -0,0035; -0,0032; -0,0013; -0,0008\}$$

gives an impression of the dynamical properties of the plant.

Models of the faulty system. The three actuator failures cause a change of the matrix B of the linearised state-space model:

- Failure f_1 of the Valve V_{TB}, which gets the input signal u_{TB}:

$$B_{\mathrm{f}_1} = 10^{-3} \cdot \begin{pmatrix} -10,62 & 0 & 0 & 0 & 0 & 0 \\ 7,11 & 0 & 0 & 0 & 0 & -1,98 \\ 0,0249 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 270,27 & 0 \\ 0 & 0 & 37,03 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15,38 & 0 & 0 \end{pmatrix}$$

- Failure f_2 of the heating of the Reactor *TS*, which acts according to the control input u_{TS}:

$$\boldsymbol{B}_{\mathrm{f}_2} = 10^{-3} \cdot \begin{pmatrix} -10,62 & 0 & 0 & 0 & 0 & 0 \\ 7,11 & 8,49 & 0 & 0 & 0 & -1,98 \\ 0,0249 & 0,0235 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 270,27 & 0 \\ 0 & 0 & 37,03 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Failure f_3 of the Pump PS, which runs according to the control input u_{PS}:

$$\boldsymbol{B}_{\mathrm{f}_3} = 10^{-3} \cdot \begin{pmatrix} -10,62 & 0 & 0 & 0 & 0 & 0 \\ 7,11 & 8,49 & 0 & 0 & 0 & 0 \\ 0,0249 & 0,0235 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 270,27 & 0 \\ 0 & 0 & 37,03 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15,38 & 0 & 0 \end{pmatrix}.$$

These matrices differ from the matrix $\boldsymbol{B}$ for the nominal model with respect to one column each, which is set to zero for the failed actuator.

Control reconfiguration by a virtual actuator. For all three fault cases, the virtual actuator described in Definition 9.2 is used for the control reconfiguration (Fig. 9.24). The scheme is the same in all cases, only the matrix $\boldsymbol{B}_f$, which is a parameter of the virtual actuator, differs. This shows that the control reconfiguration is completely automatic in the sense that a general reconfiguration algorithm can be applied, which adapts the effect of the nominal controller to the failure that has occurred.

The first experiment concerns the reconfiguration with the goal to retain the stability of the closed-loop system. For this task, a virtual actuator with parameter matrix $\boldsymbol{N} = \boldsymbol{O}$ is used.

In case of the failure of the Valve V_{TB}, the virtual actuator has been designed to have the following set of eigenvalues:

$$\begin{aligned} \sigma_{VA} &\overset{!}{=} 25\sigma \\ &= \{-6.7568;\ -0.9259;\ 0.3846;\ -0.0866;\ -0.0790;\ -0.0335;\ -0.0190\} \end{aligned} \tag{9.80}$$

This eigenvalue assignment is accomplished by the feedback matrix

$$\boldsymbol{M} = \begin{pmatrix} -12.31 & -16.05 & 77.63 & 0.15 & 5.11 & 0.40 & -3.71 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 13.39 & -0.01 & 5770 & 90.07 & -23.71 & 178.06 & 15.41 \\ 17.18 & -0.06 & 7332 & 23.26 & -31.85 & 39.14 & 25.31 \\ -1.48 & -0.01 & -642.30 & -2.04 & 2.11 & -3.43 & -1.81 \\ -130.19 & -192.04 & 239.73 & 0.75 & 18.61 & 2.21 & -12.85 \end{pmatrix}.$$

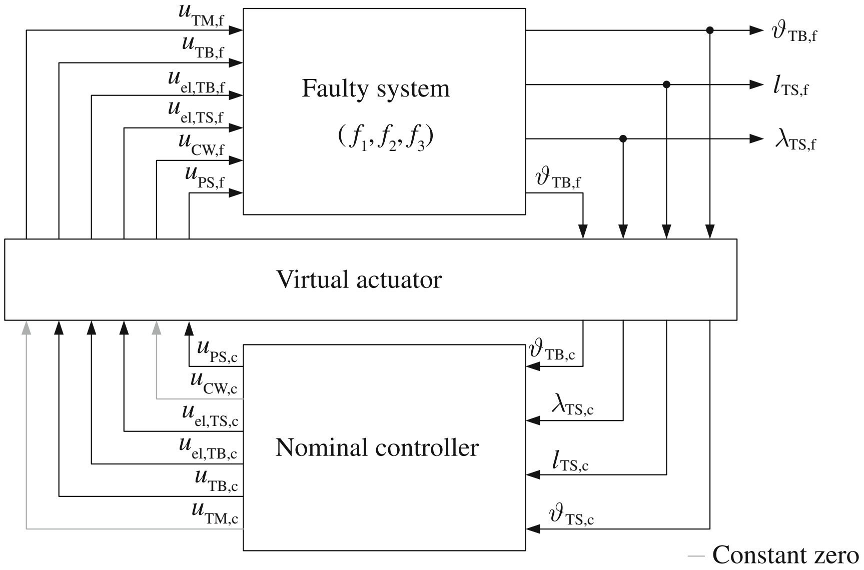


Fig. 9.24 Reconfiguration by means of a virtual actuator

It is possible, because the pair $(\boldsymbol{A}, \boldsymbol{B}_{\mathrm{f1}})$ is completely controllable. The eigenvalues are chosen with respect to the eigenvalues of the plant. They make the virtual actuator much quicker than the plant. The zero row of the matrix $\boldsymbol{M}$ ensures that the failed valve is no longer used for feedback control. Due to the separation property of the virtual actuator, the overall closed-loop system has the eigenvalues of the nominal closed-loop system and the eigenvalues given in Eq. (9.80) for the virtual actuator. Hence, the reconfigured system is stable.

Figure 9.25 approves this result. The two bars placed at time $t = 350$ s mark the time instant at which the valve is blocked and the controller reconfigured. The temperature ϑ_{TS} and the level l_{TS} remain at the set-points, whereas the conductivity cannot follow precisely the set-point change at time $t = 300$ s marked by the dashed line. This is due to the proportional controller used.

Figure 9.26 shows the six control inputs. After the valve V_{TB} is blocked, the signal u_{TB} shown in the top-right corner of the figure does no longer change. The virtual actuator uses the input signals u_{TS}, u_{TB} and u_{PS} which are also used by the nominal controller. In addition to this, the virtual actuator exploits the input u_{CW} to the cold water Valve V_{CW}, whereas the other additional input u_{TM} is not used.

The choice how to distribute the effect of the blocked valve over the remaining actuators is made implicitly by the virtual actuator. No selection procedure, with a possible involvement of a human control designer, is necessary. Therefore, the concept of the virtual actuator can be applied completely automatically.

The second experiment concerns the aim to bring all variables to be controlled back to their set-points. Here the "complete" virtual actuator with the two parameter

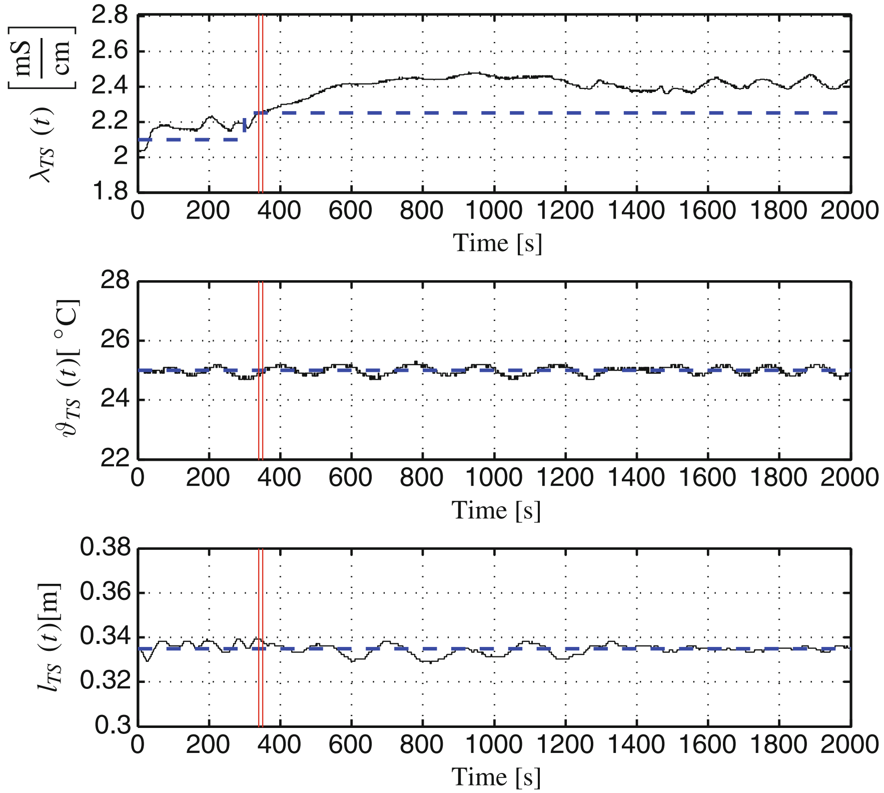


Fig. 9.25 Reconfiguration in case of the valve V_{TB}-failure with $\boldsymbol{N} = \boldsymbol{O}$

matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ is used. Besides the matrix $\boldsymbol{M}$ given above, the direct feedthrough is chosen as

$$\boldsymbol{N} = \left(\boldsymbol{C}(\boldsymbol{A} - \boldsymbol{B}_f\boldsymbol{M})^{-1}\boldsymbol{B}_f\right)^{-1}\left(\boldsymbol{C}(\boldsymbol{A} - \boldsymbol{B}_f\boldsymbol{M})^{-1}\boldsymbol{B}\right)$$

$$= \begin{pmatrix} 1 & 0.291 & -0.016 & 0.053 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.588 & 0.031 & -0.037 & 1 & 0 \\ 0 & -4.250 & -0.012 & -0.004 & 0 & 1 \end{pmatrix},$$

which ensures set-point following, because the reconfigured closed-loop system has the same static reinforcement as the nominal control loop.

The reconfiguration result is depicted in Fig. 9.27. For the same experiment as before now all three control outputs are moved back to their set-points.

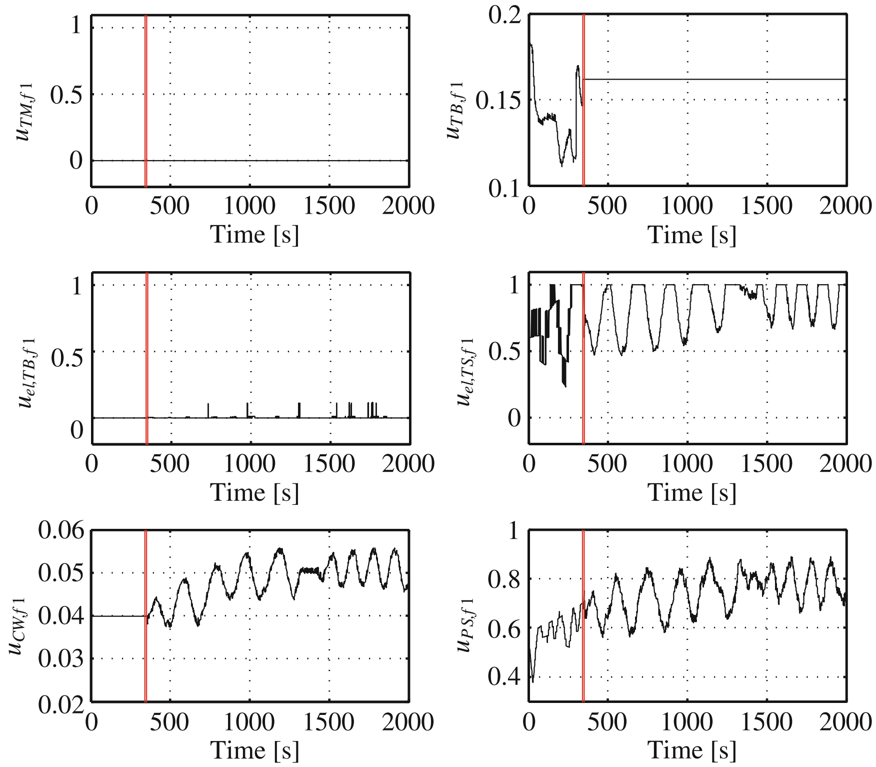


Fig. 9.26 Absolute values of the control inputs after the reconfiguration in case of the valve V_{TB}-failure

As Fig. 9.28 shows, the virtual actuator uses now the additional inputs u_{CW} and u_{TM}. The reconfiguration is completely successful including the restoration of the set-point.

9.3 Fault Recovery by Nominal Trajectory Tracking

Active fault-tolerant control implements control laws that are specific to the diagnosed fault and to the system objective to be achieved. Model-matching and the pseudo-inverse method were first introduced in flight control systems with the objective to minimise the differences between the dynamics of the healthy and the faulty systems, so as to allow pilots to keep faulty systems at hand. However, in some situations, rather than requiring the faulty system dynamics to mimic the nominal system dynamics, it is sensible to require that the faulty system follows (a best approximation of) the nominal system trajectory. Nominal trajectory tracking is of interest, for

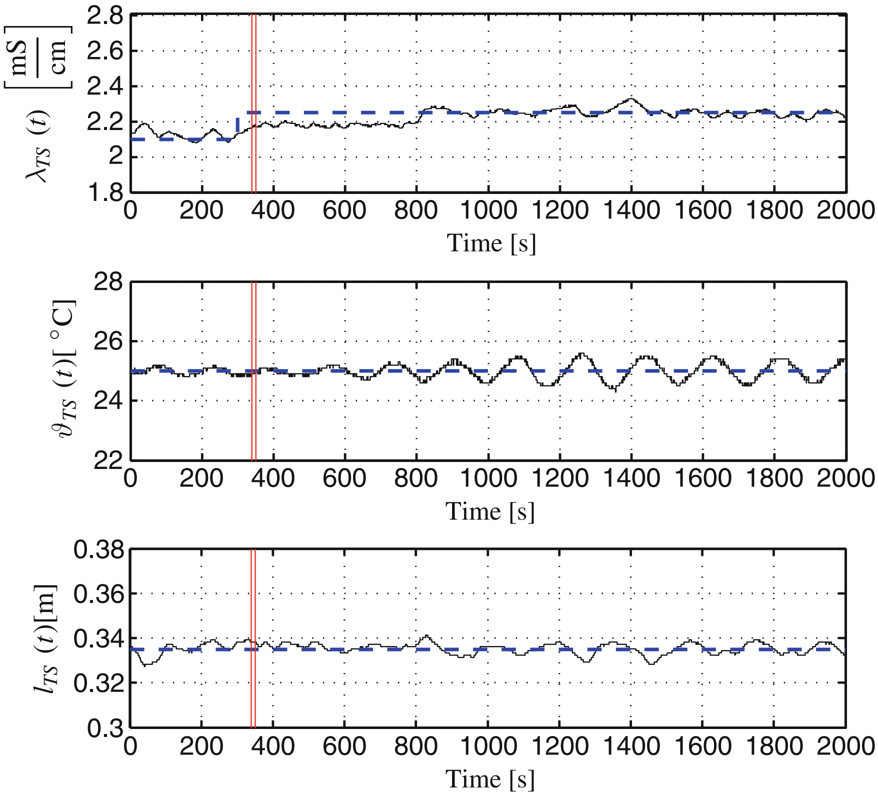


Fig. 9.27 Reconfiguration after valve V_{TB}-failure with feedthrough $\boldsymbol{N} \neq \boldsymbol{O}$

example, when unmanned vehicles are used in space missions where a rescheduling of the whole set of trajectories is impossible. Nominal trajectory tracking is the goal of the approach presented in this section.

9.3.1 Problem Setting

Nominal system. Let

$$\dot{\boldsymbol{x}}_{\text{n}}(t) = \boldsymbol{A}_{\text{n}}\boldsymbol{x}_{\text{n}}(t) + \boldsymbol{B}_{\text{n}}\boldsymbol{u}_{\text{n}}(t) \tag{9.81}$$

be the LTI model of the nominal system, where

$$\boldsymbol{u}_{\text{n}}(t) = \boldsymbol{K}_{\text{n}}\boldsymbol{x}_{\text{n}}(t) \tag{9.82}$$

is the nominal state feedback, that results in the closed-loop behaviour

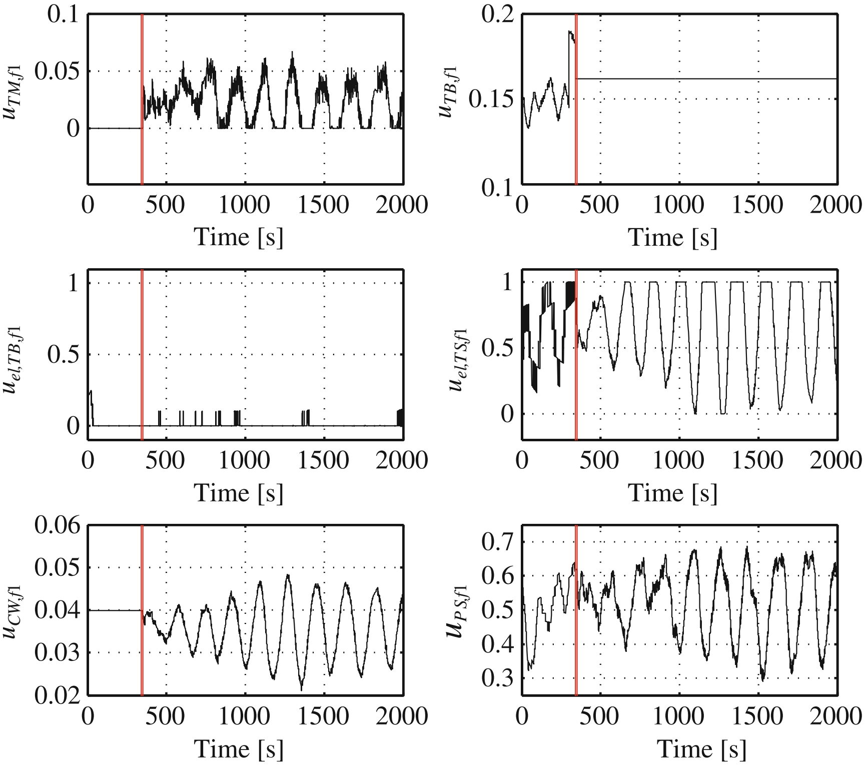


Fig. 9.28 Control input after the reconfiguration for valve V_{TM}-failure

$$\dot{\boldsymbol{x}}_{\mathrm{n}}(t) = (\boldsymbol{A}_{\mathrm{n}} + \boldsymbol{B}_{\mathrm{n}}\boldsymbol{K}_{\mathrm{n}})\boldsymbol{x}_{\mathrm{n}}(t) = \boldsymbol{M}_{\mathrm{n}}\boldsymbol{x}_{\mathrm{n}}(t), \tag{9.83}$$

where $\boldsymbol{M}_{\mathrm{n}}$ is chosen so as to satisfy some nominal requirements including stability. For example, choosing

$$\boldsymbol{M}_{\mathrm{n}} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P},$$

where $\boldsymbol{P}$ is the solution of some Riccati equation associated with the linear quadratic problem setting gives the nominal system an optimal LQ behaviour.

Faulty system. Assume a fault occurs at time t_{f} such that the faulty system can still be described by a model associated with the pair $(\boldsymbol{A}_{\mathrm{f}},\ \boldsymbol{B}_{\mathrm{f}})$ and the control law is changed to $\boldsymbol{u}_{\mathrm{f}}(t) = \boldsymbol{K}_{\mathrm{f}}\boldsymbol{x}_{\mathrm{f}}(t)$. In fault accommodation, the pair $(\boldsymbol{A}_{\mathrm{f}},\ \boldsymbol{B}_{\mathrm{f}})$ is estimated by the fault estimation module, while in system reconfiguration, it is known from the new model that results from switching off the faulty components that have been isolated by the fault isolation module.

Let $[t_{\mathrm{f}},\ t_0[$ be the time interval during which the detection, isolation, fault estimation and accommodation takes place. The post-fault trajectory satisfies:

$$
\begin{aligned}
t \in [t_\mathrm{f},\, t_0[\,:\, & \dot{\boldsymbol{x}}_\mathrm{f}(t) = (\boldsymbol{A}_\mathrm{f} + \boldsymbol{B}_\mathrm{f}\boldsymbol{K}_\mathrm{n})\boldsymbol{x}_\mathrm{f}(t) \\
t \geq t_0 \,:\, & \dot{\boldsymbol{x}}_\mathrm{f}(t) = (\boldsymbol{A}_\mathrm{f} + \boldsymbol{B}_\mathrm{f}\boldsymbol{K}_\mathrm{f})\boldsymbol{x}_\mathrm{f}(t) = \boldsymbol{M}_\mathrm{f}\boldsymbol{x}_\mathrm{f}(t)
\end{aligned}
$$

It follows that for $t \geq t_0$ the trajectory of the accommodated system is given by

$$\boldsymbol{x}_\mathrm{f}(t) = \boldsymbol{\Phi}_\mathrm{f}(t - t_0)\boldsymbol{x}_\mathrm{f}(t)(t_0)$$

while the trajectory of the nominal system would have been

$$\boldsymbol{x}_\mathrm{n}(t) = \boldsymbol{\Phi}_\mathrm{n}(t - t_0)\boldsymbol{x}_\mathrm{n}(t_0)$$

with $\boldsymbol{\Phi}_i(t - t_0) = \mathrm{e}^{\boldsymbol{M}_i(t-t_0)}$, $(i = f,\, n)$. While the model-matching approach is concerned with the difference $\boldsymbol{M}_\mathrm{n} - \boldsymbol{M}_\mathrm{f}$, the nominal trajectory tracking considers the difference $\boldsymbol{\Phi}_\mathrm{n}(t-t_0) - \boldsymbol{\Phi}_\mathrm{f}(t-t_0)$, and requests the trajectory of the accommodated system to mimic as closely as possible the trajectory of the nominal one in an attempt to rub out the effect of the fault.

Introducing two symmetric matrices $\boldsymbol{Q} \geq 0$, and $\boldsymbol{R} > 0$ and measuring the closeness of the trajectories by means of the quadratic cost

$$
\begin{aligned}
J = \frac{1}{2}\int_{t_0}^{\infty} & (\boldsymbol{x}_\mathrm{f}(t) - \boldsymbol{x}_\mathrm{n}(t))^\mathrm{T}\, \boldsymbol{Q}\, (\boldsymbol{x}_\mathrm{f}(t) - \boldsymbol{x}_\mathrm{n}(t)) \\
& + (\boldsymbol{u}_\mathrm{f}(t) - \boldsymbol{u}_\mathrm{n}(t))^\mathrm{T}\, \boldsymbol{R}\, (\boldsymbol{u}_\mathrm{f}(t) - \boldsymbol{u}_\mathrm{n}(t))\, \mathrm{d}t
\end{aligned}
\tag{9.84}
$$

provides a problem setting that allows to achieve a compromise between the discrepancies of the accommodated to nominal trajectory and the accommodated to nominal control signal.

9.3.2 Solution

Optimality condition. From the classical theory of optimal control, one gets the following set of necessary conditions

$$\dot{\boldsymbol{x}}_\mathrm{f}(t) = \boldsymbol{A}_\mathrm{f}\boldsymbol{x}_\mathrm{f}(t) + \boldsymbol{B}_\mathrm{f}\boldsymbol{u}_\mathrm{f}(t) \tag{9.85}$$

$$\dot{\boldsymbol{p}}_\mathrm{f}(t) = \boldsymbol{Q}\,(\boldsymbol{x}_\mathrm{f}(t) - \boldsymbol{x}_\mathrm{n}(t)) - \boldsymbol{A}_\mathrm{f}^\mathrm{T}\boldsymbol{p}_\mathrm{f}(t) \tag{9.86}$$

$$\boldsymbol{O} = \boldsymbol{R}\,(\boldsymbol{u}_\mathrm{f}(t) - \boldsymbol{K}_\mathrm{n}\boldsymbol{x}_\mathrm{n}(t)) - \boldsymbol{B}_\mathrm{f}^\mathrm{T}\boldsymbol{p}_\mathrm{f}(t), \tag{9.87}$$

where $\boldsymbol{p}_\mathrm{f}$ is the adjoint state vector. From (9.87), the accommodated control is

$$\boldsymbol{u}_\mathrm{f}(t) = \boldsymbol{K}_\mathrm{n}\boldsymbol{x}_\mathrm{n}(t) + \boldsymbol{R}^{-1}\boldsymbol{B}_\mathrm{f}^\mathrm{T}\boldsymbol{p}_\mathrm{f}(t). \tag{9.88}$$

Following a classical approach, the adjoint state is assumed to have the form

$$\boldsymbol{p}_{\mathrm{f}}(t) = \boldsymbol{H}\boldsymbol{x}_{\mathrm{f}}(t) + \boldsymbol{G}\boldsymbol{x}_{\mathrm{n}}(t),$$

where $\boldsymbol{H}$ and $\boldsymbol{G}$ are two matrices to be determined. Making use of (9.81), (9.85) and (9.88) one gets

$$\begin{aligned}\dot{\boldsymbol{p}}_{\mathrm{f}}(t) &= \boldsymbol{H}\left(\boldsymbol{A}_{\mathrm{f}} + \boldsymbol{B}_{\mathrm{f}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{H}\right)\boldsymbol{x}_{\mathrm{f}}(t)\\ &+ \left(\boldsymbol{H}\boldsymbol{B}_{\mathrm{f}}\boldsymbol{K}_{\mathrm{n}} + \boldsymbol{B}_{\mathrm{f}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{G} + \boldsymbol{G}\left(\boldsymbol{A}_{\mathrm{n}} + \boldsymbol{B}_{\mathrm{n}}\boldsymbol{K}_{\mathrm{n}}\right)\right)\boldsymbol{x}_{\mathrm{n}}(t)\end{aligned}$$

From (9.86) it follows that

$$\dot{\boldsymbol{p}}_{\mathrm{f}}(t) = \left(\boldsymbol{Q} - \boldsymbol{A}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{H}\right)\boldsymbol{x}_{\mathrm{f}}(t) - \left(\boldsymbol{Q} + \boldsymbol{A}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{G}\right)\boldsymbol{x}_{\mathrm{n}}(t)$$

holds and, therefore,

$$\begin{aligned}\left(\boldsymbol{Q} - \boldsymbol{A}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{H} - \boldsymbol{H}\boldsymbol{A}_{\mathrm{f}} - \boldsymbol{H}\boldsymbol{B}_{\mathrm{f}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{H}\right)\boldsymbol{x}_{\mathrm{f}}(t) &= (+\boldsymbol{H}\boldsymbol{B}_{\mathrm{f}}\boldsymbol{K}_{\mathrm{n}} + \cdots\\ \cdots + \boldsymbol{H}\boldsymbol{B}_{\mathrm{f}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{G} + \boldsymbol{G}\left(\boldsymbol{A}_{\mathrm{n}} + \boldsymbol{B}_{\mathrm{n}}\boldsymbol{K}_{\mathrm{n}}\right) &+ \boldsymbol{A}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{G})\boldsymbol{x}_{\mathrm{n}}(t)\end{aligned}$$

so that $\boldsymbol{H}$ and $\boldsymbol{G}$ must satisfy the relations

$$\boldsymbol{A}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{H} + \boldsymbol{H}\boldsymbol{A}_{\mathrm{f}} + \boldsymbol{H}\boldsymbol{B}_{\mathrm{f}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{H} - \boldsymbol{Q} = \boldsymbol{O} \tag{9.89}$$

$$\boldsymbol{Q} + \boldsymbol{H}\boldsymbol{B}_{\mathrm{f}}\boldsymbol{K}_{\mathrm{n}} + \boldsymbol{G}\left(\boldsymbol{A}_{\mathrm{n}} + \boldsymbol{B}_{\mathrm{n}}\boldsymbol{K}_{\mathrm{n}}\right) + \left(\boldsymbol{H}\boldsymbol{B}_{\mathrm{f}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}} + \boldsymbol{A}_{\mathrm{f}}^{\mathrm{T}}\right)\boldsymbol{G} = \boldsymbol{O}. \tag{9.90}$$

Equation (9.89) is a classical algebraic Riccati equation and (9.90) a Lyapunov equation that is easily solved once $\boldsymbol{H}$ has been found.

Stability. From (9.87) one gets

$$\boldsymbol{u}_{\mathrm{f}}(t) = \boldsymbol{u}_{\mathrm{n}}(t) + \boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\left(\boldsymbol{H}\boldsymbol{x}_{\mathrm{f}}(t) + \boldsymbol{G}\boldsymbol{x}_{\mathrm{n}}(t)\right) \tag{9.91}$$

and, therefore, the accommodated control is obtained by adding the compensating term $\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\left(\boldsymbol{H}\boldsymbol{x}_{\mathrm{f}}(t) + \boldsymbol{G}\boldsymbol{x}_{\mathrm{n}}(t)\right)$ to the nominal control, leading to the accommodated dynamics:

$$\dot{\boldsymbol{x}}_{\mathrm{f}} = \left(\boldsymbol{A}_{\mathrm{f}} + \boldsymbol{B}_{\mathrm{f}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{H}\right)\boldsymbol{x}_{\mathrm{f}}(t) + \boldsymbol{B}_{\mathrm{f}}\left(\boldsymbol{K}_{\mathrm{n}} + \boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{G}\right)\boldsymbol{x}_{\mathrm{n}}(t). \tag{9.92}$$

Let $\boldsymbol{z}^{\mathrm{T}}(t) = \left(\boldsymbol{x}_{\mathrm{n}}(t)^{\mathrm{T}}\ \boldsymbol{x}_{\mathrm{f}}(t)^{\mathrm{T}}\right)$. Then from Eqs. (9.83) and (9.92) one gets $\dot{\boldsymbol{z}}(t) = \boldsymbol{M}\boldsymbol{z}(t)$ with

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{A}_{\rm n} + \boldsymbol{B}_{\rm n}\boldsymbol{K}_{\rm n} & \boldsymbol{O} \\ \boldsymbol{B}_{\rm f}\left(\boldsymbol{K}_{\rm n} + \boldsymbol{R}^{-1}\boldsymbol{B}_{\rm f}^{\rm T}G\right) & A_{\rm f} + \boldsymbol{B}_{\rm f}\boldsymbol{R}^{-1}\boldsymbol{B}_{\rm f}^{\rm T}H \end{pmatrix}.$$

Since $\boldsymbol{K}_{\rm n}$ is chosen such that the nominal closed-loop matrix $\boldsymbol{A}_{\rm n}+\boldsymbol{B}_{\rm n}\boldsymbol{K}_{\rm n}$ is stable, the stability of the accommodated system follows from the stability of $\boldsymbol{A}_{\rm f}+\boldsymbol{B}_{\rm f}\boldsymbol{R}^{-1}\boldsymbol{B}_{\rm f}^{\rm T}H$, which is well known to be achieved by a unique solution $\boldsymbol{H}$ provided that the pair $(\boldsymbol{A}_{\rm f}, \boldsymbol{B}_{\rm f})$ is still controllable and that the pair $(\boldsymbol{C}, \boldsymbol{A}_{\rm f})$ is observable with $\boldsymbol{Q} = \boldsymbol{C}^{\rm T}\boldsymbol{C}$.

Admissibility. Let $(\boldsymbol{A}_{\rm f}, \boldsymbol{B}_{\rm f})$ be a fault such that $(\boldsymbol{A}_{\rm f}, \boldsymbol{B}_{\rm f})$ is controllable and $(\boldsymbol{C}, \boldsymbol{A}_{\rm f})$ is observable, then there exists a unique pair $(\boldsymbol{H}, \boldsymbol{G})$ such that the accommodated control $\boldsymbol{u}_{\rm f}(t) = \boldsymbol{u}_{\rm n}(t) + \boldsymbol{R}^{-1}\boldsymbol{B}_{\rm f}^{\rm T}\,(\boldsymbol{H}\boldsymbol{x}_{\rm f}(t) + \boldsymbol{G}\boldsymbol{x}_{\rm n}(t))$ stabilises the faulty system and is optimal with respect to the cost (9.84). However, not any such fault is recoverable, because although minimal, the cost (9.84) might be too high for the accommodated behaviour to be accepted as close enough to the nominal one.

Let $\epsilon_s(t) = \boldsymbol{x}_{\rm f}(t) - \boldsymbol{x}_{\rm n}(t)$ and $\epsilon_u(t) = \boldsymbol{u}_{\rm f}(t) - \boldsymbol{u}_{\rm n}(t)$ be the differences between the faulty and the nominal system behaviour. Using Eq. (9.91) one gets

$$\epsilon_s^{\rm T}(t)\boldsymbol{Q}\epsilon_s(t) + \epsilon_u^{\rm T}(t)\boldsymbol{R}\epsilon_u(t) = z^{\rm T}(t)\boldsymbol{S}z(t)$$

with

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{Q} + \boldsymbol{G}^{\rm T}\boldsymbol{B}_{\rm f}\boldsymbol{R}^{-1}\boldsymbol{B}_{\rm f}^{\rm T}\boldsymbol{G} & -\boldsymbol{Q} + \boldsymbol{G}^{\rm T}\boldsymbol{B}_{\rm f}\boldsymbol{R}^{-1}\boldsymbol{B}_{\rm f}^{\rm T}\boldsymbol{H} \\ -\boldsymbol{Q} + \boldsymbol{H}^{\rm T}\boldsymbol{B}_{\rm f}\boldsymbol{R}^{-1}\boldsymbol{B}_{\rm f}^{\rm T}\boldsymbol{G} & \boldsymbol{Q} + \boldsymbol{H}^{\rm T}\boldsymbol{B}_{\rm f}\boldsymbol{R}^{-1}\boldsymbol{B}_{\rm f}^{\rm T}\boldsymbol{H} \end{pmatrix}.$$

The cost can now be easily computed. Since $\boldsymbol{M}$ is stable, there is a symmetric negative definite matrix $\boldsymbol{P}$ such that

$$\boldsymbol{M}^{\rm T}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{M} = \boldsymbol{S}$$

It follows that

$$\frac{\rm d}{{\rm d}t}z^{\rm T}(t)\boldsymbol{P}z(t) = z^{\rm T}(t)\boldsymbol{S}z(t)$$

and

$$J = \frac{1}{2}\int_{t_0}^{\infty}\frac{\rm d}{{\rm d}t}z^{\rm T}(t)\boldsymbol{P}z(t){\rm d}t = -\frac{1}{2}z^{\rm T}(t_0)\boldsymbol{P}z(t_0).$$

As already seen, different admissibility conditions can be stated. For example, one can define a constant admissibility limit η, resulting in recoverable faults that satisfy the inequality

$$-\frac{1}{2}z^{\rm T}(t_0)\boldsymbol{P}z(t_0) \leq \eta \tag{9.93}$$

or a quadratic admissibility limit$-\frac{1}{2}\boldsymbol{z}^{\mathrm{T}}(t_0)\boldsymbol{P}_{\min}z\,(t_0)$ resulting in the recoverable faults if

$$\boldsymbol{P}-\boldsymbol{P}_{\min}\geq 0 \tag{9.94}$$

holds.

Remark 9.2 The state discrepancy in the time window $[t_{\mathrm{f}}, t_0[$ is not taken into account in the cost (9.84) since it depends on the fault only and the control has not yet been accommodated.

Remark 9.3 (*Recoverability versus accommodation delay*) The larger the fault the larger is the initial state difference $\epsilon_s(t_0)=\boldsymbol{x}_{\mathrm{f}}(t_0)-\boldsymbol{x}_{\mathrm{n}}(t_0)$. It follows that, depending on the admissibility condition that is defined, the recoverability of a fault depends on the fault itself (for example faults that result in the loss of controllability of unstable systems are not recoverable), but also partly on the delay introduced by the diagnosis and accommodation processes, as Eq. (9.93) suggests.

Remark 9.4 The model-matching approach can in no case provide any optimal solution to the trajectory tracking problem, since it results in the trajectories $\dot{\boldsymbol{x}}_{\mathrm{f}}(t)=\boldsymbol{M}_{\mathrm{f}}\boldsymbol{x}_{\mathrm{f}}(t)$ obtained by synthesising a matrix $\boldsymbol{M}_{\mathrm{f}}$ closest to $\boldsymbol{M}_{\mathrm{n}}$. Whatever way $\boldsymbol{M}_{\mathrm{f}}$ is computed, the input $\boldsymbol{x}_{\mathrm{n}}(t)$ is never taken into account, as (9.91) shows it should be. □

Example 9.5 Nominal trajectory tracking
Consider, a set of second order systems $(\boldsymbol{A},\ \boldsymbol{B}\,(\theta))$, where $\theta\in[0,1]$ is a parameter, such that $\theta=0$ characterises the nominal system $\boldsymbol{A}_{\mathrm{n}}=\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}$, $\boldsymbol{B}_{\mathrm{n}}=\begin{pmatrix}1\\ 5\end{pmatrix}$ and $\theta>0$ is associated with the faulty system $\boldsymbol{A}_{\mathrm{f}}=\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}$, $\boldsymbol{B}_{\mathrm{f}}=\begin{pmatrix}1-2\theta\\ 5-4\theta\end{pmatrix}$. Note that for $\theta=0.5$ the faulty system is not controllable.

Under the state feedback control $u=k_1x_1+k_2x_2$ the closed-loop matrix is

$$\boldsymbol{M}\,(k_1,k_2,\theta)=\begin{pmatrix}k_1\,(1-2\theta)-1 & k_2\,(1-2\theta)\\ k_1\,(5-4\theta) & k_2\,(5-4\theta)-1\end{pmatrix}.$$

Assuming that the control objective is to obtain the behaviour associated with the reference model $\boldsymbol{M}_{\mathrm{n}}=\begin{pmatrix}-2 & 0\\ -5 & -1\end{pmatrix}$, the pseudo-inverse method results in the feedback gains:

$$\begin{aligned}k_1\,(\theta)&=\frac{22\theta-26}{(1-2\theta)^2+(5-4\theta)^2}\\ k_2\,(\theta)&=0.\end{aligned}$$

It is easily seen that the Frobenius norm $\|\boldsymbol{M}\,(k_1,k_2,\theta)-\boldsymbol{M}_{\mathrm{n}}\|_{\mathrm{F}}$ can be zeroed to obtain an exact model-matching result only in the nominal case $\theta=0$ and this minimum is associated with the nominal control $\boldsymbol{u}_{\mathrm{n}}(t)=(-1\ \ 0)\boldsymbol{x}_{\mathrm{n}}(t)$. For $\theta\neq 0$, the pseudo-inverse method results in the closed-loop matrix:

$$\boldsymbol{M}_{\mathrm{f}}^{\mathrm{PIM}}(\theta) = \begin{pmatrix} \frac{-64\theta^2+118\theta-52}{20\theta^2-44\theta+26} & 0 \\ \frac{-88\theta^2+214\theta-130}{20\theta^2-44\theta+26} & -1 \end{pmatrix}$$

whose eigenvalues are

$$\lambda_1(\theta) = -1 \ \lambda_2(\theta) = \frac{-64\theta^2+118\theta-52}{20\theta^2-44\theta+26}.$$

It can be checked that the pseudo-inverse method provides an unstable solution for all faults wiht $\theta > 0.728$.

Let us now investigate the nominal trajectory tracking approach, using $\boldsymbol{Q} = \boldsymbol{I}_2$ and $\boldsymbol{R} = 1$. The post-fault optimal control is obtained as $\boldsymbol{u}_{\mathrm{f}}(t) = \boldsymbol{H}\boldsymbol{x}_{\mathrm{f}}(t) + \boldsymbol{G}\boldsymbol{x}_{\mathrm{n}}(t)$ provided that $\theta \neq 0.5$. With

$$\boldsymbol{W}(\theta) = \begin{pmatrix} (1-2\theta)^2 & (1-2\theta)(5-4\theta) \\ (5-4\theta)(1-2\theta) & (5-4\theta)^2 \end{pmatrix}$$

the matrix $\boldsymbol{H}$ is given by

$$\boldsymbol{H}\boldsymbol{W}(\theta)\boldsymbol{H} - 2\boldsymbol{H} - \boldsymbol{I}_2 = \boldsymbol{O}$$

while $\boldsymbol{G}$ is the solution of

$$\boldsymbol{I}_2 + \boldsymbol{H}\begin{pmatrix} -(1-2\theta) & 0 \\ -(5-4\theta) & 0 \end{pmatrix} + \boldsymbol{G}\begin{pmatrix} -2 & 0 \\ -5 & -1 \end{pmatrix} + (\boldsymbol{W}(\theta) - \boldsymbol{I}_2)\boldsymbol{G} = \boldsymbol{O}.$$

Stable case. Let us first illustrate the case where the pseudo-inverse method (PIM) provides a stable closed loop by assuming the fault $\theta = 0.6$. The PIM solution is

$$\boldsymbol{u}_{\mathrm{f}}(t)^{\mathrm{PIM}} = -1.88235\boldsymbol{x}_1^{\mathrm{PIM}}$$

which gives the closed-loop matrix

$$\boldsymbol{M}_{\mathrm{f}}^{\mathrm{PIM}} = \begin{pmatrix} -0.6235 & 0 \\ -4.8941 & -1 \end{pmatrix}.$$

The nominal trajectory tracking optimal control $\boldsymbol{u}_{\mathrm{f}}(t)$ is defined by the pair

$$\boldsymbol{H} = \begin{pmatrix} -0.4986 & -0.0181 \\ -0.0181 & -0.2650 \end{pmatrix}, \quad \boldsymbol{G} = \begin{pmatrix} 0.2789 & 0.0181 \\ -0.1258 & -0.2650 \end{pmatrix}$$

that satisfies the equations

$$\boldsymbol{H}\begin{pmatrix} 0.04 & -0.52 \\ -0.52 & 6.76 \end{pmatrix}\boldsymbol{H} - 2\boldsymbol{H} - \boldsymbol{I}_2 = \boldsymbol{O}$$

$$\boldsymbol{H}\begin{pmatrix} 0.2 & 0 \\ -2.6 & 0 \end{pmatrix} + \boldsymbol{I}_2 + \boldsymbol{G}\begin{pmatrix} -2 & 0 \\ -5 & -1 \end{pmatrix} + \begin{pmatrix} -0.96 & -0.52 \\ -0.52 & 5.76 \end{pmatrix}\boldsymbol{G} = \boldsymbol{O}.$$

Figure 9.29 shows the nominal $\boldsymbol{x}_{\mathrm{n}}(t)$, PIM $\boldsymbol{x}_{\mathrm{f}}^{\mathrm{PIM}}(t)$ and NTT (nominal trajectory tracking) $\boldsymbol{x}_{\mathrm{f}}(t)$ state trajectories, assuming that during the first 2 s, the faulty system is still controlled by the nominal control. As a result, the $\boldsymbol{x}_{\mathrm{f}}^{\mathrm{PIM}}(t)$ and $\boldsymbol{x}_{\mathrm{f}}(t)$ trajectories are identical for $t \in [10, 12[$, and $\boldsymbol{x}_{\mathrm{f}}(t)$ shows a behaviour closer to $\boldsymbol{x}_{\mathrm{n}}(t)$ only after $t = 12$. Figure 9.30 shows the significant improvements in the quadratic costs associated with the discrepancies $\left(\boldsymbol{x}_{\mathrm{n}}(t) - \boldsymbol{x}_{\mathrm{f}}^{\mathrm{PIM}}(t),\ \boldsymbol{u}_{\mathrm{n}}(t) - \boldsymbol{u}_{\mathrm{f}}(t)^{\mathrm{PIM}}\right)$ and $(\boldsymbol{x}_{\mathrm{n}}(t) - \boldsymbol{x}_{\mathrm{f}}(t),\ \boldsymbol{u}_{\mathrm{n}}(t) - \boldsymbol{u}_{\mathrm{f}}(t))$ again for a 2 s delay.

Unstable case. Let us now consider the case $\theta = 1$ in which the PIM control $\boldsymbol{u}_{\mathrm{f}}(t)^{\mathrm{PIM}} = -2\boldsymbol{x}_1$ gives the closed-loop matrix $\boldsymbol{M}_{\mathrm{f}}^{\mathrm{PIM}} = \begin{pmatrix} 1 & 0 \\ -2 & -1 \end{pmatrix}$, which is unstable. The modified PIM approach gives the control law $\boldsymbol{u}_{\mathrm{f}}(t)^{\mathrm{MPIM}} = -0.8\boldsymbol{x}_1$ which results in the stable matrix

$$\boldsymbol{M}_{\mathrm{f}}^{\mathrm{MPIM}} = \begin{pmatrix} -0.2 & 0 \\ -0.8 & -1 \end{pmatrix}.$$

The nominal control $\boldsymbol{u}_{\mathrm{f}}(t)$ is defined by the pair

$$\boldsymbol{H} = \begin{pmatrix} -0.433 & -0.067 \\ -0.067 & -0.433 \end{pmatrix}, \quad \boldsymbol{G} = \begin{pmatrix} 2.134 & 0.5 \\ 1.8 & -0.5 \end{pmatrix}$$

that satisfies

$$\boldsymbol{H} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \boldsymbol{H} - 2\boldsymbol{H} - \boldsymbol{I}_2 = \boldsymbol{O}$$

$$\boldsymbol{H} \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix} + \boldsymbol{G} \begin{pmatrix} -2 & 0 \\ -5 & -1 \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \boldsymbol{G} + \boldsymbol{I}_2 = \boldsymbol{O}.$$

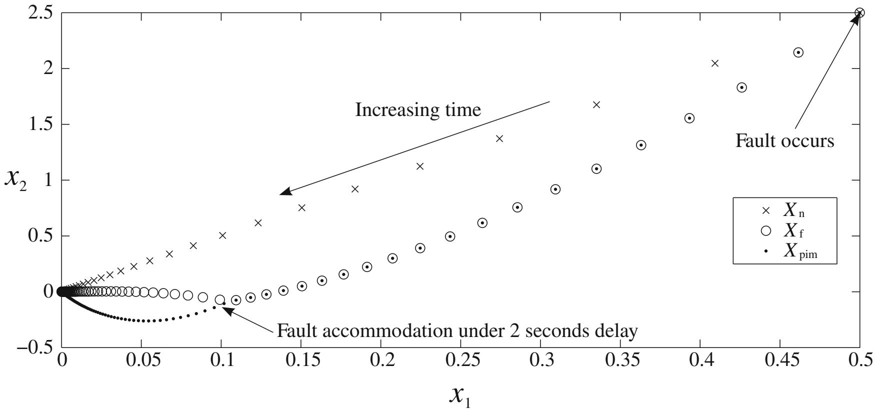


Fig. 9.29 Nominal, PIM and NTT state trajectories

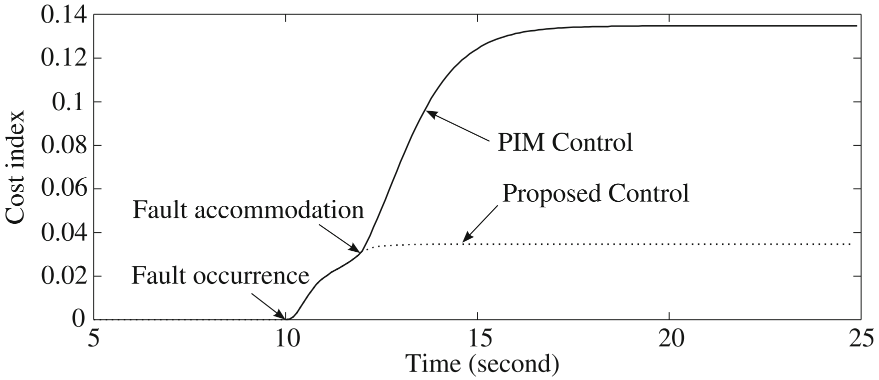


Fig. 9.30 PIM versus NTT costs

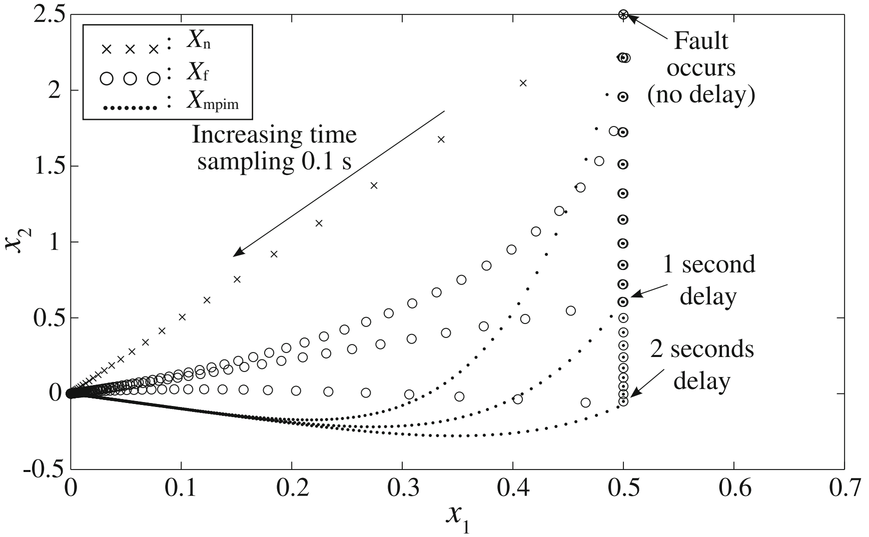


Fig. 9.31 Nominal, MPIM and NTT state trajectories

Figure 9.31 shows the state $\boldsymbol{x}_{\mathrm{n}}(t)$, modified PIM $\boldsymbol{x}_{\mathrm{f}}(t)^{\mathrm{MPIM}}$ and nominal trajectory tracking $\boldsymbol{x}_{\mathrm{f}}(t)$ trajectories, for three different fault detection, isolation, diagnosis and accommodation delays, while Fig. 9.32 shows the quadratic costs associated with the discrepancies $\left(\boldsymbol{x}_{\mathrm{n}}(t)-\boldsymbol{x}_{\mathrm{f}}(t)^{\mathrm{MPIM}}, \boldsymbol{u}_{\mathrm{n}}(t)-\boldsymbol{u}_{\mathrm{f}}(t)^{\mathrm{MPIM}}\right)$ and $(\boldsymbol{x}_{\mathrm{n}}(t)-\boldsymbol{x}_{\mathrm{f}}(t), \boldsymbol{u}_{\mathrm{n}}(t)- \boldsymbol{u}_{\mathrm{f}}(t))$ for the 2 s delay case.

In order to illustrate Remark 9.3, Fig. 9.33 shows how the trajectory tracking cost increases with the diagnosis and accommodation delay. It follows that for small delays the fault may be recoverable, while it becomes non-recoverable for larger ones, because the cost becomes inadmissible. □

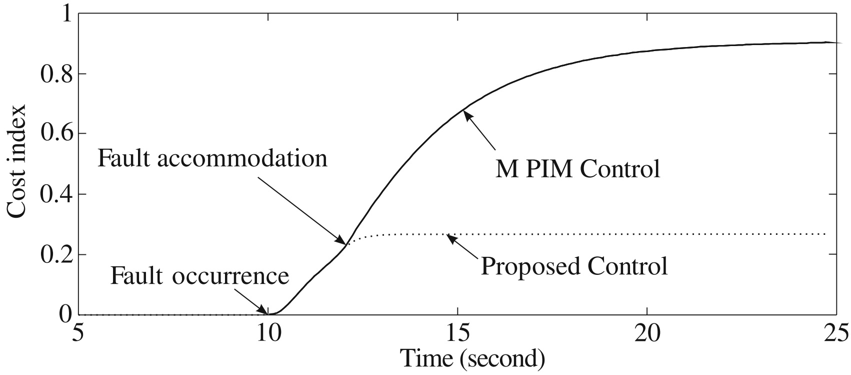


Fig. 9.32 MPIM versus NTT costs

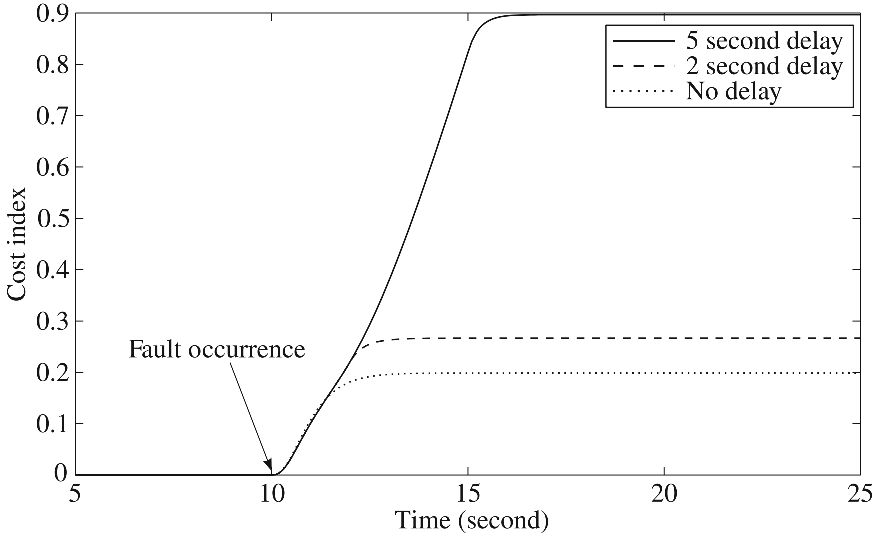


Fig. 9.33 Trajectory tracking cost versus fault accommodation delay

9.4 Fault-Tolerant $\mathcal{H}_\infty$ Design

This section introduces fault-tolerant control strategies that can be applied in a general fault case. It starts with a characterisation of all controllers that stabilise a linear system and also satisfy $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm conditions. This characterisation makes it possible to evaluate the severity of the fault with respect to the control aims and to find methods for redesign the controller automatically.

The complete description of all stabilising controllers is given by the Youla-Kucera or Q-parametrisation. The Youla parametrisation was originally defined by using coprime factorisation. However, it is simple to give an equivalent description of the Youla-Kucera parametrisation as a state-space formulation, where the parametrisation turns out to be an observer-based controller.

One of the facilities by using this parametrisation is that the closed-loop transfer function turns out to be affine in the controller parameters. This affine structure is very useful in connection with design of controllers using optimisation methods. Therefore, the method also fits well to solve the control problem arising when we wish to make a redesign for a controller when a system is in a faulty state.

The salient feature offered by the Youla-Kucera parametrisation is that it offers an elegant and very fast solution to the redesign problem for some classes of faults that leave the system stable with the existing controller but make it unable to meet the required performance.

9.4.1 System Description

Consider the plant

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}_1\boldsymbol{w}(t) + \boldsymbol{B}_2\boldsymbol{u}(t)\\ \boldsymbol{z}(t) &= \boldsymbol{C}_1\boldsymbol{x}(t) + \boldsymbol{D}_{11}\boldsymbol{w}(t) + \boldsymbol{D}_{12}\boldsymbol{u}(t)\\ \boldsymbol{y}(t) &= \boldsymbol{C}_2\boldsymbol{x}(t) + \boldsymbol{D}_{21}\boldsymbol{w}(t) + \boldsymbol{D}_{22}\boldsymbol{u}(t),\end{aligned} \tag{9.95}$$

where $\boldsymbol{x} \in |\mathcal{R}^n$ is the state, $\boldsymbol{u} \in |\mathcal{R}^r$, is the control input, $\boldsymbol{w} \in |\mathcal{R}^k$ is the external input or disturbance, $\boldsymbol{z} \in |\mathcal{R}^l$ is the controlled output and $\boldsymbol{y} \in |\mathcal{R}^m$ is the measurement output. For brevity, it is common to denote this system by the shorter notation

$$\boldsymbol{G}(s) = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{C}_1 & \boldsymbol{C}_2 \\ \boldsymbol{B}_1 & \boldsymbol{D}_{11} & \boldsymbol{D}_{21} \\ \boldsymbol{B}_2 & \boldsymbol{D}_{12} & \boldsymbol{D}_{22} \end{pmatrix},$$

It is assumed that $(\boldsymbol{A}, \boldsymbol{B}_2)$ is stabilisable and $(\boldsymbol{C}_2, \boldsymbol{A})$ is detectable.

Described in transfer function form

$$\begin{pmatrix} \boldsymbol{x}(s) \\ \boldsymbol{y}(s) \end{pmatrix} = \begin{pmatrix} \boldsymbol{G}_{11}(s) & \boldsymbol{G}_{12}(s) \\ \boldsymbol{G}_{21}(s) & \boldsymbol{G}_{22}(s) \end{pmatrix} \begin{pmatrix} \boldsymbol{w}(s) \\ \boldsymbol{u}(s) \end{pmatrix},$$

the transfer function matrix $\boldsymbol{G}(s)$ is decomposed as

$$\boldsymbol{G}(s) = \begin{pmatrix} \boldsymbol{G}_{11}(s) & \boldsymbol{G}_{12}(s) \\ \boldsymbol{G}_{21}(s) & \boldsymbol{G}_{22}(s) \end{pmatrix}.$$

Further, in order to later study the design of diagnosis in conjunction with closed-loop control, let the plant (9.95) be controlled by an output feedback controller

$$\boldsymbol{u}(t) = \boldsymbol{K}\boldsymbol{y}(t).$$

We have now the following definition of stabilisation by output feedback.

Definition 9.3 A proper system $\boldsymbol{G}(s)$ is said to be stabilisable by output feedback if there exists a proper controller $\boldsymbol{K}(s)$ internally stabilising $\boldsymbol{G}(s)$. Moreover, a proper controller $\boldsymbol{K}(s)$ is said to be admissible if it internally stabilises $\boldsymbol{G}(s)$.

Next, we have the following result for the existence of a stabilising controller $\boldsymbol{K}(s)$ for the system $\boldsymbol{G}(s)$, where here and in the following

$$\boldsymbol{K}(s) = \left[\begin{array}{c|c} \boldsymbol{M} & \boldsymbol{N} \\ \hline \boldsymbol{P} & \boldsymbol{Q} \end{array}\right]$$

is used as abbreviation of

$$\boldsymbol{K}(s) = \boldsymbol{P}(s\boldsymbol{I} - \boldsymbol{M})^{-1}\boldsymbol{N} + \boldsymbol{Q}.$$

Lemma 9.4 *There exists a proper controller $\boldsymbol{K}(s)$ achieving internal stability of the closed-loop system if and only if $(\boldsymbol{A},\ \boldsymbol{B}_2)$ is stabilisable and $(\boldsymbol{C}_2,\ \boldsymbol{A})$ is detectable. Further, let $\boldsymbol{F}$ and $\boldsymbol{L}$ be two matrices such that $\boldsymbol{A} + \boldsymbol{B}_2\boldsymbol{F}$ and $\boldsymbol{A} + \boldsymbol{L}\boldsymbol{C}_2$ are stable, then an observer-based stabilising controller is given by*

$$\boldsymbol{K}(s) = \left[\begin{array}{c|c} \boldsymbol{A} + \boldsymbol{B}_2\boldsymbol{F} + \boldsymbol{L}\boldsymbol{C}_2 + \boldsymbol{L}\boldsymbol{D}_{22}\boldsymbol{F} & \boldsymbol{F} \\ \hline -\boldsymbol{L} & \boldsymbol{O} \end{array}\right].$$

It is important to note that the stabilising controller for $\boldsymbol{G}(s)$ depends only on $\boldsymbol{G}_{22}(s)$. We need, therefore, only to look at $\boldsymbol{G}_{22}(s)$ when we are looking for stabilising controllers. This is also the case when we are using the Youla-Kucera parametrisation. In the following the argument s of transfer functions will often be omitted.

9.4.2 Youla-Kucera Parameterisation in Coprime Factorisation Form

First, let us consider two polynomials $m(s)$ and $n(s)$ with real coefficients. m and n are said to be coprime, if their greatest common divisor is 1 (they have no common zeros). It follows from Euclid's algorithm that f and g are coprime if and only if there exists polynomials $x(s)$ and $y(s)$ such that

$$mx + ny = 1. \tag{9.96}$$

This equation is called a Bezout identity. Similarly, the two stable transfer functions m and n are said to be coprime if there exists stable x and y such that Eq. (9.96) is satisfied.

Generally, two stable matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ are right coprime if they have equal number of columns and there exists stable matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ such that

$$(\boldsymbol{X} \quad \boldsymbol{Y})\begin{pmatrix} \boldsymbol{M} \\ \boldsymbol{N} \end{pmatrix} = \boldsymbol{XM} + \boldsymbol{YN} = \boldsymbol{I}.$$

This is equivalent to saying that the matrix $\begin{pmatrix} \boldsymbol{M} \\ \boldsymbol{N} \end{pmatrix}$ is stable left invertible.

Similarly, two stable matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ are left coprime if they have equal number of rows and there exists stable $\boldsymbol{X}$ and $\boldsymbol{Y}$ such that

$$(\boldsymbol{M} \quad \boldsymbol{N})\begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix} = \boldsymbol{MX} + \boldsymbol{NY} = \boldsymbol{I}$$

holds. Equivalently, $(\boldsymbol{M} \quad \boldsymbol{N})$ is stable right invertible.

Now, let $\boldsymbol{G}_{22}(s)$ be a proper real-rational matrix. A right coprime factorisation of $\boldsymbol{G}_{22}(s)$ is a factorisation $\boldsymbol{G}_{22} = \boldsymbol{NM}^{-1}$, where $\boldsymbol{N}$ and $\boldsymbol{M}$ are stable right coprime matrices. Similarly, a left coprime factorisation has the form $\boldsymbol{G}_{22} = \tilde{\boldsymbol{M}}^{-1}\tilde{\boldsymbol{N}}$, where $\tilde{\boldsymbol{N}}$ and $\tilde{\boldsymbol{M}}$ are left coprime. Note that, in these definitions, it is required that the matrices $\boldsymbol{M}$ and $\tilde{\boldsymbol{M}}$ are square and non-singular.

Based on the above, there exists the following result.

Lemma 9.5 *For each proper real-rational matrix $\boldsymbol{G}_{22}(s)$ there exists eight stable matrices satisfying the equations*

$$\boldsymbol{G}_{22} = \boldsymbol{NM}^{-1} = \tilde{\boldsymbol{M}}^{-1}\tilde{\boldsymbol{N}}$$
$$\begin{pmatrix} \tilde{\boldsymbol{X}} & \tilde{\boldsymbol{Y}} \\ -\tilde{\boldsymbol{N}} & \tilde{\boldsymbol{M}} \end{pmatrix}\begin{pmatrix} \boldsymbol{M} & -\boldsymbol{Y} \\ \boldsymbol{N} & \boldsymbol{X} \end{pmatrix} = \boldsymbol{I}.$$

This lemma defines a double coprime factorisation of $\boldsymbol{G}_{22}(s)$. It should be noted that it is always possible to make a coprime factorisation, if the system is stabilisable and detectable.

Now, let $\tilde{\boldsymbol{K}}(s)$ be a stabilising controller for $\boldsymbol{G}_{22}(s)$ and let $\tilde{\boldsymbol{K}}$ have the following factorisation

$$\tilde{\boldsymbol{K}} = \boldsymbol{UV}^{-1} = \tilde{\boldsymbol{V}}^{-1}\tilde{\boldsymbol{U}}.$$

A feedback system with positive feedback is stable if and only if

$$\begin{pmatrix} \boldsymbol{I} & -\tilde{\boldsymbol{K}} \\ -\boldsymbol{G}_{22} & \boldsymbol{I} \end{pmatrix}^{-1} \text{ is stable.}$$

Using the coprime factorisation of $\tilde{\boldsymbol{K}}$ we get the following conditions for internal stability.

Lemma 9.6 *Let $\boldsymbol{G}_{22}(s)$ be a proper real-rational matrix and*

$$\boldsymbol{G}_{22} = \boldsymbol{N}\boldsymbol{M}^{-1} = \tilde{\boldsymbol{M}}^{-1}\tilde{\boldsymbol{N}}$$

be the stable right and left coprime factorisation. Then, there exists a controller

$$\tilde{\boldsymbol{K}}_0 = \boldsymbol{U}_0\boldsymbol{V}_0^{-1} = \tilde{\boldsymbol{V}}_0^{-1}\tilde{\boldsymbol{U}}_0$$

with $\boldsymbol{U}_0$, $\boldsymbol{V}_0$, $\tilde{\boldsymbol{U}}_0$ and $\tilde{\boldsymbol{V}}_0$ stable such that

$$\begin{pmatrix} \tilde{\boldsymbol{V}}_0 & -\tilde{\boldsymbol{U}}_0 \\ -\tilde{\boldsymbol{N}} & \tilde{\boldsymbol{M}} \end{pmatrix} \begin{pmatrix} \boldsymbol{M} & \boldsymbol{U}_0 \\ \boldsymbol{N} & \boldsymbol{V}_0 \end{pmatrix} = \boldsymbol{I}.$$

Based on the above results, it is now possible to give a parametrisation of all controllers that stabilise $\boldsymbol{G}_{22}(s)$.

Theorem 9.6 *Let $\boldsymbol{G}_{22}(s)$ be a proper real-rational matrix and*

$$\boldsymbol{G}_{22} = \boldsymbol{N}\boldsymbol{M}^{-1} = \tilde{\boldsymbol{M}}^{-1}\tilde{\boldsymbol{N}}$$

be the stable right and left coprime factorisation. Then, the set of all proper controllers achieving internal stability is parameterised either by

$$\begin{aligned} \boldsymbol{K} &= (\boldsymbol{U}_0 + \boldsymbol{M}\boldsymbol{Q}_{\mathrm{r}})(\boldsymbol{V}_0 + \boldsymbol{N}\boldsymbol{Q}_{\mathrm{r}})^{-1} \\ &\det(\boldsymbol{I} + \boldsymbol{V}_0^{-1}\boldsymbol{N}\boldsymbol{Q}_{\mathrm{r}})(\infty) \neq 0 \end{aligned} \tag{9.97}$$

for stable $\boldsymbol{Q}_{\mathrm{r}}$ or by

$$\begin{aligned} \boldsymbol{K} &= (\tilde{\boldsymbol{V}}_0 + \boldsymbol{Q}_l\tilde{\boldsymbol{N}})^{-1}(\tilde{\boldsymbol{U}}_0 + \boldsymbol{Q}_l\tilde{\boldsymbol{M}}) \\ &\det(\boldsymbol{I} + \boldsymbol{Q}_l\tilde{\boldsymbol{N}}\tilde{\boldsymbol{V}}_0^{-1})(\infty) \neq 0 \end{aligned} \tag{9.98}$$

for stable $\boldsymbol{Q}_l$, where $\boldsymbol{U}_0$, $\boldsymbol{V}_0$, $\tilde{\boldsymbol{U}}_0$ and $\tilde{\boldsymbol{V}}_0$ stable satisfied the Bezout identities:

$$\begin{aligned} \tilde{\boldsymbol{V}}_0\boldsymbol{M} - \tilde{\boldsymbol{U}}_0\boldsymbol{N} &= \boldsymbol{I} \\ \tilde{\boldsymbol{M}}\boldsymbol{V}_0 - \tilde{\boldsymbol{N}}\boldsymbol{U}_0 &= \boldsymbol{I}. \end{aligned}$$

Moreover, if $\boldsymbol{U}_0$, $\boldsymbol{V}_0$, $\tilde{\boldsymbol{U}}_0$ and $\tilde{\boldsymbol{V}}_0$ are chosen such that

$$\begin{pmatrix} \tilde{\boldsymbol{V}}_0 & -\tilde{\boldsymbol{U}}_0 \\ -\tilde{\boldsymbol{N}} & \tilde{\boldsymbol{M}} \end{pmatrix} \begin{pmatrix} \boldsymbol{M} & \boldsymbol{U}_0 \\ \boldsymbol{N} & \boldsymbol{V}_0 \end{pmatrix} = \boldsymbol{I}.$$

Then we have

$$\begin{aligned} \boldsymbol{K} &= \left(\boldsymbol{U}_0 + \boldsymbol{M}\boldsymbol{Q}_\mathrm{r}\right)\left(\boldsymbol{V}_0 + \boldsymbol{N}\boldsymbol{Q}_\mathrm{r}\right)^{-1} \\ &= \left(\tilde{\boldsymbol{V}}_0 + \boldsymbol{Q}_\mathrm{r}\tilde{\boldsymbol{N}}\right)^{-1}\left(\tilde{\boldsymbol{U}}_0 + \boldsymbol{Q}_\mathrm{r}\tilde{\boldsymbol{M}}\right) \\ &= \mathcal{F}_l(\boldsymbol{J}_\mathrm{r}, \boldsymbol{Q}_\mathrm{r}), \end{aligned} \tag{9.99}$$

where

$$\boldsymbol{J}_\mathrm{r} = \begin{pmatrix} \boldsymbol{U}_0\boldsymbol{V}_0^{-1} & \tilde{\boldsymbol{V}}_0^{-1} \\ \boldsymbol{V}_0^{-1} & -\boldsymbol{V}_0^{-1}\boldsymbol{N} \end{pmatrix}$$

and $\boldsymbol{Q}_\mathrm{r}$ *is stable and satisfies that* $(\boldsymbol{I} + \boldsymbol{V}_0^{-1}\boldsymbol{N}\boldsymbol{Q}_y)(\infty)$ *is invertible.*

The Youla-Kucera parametrisation is shown in Fig. 9.34.

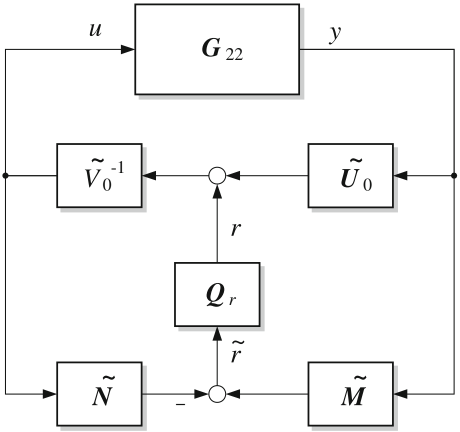


Fig. 9.34 Controller structure for the Youla-Kucera parametrisation

9.4.3 Parametrisation in the State-Space Form

The Youla-Kucera parametrisation derived in the above section was based on coprime factorisation, which may not be the form in which a particular fault-tolerant control problem is described. Further, popular toolboxes support a state-space description (e.g. MATLAB). Therefore, a state-space description will be given in this section together with a representation of the closed-loop transfer function as function of the free stable parameter $\boldsymbol{Q}$.

For the coprime factorisation in a state-space form using state feedback and observers, the following result is available: Let two coprime factorisations of $\boldsymbol{G}_{22}(s)$ be given by

$$\begin{pmatrix} \boldsymbol{M} & \boldsymbol{U}_0 \\ \boldsymbol{N} & \boldsymbol{V}_0 \end{pmatrix} = \left(\begin{array}{c|cc} \boldsymbol{A}+\boldsymbol{B}_2\boldsymbol{F} & \boldsymbol{F} & \boldsymbol{C}_2+\boldsymbol{D}_{22}\boldsymbol{F} \\ \boldsymbol{B}_2 & \boldsymbol{I} & \boldsymbol{D}_{22} \\ -\boldsymbol{L} & \boldsymbol{O} & \boldsymbol{I} \end{array} \right)$$

$$\begin{pmatrix} \tilde{\boldsymbol{V}}_0 & -\tilde{\boldsymbol{U}}_0 \\ -\tilde{\boldsymbol{N}} & \tilde{\boldsymbol{M}} \end{pmatrix} = \left(\begin{array}{c|cc} \boldsymbol{A}+\boldsymbol{L}\boldsymbol{C}_2 & \boldsymbol{F} & \boldsymbol{C}_2 \\ -(\boldsymbol{B}_2+\boldsymbol{L}\boldsymbol{D}_{22}) & \boldsymbol{I} & -\boldsymbol{D}_{22} \\ \boldsymbol{L} & \boldsymbol{O} & \boldsymbol{I} \end{array} \right),$$

where $\boldsymbol{F}$ and $\boldsymbol{L}$ are chosen such that $\boldsymbol{A}+\boldsymbol{B}_2\boldsymbol{F}$ and $\boldsymbol{A}+\boldsymbol{L}\boldsymbol{C}_2$ are both stable. It is now quite simple to give a state-space realisation of the $\boldsymbol{Q}$-parametrisation of all internal stabilising controllers. From Theorem 9.6 we have the linear fractional transformation formulation of all stabilising controllers. Using the state-space description of the coprime factorisation in $\boldsymbol{J}_y$ we get the following result.

Theorem 9.7 *Let $\boldsymbol{F}$ and $\boldsymbol{L}$ be such that $\boldsymbol{A}+\boldsymbol{B}_2\boldsymbol{F}$ and $\boldsymbol{A}+\boldsymbol{L}\boldsymbol{C}_2$ are stable. Then all controllers that internally stabilise $\boldsymbol{G}(s)$ can be parameterised as the transfer matrix from $\mathbf{y}$ to $\boldsymbol{u}$ given by $\mathcal{F}_l(\boldsymbol{J}_y, \boldsymbol{Q})$, where*

$$\boldsymbol{J}_y = \left(\begin{array}{c|cc} \boldsymbol{A}+\boldsymbol{B}_2\boldsymbol{F}+\boldsymbol{L}\boldsymbol{C}_2+\boldsymbol{L}\boldsymbol{D}_{22}\boldsymbol{F} & \boldsymbol{F} & -(\boldsymbol{C}_2+\boldsymbol{D}_{22}\boldsymbol{F}) \\ -\boldsymbol{L} & \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{B}_2+\boldsymbol{L}\boldsymbol{D}_{22} & \boldsymbol{I} & -\boldsymbol{D}_{22} \end{array} \right)$$

with any $\boldsymbol{Q} \in \mathcal{RH}_\infty$ and $\boldsymbol{I}+\boldsymbol{D}_{22}\boldsymbol{Q}(\infty)$ is non-singular.

The controller given in the theorem is sometimes called the $\boldsymbol{Q}$-observer-based controller. For $\boldsymbol{Q}=\boldsymbol{O}$ the nominal controller turns out to be a standard full-order observer-based controller. Moreover, it can be shown that the separation between the design of the state feedback gain $\boldsymbol{F}$ and the observer gain $\boldsymbol{L}$ is still valid as well as a separation between the nominal controller and the $\boldsymbol{Q}$ parameter. This can be shown by setting up a state-space description of the controller together with the $\boldsymbol{Q}$ parameter and use the state vector $\bar{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{x} & \boldsymbol{x}-\hat{\boldsymbol{x}} & \boldsymbol{x}_q \end{pmatrix}$, where $\boldsymbol{x}_q$ is the state vector for $\boldsymbol{Q}$.

Next, let us look at the closed-loop transfer function when we have applied a $\boldsymbol{Q}$-parameterised controller as given in Theorem 9.7. The closed-loop transfer function is given by the following linear fractional transformation:

$$\boldsymbol{z} = \mathcal{F}_l(\boldsymbol{G}, \boldsymbol{K})\boldsymbol{w} = \mathcal{F}_l(\boldsymbol{G}, \mathcal{F}_l(J_y, \boldsymbol{Q}))\boldsymbol{w} = \mathcal{F}_l(\boldsymbol{T}, \boldsymbol{Q})\boldsymbol{w}.$$

We need now just to give a state-space description of $\boldsymbol{T}$. By using straightforward and tedious algebra, we get the following result.

Theorem 9.8 *Let $\boldsymbol{F}$ and $\boldsymbol{L}$ be such that $\boldsymbol{A}+\boldsymbol{B}_2\boldsymbol{F}$ and $\boldsymbol{A}+\boldsymbol{L}\boldsymbol{C}_2$ are stable. Then, the set of a closed-loop transfer matrices from $\boldsymbol{w}$ to $\boldsymbol{z}$ achievable by an stabilising proper controller is equal to*

$$\mathcal{F}_l(\boldsymbol{T}, \boldsymbol{Q}) = \boldsymbol{T}_{11} + \boldsymbol{T}_{12}\boldsymbol{Q}\boldsymbol{T}_{21}, \ \boldsymbol{Q} \in \mathcal{RH}_\infty, \ \boldsymbol{I}+\boldsymbol{D}_{22}\boldsymbol{Q}(\infty),$$

where $\boldsymbol{T}$ is given by

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{T}_{11} & \boldsymbol{T}_{12} \\ \boldsymbol{T}_{21} & \boldsymbol{T}_{22} \end{pmatrix} = \left(\begin{array}{cc|cc} \boldsymbol{A}+\boldsymbol{B}_2\boldsymbol{F} & -\boldsymbol{B}_2\boldsymbol{F} & \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \boldsymbol{O} & \boldsymbol{A}+\boldsymbol{L}\boldsymbol{C}_2 & \boldsymbol{B}_1+\boldsymbol{L}\boldsymbol{D}_{21} & \boldsymbol{O} \\ \hline \boldsymbol{C}_1+\boldsymbol{D}_{12}\boldsymbol{F} & -\boldsymbol{D}_{12}\boldsymbol{F} & \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{O} & \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{O} \end{array}\right).$$

It is important to note that the closed-loop transfer matrix $\boldsymbol{T}$ is an affine function of the controller parameter matrix $\boldsymbol{Q}$, since $\boldsymbol{T}_{22} = \boldsymbol{O}$. This is the reason why the $\boldsymbol{Q}$-parametrisation is so useful, particularly in connection with optimisation of controllers by using numerical tools.

9.4.4 Simultaneous Design of the Controller and the Residual Generator

In the closed loop, there is an interaction between the sensitivity of the residual generated by a fault detection filter and the natural suppression of any fault within a closed loop. The design of closed-loop control and residual generator can, therefore, be considered an integrated design problem. Consider, the simultaneous design of the feedback controller and the residual generator. The design setup is illustrated in Fig. 9.35 (left). It uses the standard problem philosophy.

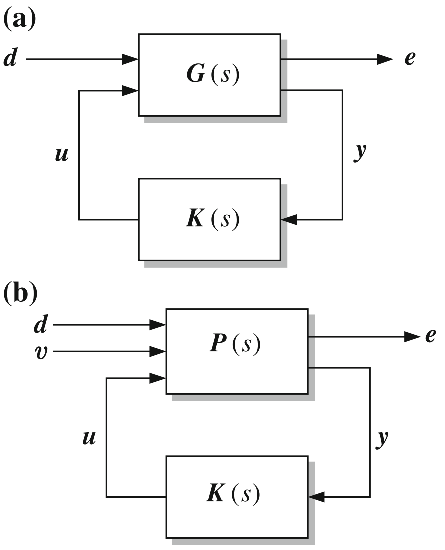


Fig. 9.35 Control system in standard configuration (*left*) and in generalised setup for fault-tolerant control (*right*)

As stated earlier, the standard design provides a controller $\boldsymbol{K}(s)$ for which the closed loop is internally stable and a suitable norm of the closed-loop transfer function from $\boldsymbol{w}$ to $\boldsymbol{z}$ is minimised or made smaller than a pre-specified level.

Instead of using a standard controller as shown in Fig. 9.35 (left), a controller with two outputs can be employed:

$$\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{a} \end{pmatrix} = \begin{pmatrix} \boldsymbol{K}_1 \\ \boldsymbol{K}_2 \end{pmatrix} \boldsymbol{y}.$$

The additional output signal $\boldsymbol{a}$ is a diagnostic signal, which will be applied to derive an estimate of faults in the controlled system.

Let the open-loop transfer function be given by:

$$\begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{G}_{\mathrm{ed}} & \boldsymbol{G}_{\mathrm{ef}} & \boldsymbol{G}_{\mathrm{eu}} \\ \boldsymbol{G}_{\mathrm{yd}} & \boldsymbol{G}_{\mathrm{yf}} & \boldsymbol{G}_{\mathrm{yu}} \end{pmatrix} \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{f} \\ \boldsymbol{u} \end{pmatrix} \tag{9.100}$$

To obtain a good estimation of the individual faults, fault models are included in the generalised system as frequency weightings on the faults signals

$$\boldsymbol{f} = \boldsymbol{W}_{\mathrm{f}}(s)\boldsymbol{v},$$

where $\boldsymbol{v}$ is a signal that is anticipated to have a flat power spectrum. The generalised setup is shown in Fig. 9.35 (right).

Now we need to formulate the design setup in Fig. 9.35 (right) as a standard design problem as illustrated in Fig. 9.35 (left). For doing this, define an additional output $\boldsymbol{r}$ as the fault estimation error:

$$\boldsymbol{r} = \boldsymbol{f} - \boldsymbol{a}. \tag{9.101}$$

This is the standard way of formulating a filter design problem in the standard setup. The generalised system $\boldsymbol{P}(s)$ is then given by:

$$\begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{r} \\ \boldsymbol{y} \end{pmatrix} = \boldsymbol{P}(s) \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{v} \\ \boldsymbol{u} \end{pmatrix} \tag{9.102}$$

with

$$\boldsymbol{P}(s) = \left(\begin{array}{cc|cc} \boldsymbol{G}_{\mathrm{ed}} & \boldsymbol{G}_{\mathrm{ef}}\boldsymbol{W}_{\mathrm{f}} & \boldsymbol{G}_{\mathrm{eu}} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{W}_{\mathrm{f}} & \boldsymbol{O} & -\boldsymbol{I} \\ \hline \boldsymbol{G}_{\mathrm{yd}} & \boldsymbol{G}_{\mathrm{yf}}\boldsymbol{W}_{\mathrm{f}} & \boldsymbol{G}_{\mathrm{yu}} & \boldsymbol{O} \end{array}\right).$$

Using the system setup in (9.102) and the controller

$$\boldsymbol{u} = \boldsymbol{K}(s)\boldsymbol{y}$$

we get the following closed-loop transfer function

$$\begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{r} \end{pmatrix} = \boldsymbol{T}_{\mathrm{cl}}(s) \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{v} \end{pmatrix}$$

with

$$\begin{aligned} \boldsymbol{T}_{\mathrm{cl}}(s) = &\begin{pmatrix} \boldsymbol{G}_{\mathrm{ed}} & \boldsymbol{G}_{\mathrm{ef}}\boldsymbol{W}_{\mathrm{f}} \\ \boldsymbol{O} & \boldsymbol{W}_{\mathrm{f}} \end{pmatrix} + \\ &\begin{pmatrix} \boldsymbol{G}_{\mathrm{eu}} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{I} \end{pmatrix} \boldsymbol{K}(s)(\boldsymbol{I} - \begin{pmatrix} \boldsymbol{G}_{\mathrm{yu}} & \boldsymbol{O} \end{pmatrix} \boldsymbol{K}(s))^{-1} \begin{pmatrix} \boldsymbol{G}_{\mathrm{yd}} & \boldsymbol{G}_{\mathrm{yf}}\boldsymbol{W}_{\mathrm{f}} \end{pmatrix}. \end{aligned}$$

For simplicity, assume that $\boldsymbol{G}(s)$ is open-loop stable (the unstable case can be dealt with as well in this methodology, but is computationally more difficult). Then, the Youla-Kucera parameterisation of all stabilising controllers can be obtained by making the substitutions

$$\begin{aligned} \boldsymbol{Q}(s) &= \boldsymbol{K}(s)(\boldsymbol{I} - \begin{pmatrix} \boldsymbol{G}_{\mathrm{yu}} & \boldsymbol{O} \end{pmatrix} \boldsymbol{K}(s))^{-1} \\ \boldsymbol{K}(s) &= \boldsymbol{Q}(s)(\boldsymbol{I} + \begin{pmatrix} \boldsymbol{G}_{\mathrm{yu}} & \boldsymbol{O} \end{pmatrix} \boldsymbol{Q}(s))^{-1}, \end{aligned} \tag{9.103}$$

where $\boldsymbol{Q}(s)$ is a stable proper transfer function, namely the Youla parameter. Further, let $\boldsymbol{Q}(s)$ be partitioned as:

$$\boldsymbol{Q}(s) = \begin{pmatrix} \boldsymbol{Q}_1(s) \\ \boldsymbol{Q}_2(s) \end{pmatrix}.$$

Then, the following equation for the closed-loop transfer function $\boldsymbol{T}_{\mathrm{cl}}$ is obtained:

$$\boldsymbol{T}_{\mathrm{cl}}(s) = \begin{pmatrix} \boldsymbol{G}_{\mathrm{ed}} + \boldsymbol{G}_{\mathrm{eu}}\boldsymbol{Q}_1\boldsymbol{G}_{\mathrm{yd}} & \boldsymbol{G}_{\mathrm{ef}}\boldsymbol{W}_{\mathrm{f}} + \boldsymbol{G}_{\mathrm{eu}}\boldsymbol{Q}_1\boldsymbol{G}_{\mathrm{yf}}\boldsymbol{W}_{\mathrm{f}} \\ -\boldsymbol{Q}_2\boldsymbol{G}_{\mathrm{yd}} & \boldsymbol{W}_{\mathrm{f}} - \boldsymbol{Q}_2\boldsymbol{G}_{\mathrm{yf}}\boldsymbol{W}_{\mathrm{f}} \end{pmatrix}. \tag{9.104}$$

Note that $\boldsymbol{Q}_1$ only appears in the first row of $\boldsymbol{T}_{\mathrm{cl}}$ and $\boldsymbol{Q}_2$ only in the second row. A separation between the design of $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ has, therefore, been obtained, which is a salient feature of this design approach.

Calculating $\boldsymbol{K}(s)$ directly from (9.103) results in the following equation:

$$\begin{aligned} \boldsymbol{K}(s) &= \begin{pmatrix} \boldsymbol{Q}_1(s)(\boldsymbol{I} + \boldsymbol{G}_{\mathrm{yu}}\boldsymbol{Q}_1(s))^{-1} \\ \boldsymbol{Q}_2(s)(\boldsymbol{I} + \boldsymbol{G}_{\mathrm{yu}}\boldsymbol{Q}_1(s))^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{Q}_1(s)(\boldsymbol{I} + \boldsymbol{G}_{\mathrm{yu}}\boldsymbol{Q}_1(s))^{-1} \\ \boldsymbol{Q}_2(s)(\boldsymbol{I} - \boldsymbol{G}_{\mathrm{yu}}\boldsymbol{K}_1(s)) \end{pmatrix}. \end{aligned} \tag{9.105}$$

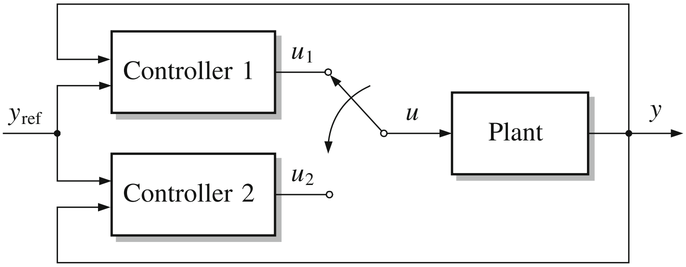


Fig. 9.36 Two-controller scheme

The result indicates that also the original controller structure is separated in a design of the feedback controller $\boldsymbol{K}_1(s)$ and a design of the fault detection filter $\boldsymbol{K}_2(s)$, which depends upon the controller $\boldsymbol{K}_1(s)$.

This result is essential for proper design of residual generators working in closed loop. It is also important for the redesign problem since the effect that the redesigned controller has on the diagnostic filters cannot be ignored.

9.5 Handling the Fault Recovery Transients

9.5.1 Switching Between Controllers

In the previous sections, controller reconfiguration or accommodation often amounts to switching from the nominal controller to a newly designed controller or from one to another element of a bank of controllers. When the considered control laws are of the state feedback or output feedback type, no precaution is required to switch between controllers with the same reference input. However, the situation is different for dynamical controllers. Indeed, the state of the controller which is not in the loop has to be initialised properly before this controller is introduced in the loop, in order to avoid bumps in the system response. One method to achieve this goal is presented here. It amounts to feeding back to each controller, be it active in the loop or not, the manipulated variable actually applied to the process (namely the process input). This mechanism is similar to an anti-windup strategy, which is normally used to handle actuator saturation in a control loop.

Without loss of generality, a situation with two controllers is considered here, so that one controller, say controller 1, is active in the loop, and controller 2 is the controller towards which switching occurs in the fault case (Fig. 9.36). Both controllers are supposed to be described by a linear state-space model of the form

$$\begin{aligned} \dot{\boldsymbol{x}}_{ci}(t) &= \boldsymbol{A}_{ci}\boldsymbol{x}_{ci}(t) + \boldsymbol{B}_{ci}\boldsymbol{y}_{\text{ref}}(t) + \boldsymbol{E}_{ci}\,\boldsymbol{y}(t) \quad && \boldsymbol{x}_{ci}(0) = \boldsymbol{x}_{ci}^{0} \\ \boldsymbol{u}_i(t) &= \boldsymbol{C}_{ci}\boldsymbol{x}_{ci}(t) + \boldsymbol{D}_{ci}\boldsymbol{y}_{\text{ref}}(t) + \boldsymbol{F}_{ci}\boldsymbol{y}(t) \quad && i = 1, 2, \end{aligned} \tag{9.106}$$

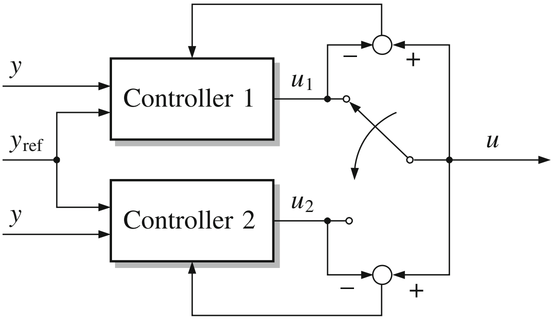


Fig. 9.37 Two-controller scheme with anti-windup mechanism

where $\boldsymbol{x}_{ci}(t), \boldsymbol{u}_i(t), \boldsymbol{y}(t)$ and $\boldsymbol{y}_{\text{ref}}(t)$ are respectively the controller state, the controller output, the measured plant output and the reference.

As explained in the previous paragraph, to obtain a smooth switching towards controller 2, the state of this controller must be properly initialised. This can be achieved thanks to an observer-based anti-windup mechanism. It amounts to feeding back the difference $\boldsymbol{u}(t)-\boldsymbol{u}_2(t)$ between the plant input and the output of controller 1 towards controller 2:

$$\begin{aligned} \dot{\boldsymbol{x}}_{c2}(t) &= \boldsymbol{A}_{c2}\boldsymbol{x}_{c2}(t) + \boldsymbol{B}_{c2}\boldsymbol{y}_{\text{ref}}(t) + \boldsymbol{E}_{c2}\,\boldsymbol{y}(t) + \boldsymbol{L}_2(\boldsymbol{u}(t) - \boldsymbol{u}_2(t)) \\ \boldsymbol{u}_2(t) &= \boldsymbol{C}_{c2}\boldsymbol{x}_{c2}(t) + \boldsymbol{D}_{c2}\boldsymbol{y}_{\text{ref}}(t) + \boldsymbol{F}_{c2}\boldsymbol{y}(t). \end{aligned} \tag{9.107}$$

Substituting the output equation for $\boldsymbol{u}_2(t)$ in the state equation of (9.107) yields

$$\begin{aligned} \dot{\boldsymbol{x}}_{c2}(t) &= (\boldsymbol{A}_{c2} - \boldsymbol{L}_2\boldsymbol{C}_{c2})\boldsymbol{x}_{c2}(t) + (\boldsymbol{B}_{c2} - \boldsymbol{L}_2\boldsymbol{D}_{c2})\boldsymbol{y}_{\text{ref}}(t) \\ &\quad +(\boldsymbol{E}_{c2} - \boldsymbol{L}_2\boldsymbol{F}_{c2})\,\boldsymbol{y}(t) + \boldsymbol{L}_2\boldsymbol{u}(t) \\ \boldsymbol{u}_2(t) &= \boldsymbol{C}_{c2}\boldsymbol{x}_{c2}(t) + \boldsymbol{D}_{c2}\boldsymbol{y}_{\text{ref}}(t) + \boldsymbol{F}_{c2}\boldsymbol{y}(t) \end{aligned} \tag{9.108}$$

and shows that $\boldsymbol{L}_2$ should be chosen in such a way that $(\boldsymbol{A}_{c2} - \boldsymbol{L}_2\boldsymbol{C}_{c2})$ has all its eigenvalues inside the open left-half plane in order for $\boldsymbol{x}_{c2}(t)$ to reach a steady-state value when controller 2 is not inserted in the loop, in the absence of change in $\boldsymbol{y}_{\text{ref}}(t)$, $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$. Possible options consist in choosing $\boldsymbol{L}_2$ so that all eigenvalues lie at the origin, or to use $\boldsymbol{L}_2 = \boldsymbol{B}_{c2}\boldsymbol{D}_{c2}^{-1}$, which corresponds to the so-called conditioning technique. The latter approach requires a square full-rank matrix $\boldsymbol{D}_{c2}$, although this conditions can be weakened. It also requires that the zeros of the controller lie in the open left-half plane.

Obviously, for reason of symmetry, to allow switching from controller 2 to controller 1, the latter controller must be provided with a similar anti-windup feature. Its state-space equation is thus written like (9.108), with index 1 substituted for 2. The resulting block diagram is given in Fig. 9.37.

9.5.2 Progressive Fault Accommodation

In the ideal fault-tolerant linear quadratic problem described in Sect. 7.2 the fault detection, isolation and estimation process take no time. However, it has been already noted that in practice, three time periods exist:

Time window	System situation	System	Controller
$[0, t_\mathrm{f}[$	Nominal operation	$(\boldsymbol{A}\,, \boldsymbol{B})$	$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}'P\boldsymbol{x}$
$[t_\mathrm{f}, t_\mathrm{a}[$	Fault detection, isolation and estimation process, fault accommodation process delay	$(\boldsymbol{A}\,, \boldsymbol{B}_\mathrm{f})$	$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^\mathrm{T}\boldsymbol{P}\boldsymbol{x}$
$[t_\mathrm{a}, \infty)$	Fault is accommodated	$(\boldsymbol{A}\,, \boldsymbol{B}_\mathrm{f})$	$\boldsymbol{u}_\mathrm{f} = -\boldsymbol{R}^{-1}\boldsymbol{B}_\mathrm{f}^\mathrm{T}\,\boldsymbol{P}_\mathrm{f}\boldsymbol{x}$

During the time period $[t_\mathrm{f}, t_\mathrm{a}[$ the faulty system $(\boldsymbol{A}, \boldsymbol{B}_\mathrm{f})$ is still controlled by the nominal control $\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^\mathrm{T}\boldsymbol{P}\boldsymbol{x}$. This control is optimal for $(\boldsymbol{A}, \boldsymbol{B})$ and the closed loop $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\mathrm{T}\boldsymbol{P}$ is stable, but no guarantee can be given when $\boldsymbol{B}$ is replaced by $\boldsymbol{B}_\mathrm{f}$ and the closed loop $\boldsymbol{A} - \boldsymbol{B}_\mathrm{f}\boldsymbol{R}^{-1}\boldsymbol{B}^\mathrm{T}\boldsymbol{P}$ may be unstable. If $t_\mathrm{a} - t_\mathrm{f}$ is not small enough, although the new control law $\boldsymbol{u}_\mathrm{f}$ will recover the system stability and provide the best possible performance when applied, the system state may violate some physical limits or it may lead to a non-admissible value of the system cost. Note that physical limits are not formalised in the standard LQ problem setting, but they are usually taken into account by an appropriate choice of the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$.

Therefore, to solve practical problems the fault detection, isolation and estimation process delay as well as the fault accommodation process delay have to be made as small as possible. As far as fault accommodation is concerned, this can be obtained by two complementary strategies:

- Design an algorithm that computes the accommodated control $\boldsymbol{u}_\mathrm{f}$ (i.e. that solves the algebraic Riccati equation) in minimum time,
- Design an algorithm that computes a sequence of controls that will eventually converge to $\boldsymbol{u}_\mathrm{f}$ and will stabilise the system as soon as possible. Such an algorithm belongs to the family of "anytime" algorithms, which means that the result of any iteration is acceptable, and it will improve as the number of iterations increases. This is the *progressive accommodation* strategy.

Newton-Raphson iteration scheme for solving the algebraic Riccati equation. The Newton-Raphson iteration scheme has been proposed in the literature as an effective way of solving the algebraic Riccati equation. Let $\boldsymbol{P}_i$ be the unique solution of the Lyapunov equation

$$\boldsymbol{P}_i(\boldsymbol{A} - \boldsymbol{B}_\mathrm{f}\boldsymbol{F}_{i-1}) + (\boldsymbol{A} - \boldsymbol{B}_\mathrm{f}\boldsymbol{F}_{i-1})^\mathrm{T}\boldsymbol{P}_i = -\boldsymbol{Q} - \boldsymbol{F}_{i-1}^\mathrm{T}\boldsymbol{R}\boldsymbol{F}_{i-1}, \tag{9.109}$$

where

$$\boldsymbol{F}_i = \boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{f}}\boldsymbol{P}_i \tag{9.110}$$

for all $i = 1, 2, \ldots$ and the initial $\boldsymbol{F}_0$ is given. If $\boldsymbol{A} - \boldsymbol{B}_{\mathrm{f}}\boldsymbol{F}_0$ is stable, then all matrices $\boldsymbol{P}_i$ are positive definite, and one has the convergence result

$$\begin{array}{l}(1)\ \boldsymbol{P}_0 \geq \boldsymbol{P}_1 \geq \cdots \geq \boldsymbol{P}_i \geq \boldsymbol{P}_{i+1} \geq \cdots \geq \boldsymbol{P}_{\mathrm{f}}, \quad i = 1, 2, \ldots \\ (2)\ \lim_{i\to\infty} \boldsymbol{P}_i = \boldsymbol{P}_{\mathrm{f}},\end{array} \tag{9.111}$$

where $\boldsymbol{P}_{\mathrm{f}}$ is the solution of the algebraic Riccati equation

$$\boldsymbol{P}_{\mathrm{f}}(\boldsymbol{A} - \boldsymbol{B}_{\mathrm{f}}\boldsymbol{F}_{\mathrm{f}}) + (\boldsymbol{A} - \boldsymbol{B}_{\mathrm{f}}\boldsymbol{F}_{\mathrm{f}})^{\mathrm{T}}\boldsymbol{P}_{\mathrm{f}} = -\boldsymbol{Q} - \boldsymbol{F}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{F}_{\mathrm{f}}.$$

Progressive Accommodation (PA) scheme. The PA scheme is based on the Newton-Raphson algorithm. Assume that iteration i takes a time Δ_i, and consider the sequence

$$t_i = t_{\mathrm{init}} + \sum_{j=1}^{i} \Delta_j, \quad i = 1, 2, \ldots,$$

where t_{init} is the time at which the Newton-Raphson algorithm is initialised after the fault has been detected, isolated and estimated (note that $t_{\mathrm{f}} < t_{fdi} < t_{\mathrm{init}}$). t_i is the time at which the result $\boldsymbol{F}_i$ becomes available (note that the constancy of Δ_i is not necessary, the scheme can, therefore, be employed whatever the tasks scheduling strategy of the FTC computer). The idea of progressive accommodation is to apply the feedback control law $\boldsymbol{u}_i = -\boldsymbol{F}_i\boldsymbol{x}$ on the time interval $\left[t_i, t_{i+1}\right[$. As a result, the system behaviour after the fault occurrence is

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A} - \boldsymbol{B}_{\mathrm{f}}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P})\boldsymbol{x}(t), \quad t \in [t_{\mathrm{f}}, t_{\mathrm{init}}[\tag{9.112}$$

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}_{\mathrm{f}}\boldsymbol{F}_0)\boldsymbol{x}, \quad t \in [t_{\mathrm{init}}, t_1[\tag{9.113}$$

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}_{\mathrm{f}}\boldsymbol{F}_i)\boldsymbol{x}, \quad t \in [t_i, t_{i+1}[\quad i = 1, 2, \ldots, \tag{9.114}$$

where $\boldsymbol{F}_0$ is the Newton-Raphson initialisation at time t_{init}. It can be shown from (9.111) that if the system $(\boldsymbol{A}, \boldsymbol{B}_{\mathrm{f}})$ is stabilised by $\boldsymbol{F}_0$, then it is stabilised by all $\boldsymbol{F}_i$, and each $\boldsymbol{F}_i$ is better than the previous one with respect to the LQ cost. Moreover, PA results in a lower cost than the one associated with controlling the system by the nominal control until the Newton-Raphson algorithm has converged (which means the accommodated solution is computed) and then applying the accommodated control. Figure 9.38 shows the fault-tolerant system architecture using the PA scheme.

$\boldsymbol{A} - \boldsymbol{B}_{\mathrm{f}}\boldsymbol{F}_0$ being stable is only a sufficient condition for the PA procedure to converge. Convergence may be obtained in some cases, even when the initial feedback does not stabilise the system. This happens in the following example.

Example 9.6 Progressive accommodation of a first-order system
Consider the following LQ problem

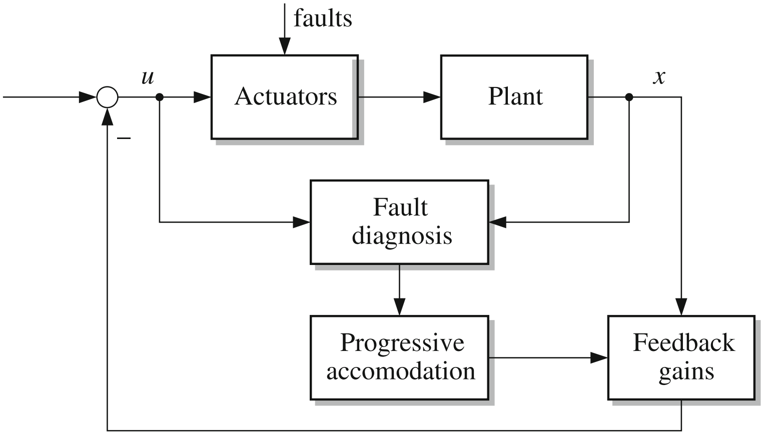


Fig. 9.38 Progressive accommodation scheme

$$\dot{\boldsymbol{x}}(t) = -\boldsymbol{x}(t) + \boldsymbol{u}(t), \quad \boldsymbol{x}(0) = 4$$
$$\boldsymbol{J} = \int_0^\infty [\boldsymbol{x}^2(t) + \boldsymbol{u}^2(t)]\mathrm{d}t.$$

The nominal Algebraic Riccati Equation is $\boldsymbol{P}^2 + 2\boldsymbol{P} - 1 = \boldsymbol{O}$ leading to the optimal control $\boldsymbol{u}(t) = (1 - \sqrt{2})\boldsymbol{x}(t)$, and closed-loop behaviour $\dot{\boldsymbol{x}} = -\sqrt{2}\boldsymbol{x}$. Let the faulty system be

$$\dot{\boldsymbol{x}}(t) = -\boldsymbol{x} - 2\sqrt{2}\boldsymbol{u}(t) \qquad t \geq 1$$

Under the nominal control, the faulty system behaviour is $\dot{\boldsymbol{x}}(t) = (3 - 2\sqrt{2})\boldsymbol{x}(t)$ which is unstable. The new Algebraic Riccati Equation is $8\boldsymbol{P}_\mathrm{f}^2 + 2\boldsymbol{P}_\mathrm{f} - 1 = \boldsymbol{O}$ whose stable solution gives the optimal control of the faulty system $\boldsymbol{u}_\mathrm{f} = \frac{\sqrt{2}}{2}\boldsymbol{x}$ and the closed-loop behaviour $\dot{\boldsymbol{x}}(t) = -3\boldsymbol{x}(t)$. The Newton-Raphson algorithm results in

$$\boldsymbol{P}_i = \frac{1 + 8\boldsymbol{P}_{i-1}^2}{2(1 + 8\boldsymbol{P}_{i-1})},$$

which converges to the solution of the new Algebraic Riccati Equation. The table below shows the evolution of $\boldsymbol{P}_i$, while Fig. 9.39 shows the evolution of the system state when fault accommodation is applied after convergence of the Newton-Raphson scheme (which takes 3 iterations, with $\Delta = 1s$) (classical approach, dashed line), and using the progressive accommodation scheme (continuous line).

i	0	1	2	3	4	5
$k_i \times 10^2$	41.42	27.5	25.08	25	25	25

□

Example 9.7 Progressive accommodation in nominal trajectory tracking
Let us consider again the second-order system of the nominal trajectory tracking example, under the fault

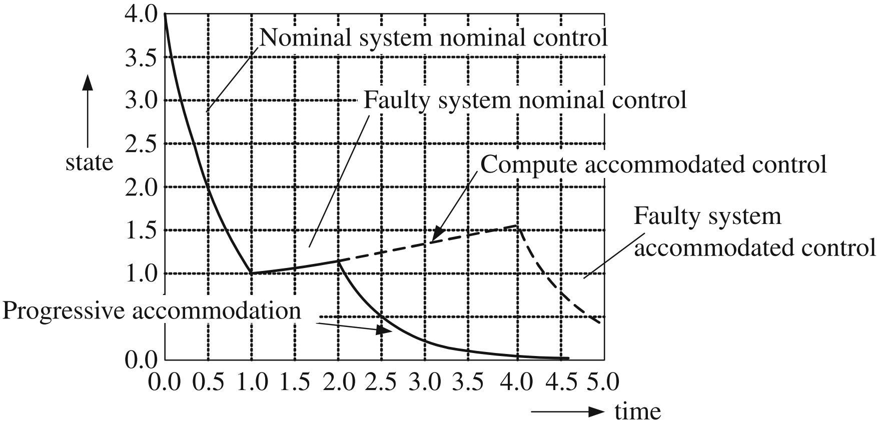


Fig. 9.39 Comparison of the classical and the progressive accommodation schemes

$$\boldsymbol{A}_\mathrm{f} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \boldsymbol{B}_\mathrm{f} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

associated with the fault parameter $\theta = 1$. Remember that in this case, the pseudo-inverse method leads to an unstable system, and the modified pseudo-inverse method (MPIM) has to be applied. The alternative approach based on optimal nominal trajectory tracking gives the control

$$\boldsymbol{u}_\mathrm{f}(t) = \left(\boldsymbol{K}_n + \boldsymbol{R}^{-1}\boldsymbol{B}_\mathrm{f}^\mathrm{T}\boldsymbol{G}\right)\boldsymbol{x}_\mathrm{n}(t) + \boldsymbol{R}^{-1}\boldsymbol{B}_\mathrm{f}^\mathrm{T}\boldsymbol{H}\boldsymbol{x}_\mathrm{f}(t)$$

where

$$\boldsymbol{H} = \begin{pmatrix} -0.433 & -0.067 \\ -0.067 & -0.433 \end{pmatrix} \quad \text{and} \quad \boldsymbol{G} = \begin{pmatrix} 2.134 & 0.5 \\ 1.800 & -0.5 \end{pmatrix}$$

satisfy the Riccati equation

$$\boldsymbol{H}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\boldsymbol{H} - 2\boldsymbol{H} - \boldsymbol{I}_2 = \boldsymbol{O} \tag{9.115}$$

and the Lyapunov equation

$$\boldsymbol{H}\begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix} + \boldsymbol{G}\begin{pmatrix} -2 & 0 \\ -5 & -1 \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}\boldsymbol{G} + \boldsymbol{I}_2 = \boldsymbol{O}. \tag{9.116}$$

Figure 9.31 that shows the state $\boldsymbol{x}_\mathrm{n}$, MPIM $\boldsymbol{x}_\mathrm{f}^\mathrm{MPIM}$ and nominal trajectory tracking $\boldsymbol{x}_\mathrm{f}$ trajectories, for three different fault detection, isolation, diagnosis and accommodation delays is recalled here as Fig. 9.40.

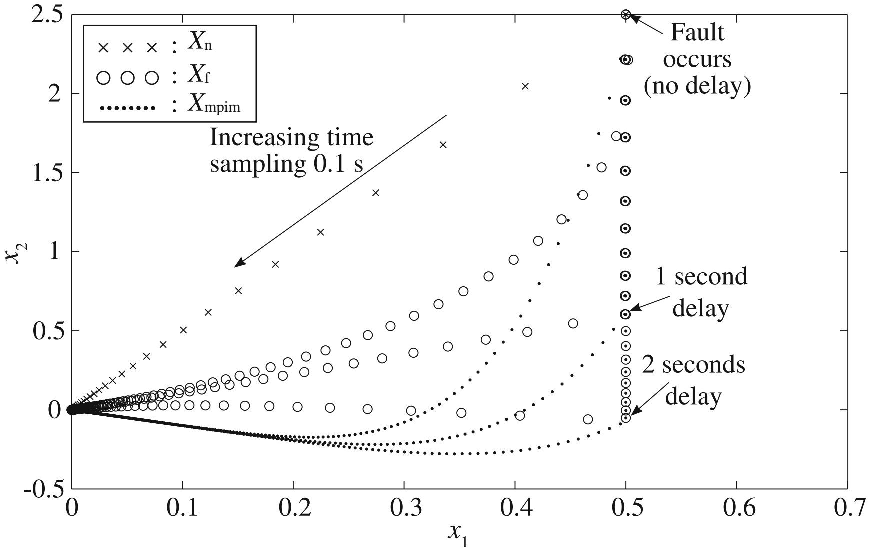


Fig. 9.40 Nominal, MPIM and NTT state trajectories

Assume a 2 s accommodation delay, which is composed of 1 s for the detection, isolation and diagnosis procedure, which ends with an estimate of matrix $\boldsymbol{B}_f$, and 1 s for control accommodation that solves Eqs. (9.115) and (9.116) based on this estimate. Let us now illustrate how much the efficiency of the fault accommodation scheme is improved by using the Progressive Accommodation approach. Solving the Riccati equation takes five Newton-Ralphson iterations, according to the following sequence:

$$\boldsymbol{H}_1 = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \boldsymbol{H}_2 = \begin{pmatrix} -0.7 & 0.2 \\ 0.2 & -0.7 \end{pmatrix}$$

$$\boldsymbol{H}_3 = \begin{pmatrix} -0.4839 & -0.0161 \\ -0.0161 & -0.4839 \end{pmatrix}, \quad \boldsymbol{H}_4 = \begin{pmatrix} -0.4357 & -0.0643 \\ -0.0643 & -0.4357 \end{pmatrix}$$

$$\boldsymbol{H}_5 = \begin{pmatrix} -0.4330 & -0.0670 \\ -0.0670 & -0.4330 \end{pmatrix}.$$

For this system, using the first iteration value $\boldsymbol{H}_1$ (which is obtained after 0, 2 s) instead of waiting the Riccati equation solution $\boldsymbol{H}_5$ for 1 s, allows to stabilise the system much sooner, and hence gives improved results.

Figure 9.41 compares the direct accommodation control and the progressive accommodation one. It is seen that progressive accommodation practically *rubs out* the effect of the accommodation delay: the resulting trajectories are quite close to the ones associated with a 1 s delay. □

9.6 Exercises

Exercise 9.1 Reconfiguration by model-matching techniques
Consider a stable plant

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} -\frac{1}{T_1} & 0 \\ \frac{1}{T_2} & -\frac{1}{T_2} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}$$

$$y(t) = (1 \quad 1) \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

and the stabilising proportional controller

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} -k_1 \\ 0 \end{pmatrix} y(t).$$

If the actuator 1 fails, the control loop should be closed with the help of the redundant control input $u_2(t)$. Does the model-matching approach yield a stable closed-loop system? Is the performance of the closed-loop system improved with respect to the nominal loop if the Markov parameter approach is used? □

Exercise 9.2 Fault-tolerant control of the three-tank system
Consider the three-tank system introduced in Sect. 2.2, where in the first part of the exercise the redundant hardware is switched off. Use a continuous PI-controller for the level $h_1(t)$ of the left tank and assume that the sensor used in this control loop fails. What is the result of the model-matching approach to this problem if the level $h_2(t)$ of the second tank is continuously measured and used for the controller of the left tank?

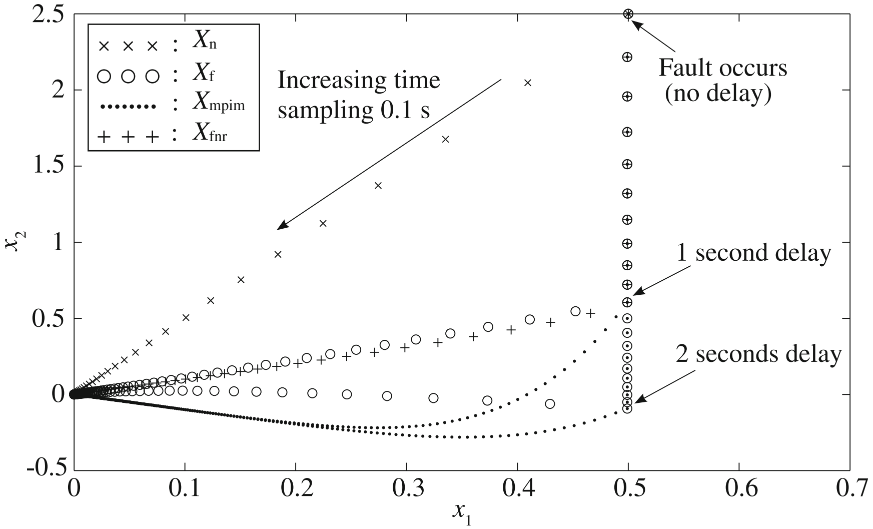


Fig. 9.41 Progressive accommodation in the nominal trajectory tracking state trajectories

Consider now a continuous level controller for Tank T_2 and assume that the actuator V_{12H} fails. The level $h_2(t)$ should be controlled by switching on the Tank T_3 and using the set-point $h_{3\text{ref}}(t)$ of the level controller of this tank as the control input to bring the level $h_2(t)$ of Tank T_2 towards the setpoint $h_{2\text{ref}}(t)$. Apply the existence conditions for model-matching and the virtual actuator to this problem. How can the failure of the control loop in Tank T_2 be compensated? □

Exercise 9.3 Virtual actuator
For the unstable system

$$\begin{aligned}\begin{pmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{pmatrix} &= \begin{pmatrix}0 & 1\\ 1 & -2\end{pmatrix}\begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix} + \begin{pmatrix}2 & 0\\ 1 & 1\end{pmatrix}\begin{pmatrix}u_1(t)\\ u_2(t)\end{pmatrix}\\ y(t) &= (2 \quad 1)\begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix}\end{aligned}$$

a proportional controller

$$\begin{pmatrix}u_1(t)\\ u_2(t)\end{pmatrix} = -\begin{pmatrix}0\\ k_2\end{pmatrix} y(t).$$

should be found that stabilises the system. In case of the actuator failure a virtual actuator should be used to stabilise the system with the nominal controller. Find reasonable parameters of the virtual actuator and prove that the reconfigured closed-loop system is stable. □

Exercise 9.4 Nominal and model-matching control for a single-axis satellite
This exercise is a continuation of Exercises 5.3, 6.3 and 6.4. The objective is to perform attitude control for a single axis of a satellite. In this exercise, actuator dynamics need be considered and two states x_3 and x_4 have been added to describe actuator dynamics.

A state-space model for the nominal plant is given by:

$$\begin{aligned}\dot{x}_1(t) &= I^{-1}(x_3(t) + x_4(t) + d(t))\\ \dot{x}_2(t) &= x_1(t)\\ \tau_1\dot{x}_3(t) &= -x_3(t) + b_1 u_1(t)\\ \tau_2\dot{x}_4(t) &= -x_4(t) + b_3 u_2(t)\\ y_1(t) &= x_1(t) + w_1(t)\\ y_2(t) &= x_2(t) + w_2(t)\\ y_3(t) &= x_3(t) + w_3(t).\end{aligned}$$

The system has actuator 1 (the input to which is $u_1(t)$) as the primary actuator. A second actuator (with input $u_2(t)$) is intended for secondary actuation should the primary one fail. The secondary actuator has a time constant that is larger than that of actuator 1.

Parameters. The nominal values of parameters are

$$\begin{aligned}\tau_1 &= 0.5\text{ s}\\ \tau_2 &= 2.5\text{ s}\\ b_1 &= 1.0\\ b_2 &= 1.0\end{aligned}$$

1. Design a nominal controller that use $sy_1(t)$ and $y_2(t)$ as measurements and $u_1(t)$ as actuator input,

$$u_1(t) = -(l_1 y_1(t) + l_2 y_2(t)). \tag{9.117}$$

The closed-loop system should have two real eigenvalues at $s = -0.5 \frac{\mathrm{rad}}{\mathrm{s}}$. Disregard the actuator that is not in use.

A fault on actuator 1 renders this actuator useless and it is needed to use actuator 2 instead.

2. Investigate whether ideal model matching is possible with this form of controller when this fault happens.
3. Design a dynamical controller that will provide model matching in the frequency domain. □

9.7 Bibliographical Notes

Fault-tolerant model-matching design. [151] is one of the earliest papers on controller reconfiguration by model-matching. [117] describes an improvement of the pseudo-inverse method for the ensurance of stability. A survey of the methods are given in [216]. A proof of Lemma 9.4 can be found in [412].

Further extensions of the pseudo-inverse method that result in the *admissible model-matching approach* have been recently given in [322], with extensions to the linear quadratic problem in [40, 328] and aerospace applications in [39].

Control reconfiguration for actuator or sensor failures. The ideas of the virtual sensor and virtual actuator have been developed in [220]. A thorough treatment can be found in the monographs [288, 340]. These concepts have been experimentally tested at a laboratory process [289], a two-degrees-of-freedom helicopter model [221] and a fuel cell [282]. The generalised version of the virtual actuator, which is explained in this chapter, has been proposed in [213]. Design methods for virtual sensors and virtual actuators can be found in [216, 290, 294, 310, 311] with extensions to nonlinear systems in [291, 293]. The conceptual similarities and differences of the virtual actuator and the dual observer are described in [292]. Alternative method that likewise use the fault-hiding principle are described, for example, in [202].

Fault-tolerant $\boldsymbol{H_\infty}$ design. Controller redesign based on the Youla-Kucera parametrisation is described in [188, 251, 359]. In [248], the Youla-Kucera parametrisation has been applied in connection with tuning controllers. The exact, the almost exact and the optimal design problems for $\boldsymbol{Q}$ have been considered in detail in [296].

The results in Sect. 9.4 are based on [344, 345] which focus on the use of fault estimation within a reliable control framework [375]. The methods for reconfiguration design are new within the fault-tolerant control domain. A few schemes have come into real application. Predetermined design for accommodation was demonstrated for a satellite in [42, 43].

The design methods considered in Sect. 9.5 are based on the same conditions as the methods described in [317]. Further results on using the Youla-Kucera parameterisation for fault-tolerant control in the additive fault, multiplicative fault and parameter fault cases can be found in [243, 247, 250, 251, 346]. An architecture for fault-tolerant control, based on joint controller and FDI design was presented in [241].

Theorem 9.6 has been proved in [412].

Handling of the fault recovery transients. The link between mastering the transient of controller switching and handling actuator saturation has been recognised for a long time. Indeed, both problems involve the discrepancy between the controller output and the process input, which might lead to performance degradation and even instability of the closed-loop. Anti-windup methods have thus been developed with a view to handle both problems (see [9] for the observer-based approach, and [140, 141, 276] for the conditioning technique). In [134, 406], they are explicitly introduced in multi-controller schemes such as found in hybrid or switched-mode systems in order to avoid undesirable switching transients. The anti-windup mechanisms used in [134] are high-gain feedback loops around each idle controller, which force the controller outputs to track the process input, while in [406] each controller is augmented with dynamics identical to that of the plant in order to allow the controller state to evolve in an appropriate way when the controller is not connected to the plant input. The latter scheme is cumbersome when the number of controllers is large.

Dedicated methods have also been studied for handling transients in controller switching. In [133], the authors recast the problem in an associated tracking problem, where the standby controller is viewed as a dynamical system of which the output should track the manipulated signal (plant input) by means of a two-degree-of-freedom controller. In [397], a simple new realisation of a set of linear SISO controllers is described that inherently assures bumpless transient upon switching between controllers.

Reference [396] made a rigorous analysis of controller switching and suggested an adaptive method to obtain bounded signals with a nominal controller from the instant a fault is detected until controller reconfiguration is made.

Progressive accommodation was first introduced in [402] to handle aircraft actuator faults, and the general approach was presented in [403]. In [66] “anytime algorithms” are used as an interesting tool to address fault recovery transients, since they produce solutions that are improving as the number of iterations increases, while any current solution can be applied before complete convergence is achieved.

In [334] the fault-tolerant linear quadratic problem is extended to the trajectory tracking problem which arises when a pre-designed system trajectory is to be followed as closely as possible (for example in space rendez-vous missions) instead of recomputing an optimal trajectory associated with the current configuration.