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

\[ u(t) = k(y(t), y_{ref}(t)) \]

is used to attenuate the disturbance \( d \) and to ensure command tracking with respect to the command input \( y_{ref} \). The control law \( k \) 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.
9.1 Fault-Tolerant Model-Matching Design

9.1.1 Reconfiguration Problem

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 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 $y$ to be controlled and the input vector $u$ to be used,
- the control law $k$ including the controller parameters,
- the set-point $y_{\text{ref}}$ 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).
9.1 Fault-Tolerant Model-Matching Design

9.1.1 Reconfiguration Problem

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

\[
\dot{x}(t) = g(x(t), u(t), f), \quad x(0) = x_0 \quad \text{(9.1)}
\]
\[
y(t) = h(x(t), u(t), f) \quad \text{(9.2)}
\]

with state \( x \in \mathbb{R}^n \), input \( u \in \mathbb{R}^m \) and output \( y \in \mathbb{R}^r \) 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 \( k \)
Control specifications
Fault \( f \)

**Find:** Control configuration and new control law \( k_f \).
9.1 Fault-Tolerant Model-Matching Design

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

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \quad (9.3) \\
y(t) &= Cx(t) \quad (9.4)
\end{align*}
\]

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

\[
\begin{align*}
\dot{x}(t) &= (A - BK)x(t) \\
y(t) &= Cx(t).
\end{align*}
\]

If the controller does not use all the inputs \( u_i \) of the input vector \( u \), the matrix \( K \) has zero rows, which is typical for plants with redundant actuators.

When the fault \( f \) occurs, the faulty plant is given by

\[
\begin{align*}
\dot{x}(t) &= A_fx(t) + B_fu(t) \quad (9.5) \\
y(t) &= C_fx(t). \quad (9.6)
\end{align*}
\]
9.1 Fault-Tolerant Model-Matching Design

9.1.2 Pseudo-Inverse Method

where the fault $f$ has changed the system properties, which are now described by the matrices $A_f$, $B_f$ and $C_f$. If the sets of available input or output signals have changed, the matrices $B_f$ and $C_f$ have vanishing columns or rows, respectively.

A new state feedback controller

$$u(t) = -K_f x(t)$$

should be found such that the closed-loop system

$$\dot{x}(t) = (A_f - B_f K_f) x(t)$$

$$y(t) = C_f x(t)$$

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

$$A - BK = A_f - B_f K_f,$$

(9.7)

which means that both closed-loop systems have similar dynamics.
9.1 Fault-Tolerant Model-Matching Design

9.1.2 Pseudo-Inverse Method

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

$$\|(A - BK) - (A_f - B_f K_f)\|.$$  \hspace{1cm} (9.8)

The solution to this problem is given by

$$K_f = B_f^+ (A_f - A + BK) = \left(B_f^T B_f\right)^{-1} B_f^T (A_f - A + BK).$$  \hspace{1cm} (9.9)
9.1 Fault-Tolerant Model-Matching Design

9.1.2 Pseudo-Inverse Method

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 $K_f$ 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 $A_f - BK_f$ and the performance of the control loop have to be evaluated separately.

Extensions of this method ensure the stability without a separate test.

9.1 Fault-Tolerant Model-Matching Design

9.1.3 Model-Matching Control for Sensor Failures

This section considers the case of complete sensor failures. If the $i$th sensor fails, the output $y_i$ is set to zero. In the plant model the matrix $C$ changes to $C_f$, whose $i$th 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

$$u(t) = -K y(t)$$

for which the nominal closed-loop system is described by

$$\dot{x}(t) = (A - BK) x(t)$$
$$y(t) = C x(t).$$
9.1 Fault-Tolerant Model-Matching Design

9.1.3 Model-Matching Control for Sensor Failures

For the faulty plant, the new controller

\[ u(t) = -K_f y_f(t) \]

should be found such that the closed loop

\[ \dot{x}(t) = (A - B K_f C_f) x(t) \]
\[ y_f(t) = C_f x(t) \]

has the same dynamics as the nominal loop.

---

9.1 Fault-Tolerant Model-Matching Design

9.1.3 Model-Matching Control for Sensor Failures

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

\[ K_f C_f = K C. \quad (9.10) \]

To find an appropriate matrix \( K_f \) is possible only if the condition

\[ \text{Kern}(C_f) \subseteq \text{Kern}(C) \quad (9.11) \]

The condition means that the measurement information obtained by the full output vector \( y \) is the same as the information obtained by the remaining sensors through \( y_f \).

The kernel of \( C \) is the set of vectors \( x \) for which \( C x = 0 \) holds.
9.1 Fault-Tolerant Model-Matching Design

9.1.3 Model-Matching Control for Sensor Failures

The condition (9.11) can be written in an equivalent form as

\[ \text{rank } C_f = \text{rank } \begin{pmatrix} C \\ C_f \end{pmatrix} \]

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

\[ u(t) = -K P y(t) \]  \hspace{1cm} (9.12)

solves the reconfiguration problem where

\[ P = C C_f^+ = C C_f^T (C_f C_f^T)^{-1} \]  \hspace{1cm} (9.13)

satisfies the relation

\[ C = PC_f. \]  \hspace{1cm} (9.14)

---

**Situation 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, \( y_f = a y \) holds for some scalar \( a \).
- A sensors is at fault which has at least one parallel redundant sensor. The matrix \( P \) 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 \( P \).

The later two cases are only possible if \( C \) does not have full rank, which is likely in special applications only.
9.1 Fault-Tolerant Model-Matching Design

9.1.4 Model-Matching Control for Actuator Failures

In case of an actuator failure, the matrix $B$ is replaced by the matrix $B_f$ with zero column for the failing actuator. The output feedback

$$u(t) = -Ky(t)$$

which leads to the closed-loop system

$$\dot{x}(t) = (A - B_f K) x(t)$$
$$y(t) = Cx(t).$$

should be replaced by a new controller

$$u_f(t) = -K_f y(t)$$

such that the closed loop

$$\dot{x}(t) = (A - B_f K_f C) x(t)$$
$$y(t) = Cx(t)$$

has the same dynamics as the nominal loop.

A solution $K_f$ to this equation exists only if the condition

$$\text{Im} \ (B_f) \supseteq \text{Im} \ (B)$$

holds, where $\text{Im}$ denotes the image of a matrix. An equivalent formulation of the condition (9.16) is given by

$$\text{rank } B_f = \text{rank } (B \ B_f)$$

The image of $C$ is the set of vectors $y$, for which a vector $x$ exists such that $y = Cx$ holds.
9.1 Fault-Tolerant Model-Matching Design

9.1.4 Model-Matching Control for Actuator Failures

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

\[ u(t) = -NKy(t), \]  

(9.17)

where

\[ N = B_f^+B = \left( B_f^TB_f \right)^{-1}B_f^TB \]  

(9.18)

is a matrix satisfying the relation

\[ B_fN = B. \]  

(9.19)

The new controller \( K_f = NK \) 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.

Fig. 9.3 Example demonstrating the model-matching reconfiguration strategy
9.1 Fault-Tolerant Model-Matching Design

Example 9.1 Model-matching for actuator failures
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

\[ B = (b \ kb) \]

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:

\[ u_1(t) = u_C(t) = -Ky(t) \]

for some controller \( K \) and some output \( y \) of the tank system.
If the corresponding actuator fails, the controller should be switched to the second input, where

\[ B_f = (0 \ kb) \]

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

\[ N_{21} = (k^2b^Tb)^{-1}kb^Tb = \frac{1}{k}, \]

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

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

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

\[ B = (b \ kb) \]

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:

\[ u_1(t) = u_C(t) = -Ky(t) \]

for some controller \( K \) and some output \( y \) of the tank system.
If the corresponding actuator fails, the controller should be switched to the second input, where

\[ B_f = (0 \ kb) \]

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

\[ N_{21} = (k^2b^Tb)^{-1}kb^Tb = \frac{1}{k}, \]

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

\[ u_2(t) = \frac{1}{k}u_C(t) \]
9.1 Fault-Tolerant Model-Matching Design

**Example 9.1 Model-matching for actuator failures**

This is an obvious solution: As the gain of the new actuator is $k$-times the gain of the old one, the old input $u_c(t)$ 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

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

such that the model now reads as

$$\dot{x}(t) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b & k b & A \end{pmatrix} \dot{x}(t) + \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}$$

$$y(t) = \begin{pmatrix} O & O & C \end{pmatrix} \dot{x}(t)$$
9.1 Fault-Tolerant Model-Matching Design

Example 9.1 Model-matching for actuator failures

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

\[ \mathbf{B} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{B}_f = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \]

the solution

\[ \mathbf{N}_{21} = \left( \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}^T \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \mathbf{0} \]

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 \( \mathbf{B} \mathbf{u} \) of the nominal controller by the reconfigured controller \( \mathbf{B}_f \mathbf{N} \mathbf{u} \). This is impossible in this example, because the nominal controller has a direct effect only on the state variable \( x_{a1} \) and no effect at all on the state variable \( x_{a2} \) whereas the redundant input leads to the reverse situation. Hence, no choice of \( \mathbf{N} \) 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 Fault-Tolerant Model-Matching Design

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 of concentrates on the forcing action at point P 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 Q in the figure. Such redundancies are hidden from a forcing action perspective.

Fig. 9.4 Input/output-based reconfiguration after actuator failures
9.1 Fault-Tolerant Model-Matching Design

9.1.5 Markov Parameter Approach to Control Reconfiguration for Actuator Failures

The Markov parameters

\[ G_i = CA^{i-1}B, \quad i = 1, \ldots, n \]  

(9.20)

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

\[ P(s) = \sum_{i=0}^{\infty} G_i s^{-i}. \]  

(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

\[ u_c(t) = Nu_f(t). \]  

(9.22)

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

\[ S_O = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} \in \mathbb{R}^{n \times n} \]  

(9.23)
9.1 Fault-Tolerant Model-Matching Design

9.1.5 Markov Parameter Approach to Control Reconfiguration for Actuator Failures

The design problem to Markov parameter recovery can be posed as

\[ N = \arg \min_N \| S_O B_f N - S_O B \| \]  \hspace{1cm} (9.24)

with the solution

\[ N = (S_O B_f)^+ S_O B. \] \hspace{1cm} (9.25)

If the condition

\[ \text{Im} (S_O B_f) \supseteq \text{Im} (S_O B) \] \hspace{1cm} (9.26)

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

\[ \text{rank} (S_O B_f) = \text{rank} (S_O B_f \, S_O B). \] \hspace{1cm} (9.27)

This condition is equivalent to

\[ \text{rank} (S_O B_f) = \text{rank} (S_O B_f \, S_O B). \] \hspace{1cm} (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.
9.1 Fault-Tolerant Model-Matching Design

9.1.5 Markov Parameter Approach to Control Reconfiguration for Actuator Failures

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

\[ u(t) = -NKy(t), \tag{9.28} \]

Where

\[ N = (S_O B_f)^+ S_O B \tag{9.29} \]

is a matrix satisfying the relation

\[ CA^{(i-1)} B_f N = CA^{(i-1)} B, \quad i = 1, \ldots, n. \tag{9.30} \]

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

---

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

\[ S_O B = (\gamma \ k\gamma)b \quad \text{with} \quad \gamma = (C \ CA\ldots)^T, \tag{9.31} \]

whereas after the fault the relation

\[ S_O B_f = (0 \ k\gamma)b \tag{9.32} \]

holds.
9.1 Fault-Tolerant Model-Matching Design

9.1.5 Markov Parameter Approach to Control Reconfiguration for Actuator Failures

Example 9.1 (cont.) Model-matching for actuator failures: Markov approach

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

\[ S_{O} B_{f} N = S_{O} B \quad (9.33) \]

is found using Eq. (9.25) as

\[ N = \begin{pmatrix} 0 & 0 \\ \frac{1}{k} & 1 \end{pmatrix}. \quad (9.34) \]

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

\[ u_{p} \]

\[ T_{1} \]

\[ T_{2} \]

\[ u_{U} \]

\[ u_{L} \]

\[ h_{1} \]

\[ h_{2} \]

\[ h_{01} \]

\[ h_{02} \]

Fig. 9.5 Reconfiguration of a two-tank system
9.1 Fault-Tolerant Model-Matching Design

9.1.5 Markov Parameter Approach to Control Reconfiguration for Actuator Failures

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 $x = (v_L, v_H, h_1, h_2)^T$, the tank system is described by the linear model (9.3), (9.4) with

\[
A = 10^3 \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & -3.2 & -3.4 \\
-3.2 & 3.4 & 7.1 & 3.6 \\
3.2 & 3.4 & 7.1 & -18
\end{pmatrix},
\]

\[
B = 10^3 \begin{pmatrix}
0 & 10^{-3} & 0 \\
0 & 0 & 10^{-3} \\
8.1 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix},
\]

\[
B_f = 10^3 \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 10^{-3} & 0 \\
8.1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix},
\]

\[
C = \begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}.
\]
9.1 Fault-Tolerant Model-Matching Design

9.1.5 Markov Parameter Approach to Control Reconfiguration for Actuator Failures

Example 9.2 Markov parameter approach applied to the two-tank example

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

$$N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.9167 & 1 \end{pmatrix}.$$
9.1 Fault-Tolerant Model-Matching Design

Example 9.2 Markov parameter approach applied to the two-tank example

![Graph of Level in T2 and Valve Control]

*Fig. 9.6* Experimental results with the reconfigured tank system: After the failure of the lower valve ($u_1$, 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

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 $u_c$, the same (or approximately the same) output $y_c$ 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.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.1 The Idea of Virtual Sensors and Virtual Actuators

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 a “virtual sensor”, because it reconstructs that element $y_i$ of the output vector $y_c$ 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.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.1 The Idea of Virtual Sensors and Virtual Actuators

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.

![Diagram of two coupled tanks](image)

Fig. 9.8 Reconfiguration problem for the tank example

9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.1 The Idea of Virtual Sensors and Virtual Actuators

Example 9.3 Two-tank reconfiguration problem

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 $y_{1\text{ref}}$ and $y_{2\text{ref}}$. 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 $y_{2\text{ref}}$.

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.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.1 The Idea of Virtual Sensors and Virtual Actuators

Example 9.3 Two-tank 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).

---

**Fig. 9.9** Block diagram of the reconfiguration problem
9.2 Control Reconfiguration for Actuator or Sensor Failures

Example 9.3 Two-tank reconfiguration problem

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.

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:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ed(t), \quad x(0) = x_0 \quad (9.35)$$

$$y(t) = Cx(t). \quad (9.36)$$

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, $B$ and $C$ may not have full rank.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.2 Reconfiguration Problem

The nominal process is stabilised by a nominal controller with output $u(t)$ and inputs $y(t)$ and $y_{ref}(t)$. 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 as (9.37) is used.

$$u_c(t) = -K y_c(t) + V y_{ref}(t) \quad (9.37)$$

For nominal system:

$$u(t) = u_c(t)$$
$$y(t) = y_c(t)$$

This control loop is assumed to be stable and to satisfy the performance requirements concerning set-point tracking and disturbance rejection.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.2 Reconfiguration Problem

Fault cases.

- In the case that the fault $f$ indicates a loss of sensor $i$, the $i^{th}$ row of the matrix $C$ is changed into zeros and the new matrix is denoted by $C_f$.
- If the $j^{th}$ actuator fails, the $j^{th}$ column of the matrix $B$ is set to zero and the resulting matrix denoted by $B_f$.
- 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 stabilizing controller exists. The input and the output of the faulty plant are denoted by $u_f$ or $y_f$, 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 $y_f(t) = y(t)$ should hold for any $d(t)$, $y_{ref}(t)$ and $x_0$.

It will be demonstrated that this strong goal is only feasible in very special cases.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.2 Reconfiguration Problem

Weak reconfiguration goal:
The weak goal consists of a static and a dynamical part. Considering the static behaviour, the output \(y_f\) of the reconfigured loop should have the same value as for the nominal system. This means that for constant values of \(y_{\text{ref}}\) and \(d\), the following relation should hold:

\[
y_f(t) \to y(t) \quad \text{for} \quad t \to \infty
\]

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 \(i\)th 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

\[
\dot{x}_f(t) = Ax_f(t) + Bu_f(t) + Ed(t), \quad x_f(0) = x_{f0} \tag{9.40}
\]

\[
y_f(t) = C_f x_f(t), \tag{9.41}
\]
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Sensor

If condition (9.11) satisfied, then the reconfigured controller (9.12) can be used.

If condition (9.11) 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

\[
\begin{align*}
\dot{x}_V(t) &= A_V x_V(t) + B_V u_c(t) + L y_f(t), \quad x_V(0) = x_{V0} \\
u_f(t) &= u_c(t) \\
y_c(t) &= C_V x_V(t) + P y_f(t)
\end{align*}
\]

(9.42) (9.43) (9.44)

with the state \(x_V \in \mathbb{R}^n\) and with matrices

\[
\begin{align*}
A_V &= A - LC_f \\
B_V &= B \\
C_V &= C - PC_f.
\end{align*}
\]

(9.45) (9.46) (9.47)

\(P\) and \(L\) denote matrices that can be freely chosen.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Sensor

The main part of the virtual sensor is the state observer with the state vector $x_V(t)$.

The complete output $y_c(t)$ of the plant can be approximately determined: $y_c(t) \approx Cx_V(t)$. 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 $P$ is a design parameter.

- For $P = 0$ only observed values are used.

$$\frac{\dot{x}_f(t)}{\dot{x}_V(t)} = \begin{pmatrix} A & O \\ LC_f A - LC_f & 0 \end{pmatrix} \begin{pmatrix} x_f(t) \\ x_V(t) \end{pmatrix} + \begin{pmatrix} B \\ B \end{pmatrix} u_c(t) + \begin{pmatrix} E \\ O \end{pmatrix} d(t) \quad (9.48)$$

$$y_c(t) = \begin{pmatrix} PC_f C_V \end{pmatrix} \begin{pmatrix} x_f(t) \\ x_V(t) \end{pmatrix} \quad (9.49)$$

A state transformation is performed in order to introduce the observation error $x_{\Delta}(t) = x_V(t) - x_f(t)$: Eqs. (9.48), (9.49) are equivalent to

$$\frac{\dot{x}_f(t)}{\dot{x}_{\Delta}(t)} = \begin{pmatrix} A & O \\ O A - LC_f & 0 \end{pmatrix} \begin{pmatrix} x_f(t) \\ x_{\Delta}(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u_c(t) + \begin{pmatrix} E \\ -E \end{pmatrix} d(t) \quad (9.50)$$

$$y_c(t) = \begin{pmatrix} C \\ C_V \end{pmatrix} \begin{pmatrix} x_f(t) \\ x_{\Delta}(t) \end{pmatrix} \quad (9.51)$$

$$\begin{pmatrix} x_f(0) \\ x_{\Delta}(0) \end{pmatrix} = \begin{pmatrix} x_f(0) \\ x_V(0) - x_f(0) \end{pmatrix}.$$
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Sensor

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

\[
\begin{align*}
\begin{pmatrix}
\dot{x}_f(t) \\
\dot{x}_\Delta(t)
\end{pmatrix} &= 
\begin{pmatrix}
A - BK_C & -BK_C \h\nO & A - LC_f
\end{pmatrix}
\begin{pmatrix}
x_f(t) \\
x_\Delta(t)
\end{pmatrix} + \\
&+ \begin{pmatrix}
E \\
-E
\end{pmatrix} d(t) + \begin{pmatrix}
BV \\
O
\end{pmatrix} y_{ref}(t)
\end{align*}
\]

(9.52)

\[
y_f(t) = \begin{pmatrix}
C_f \\
O
\end{pmatrix} \begin{pmatrix}
x_f(t) \\
x_\Delta(t)
\end{pmatrix}
\]

(9.53)

The trajectory of this system depends on the initial state, the reference input \( y_{ref} \) and the disturbance \( d \) (Fig. 9.10). As the system is linear, the behaviour can be analysed separately for these three excitations.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Sensor

**Autonomous behaviour.** For $y_{ref}(t) = 0$ and $d(t) = 0$ the system (9.52), (9.53) simplifies to

$$\begin{bmatrix}
\dot{x}_f(t) \\
\dot{x}_\Delta(t)
\end{bmatrix} =
\begin{bmatrix}
A - BK_C & -BK_CV \\
O & A - LC_f
\end{bmatrix}
\begin{bmatrix}
x_f(t) \\
x_\Delta(t)
\end{bmatrix}$$

(9.54)

$$y_f(t) = (C_f O) \begin{bmatrix} x_f(t) \\ x_\Delta(t) \end{bmatrix}$$

(9.55)

$$\begin{bmatrix} x_f(0) \\ x_\Delta(0) \end{bmatrix} = \begin{bmatrix} x_{f0} \\ x_{\Delta0} - x_{f0} \end{bmatrix}.$$

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

---

9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Sensor

**Theorem 9.1** (Separation principle for the virtual sensor) The set $\sigma$ 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):

$$\sigma = \sigma\{A - BK_C\} \cup \sigma\{A - LC_f\}.$$

The stability of the closed-loop is guaranteed if the nominal control loop is stable (depending on $K$) and if the observer is stable (depending on $L$). The second condition can be satisfied by an appropriate choice of $L$ because the pair $(A, C_f)$ is assumed to be observable.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Sensor

**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:

\[
\begin{pmatrix}
\dot{x}_f(t) \\
\dot{x}_\Delta(t)
\end{pmatrix}
= \begin{pmatrix}
A - BK C - BK C_V \\
0
\end{pmatrix}
\begin{pmatrix}
x_f(t) \\
x_\Delta(t)
\end{pmatrix} + \begin{pmatrix}
E \\
-E
\end{pmatrix} d(t)
\]

\[
y_f(t) = \begin{pmatrix}
C_f & O
\end{pmatrix}
\begin{pmatrix}
x_f(t) \\
x_\Delta(t)
\end{pmatrix}
\]

\[
\begin{pmatrix}
x_f(0) \\
x_\Delta(0)
\end{pmatrix} = \begin{pmatrix}
O \\
O
\end{pmatrix}
\]

It is obvious that the output \( y_f \) is different from the output \( y \) 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 \( K \) and \( L \), while the zeros are affected by \( P \).

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.*
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Sensor

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

\[
\begin{align*}
\dot{x}_f(t) &= A x_f(t) + B_f u_f(t) + E d(t), \quad x_f(0) = x_{f0} \quad (9.56) \\
y_f(t) &= C x_f(t), \quad (9.57)
\end{align*}
\]

where zero columns in the matrix \( B_f \) reflect the failing actuators. If the condition (9.16) is satisfied, the static reconfiguration block

\[
\begin{align*}
&u_f(t) = N u_c(t) \\
y_c(t) = y_f(t)
\end{align*}
\]

can be used. In the following, the more general case is investigated, where this condition is not satisfied.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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.

\[
\begin{align*}
\dot{x}_\Delta(t) &= A_\Delta x_\Delta(t) + B_\Delta u_c(t), \quad x_\Delta(0) = x_{\Delta 0} \quad (9.58) \\
u_f(t) &= C_\Delta x_\Delta(t) + D_\Delta u_c(t) \quad (9.59) \\
y_c(t) &= C x_\Delta(t) + y_f(t) \quad (9.60)
\end{align*}
\]

with the state \(x_\Delta \in \mathbb{R}^n\) and the matrices

\[
\begin{align*}
A_\Delta &= A - B_f M \\ B_\Delta &= B - B_f N \\ C_\Delta &= M \\ D_\Delta &= N. 
\end{align*}
\]

\(M\) and \(N\) denote matrices that can be freely chosen.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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

\[
\begin{align*}
\begin{pmatrix}
\dot{x}_f(t) \\
\dot{x}_\Delta(t)
\end{pmatrix} &= \begin{pmatrix}
A & B_f M \\
O & A - B_f M
\end{pmatrix} \begin{pmatrix}
x_f(t) \\
x_\Delta(t)
\end{pmatrix} \\
&\quad + \begin{pmatrix}
B_f N \\
B - B_f N
\end{pmatrix} u_c(t) + \begin{pmatrix}
E \\
O
\end{pmatrix} d(t) \\
y_c(t) &= \begin{pmatrix}
C \\
C
\end{pmatrix} \begin{pmatrix}
x_f(t) \\
x_\Delta(t)
\end{pmatrix} .
\end{align*}
\]
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

The introduction of the new state \( \hat{x}(t) = x_f(t) + x_\Delta(t) \) leads to the following equivalent model:

\[
\frac{d}{dt} \begin{pmatrix} \hat{x}(t) \\ x_\Delta(t) \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & A - B_f M \end{pmatrix} \begin{pmatrix} \hat{x}(t) \\ x_\Delta(t) \end{pmatrix} + \begin{pmatrix} B \\ B - B_f N \end{pmatrix} u_c(t) + \begin{pmatrix} E \\ O \end{pmatrix} d(t)
\]

\[
y_c(t) = \begin{pmatrix} C \\ O \end{pmatrix} \begin{pmatrix} \hat{x}(t) \\ x_\Delta(t) \end{pmatrix}
\]

\[
\begin{pmatrix} \hat{x}(0) \\ x_\Delta(0) \end{pmatrix} = \begin{pmatrix} x_0 + x_{\Delta 0} \\ x_{\Delta 0} \end{pmatrix}.
\]

Note that the state \( x_\Delta \) of the second subsystem is not observable by \( y_c \). Hence, this state does not influence the I/O-behaviour of the reconfigured plant, whose model can be reduced to

\[
\dot{x}(t) = Ax(t) + Bu_c(t), \quad x(0) = x_0 + x_{\Delta 0}
\]

\[
y_c(t) = Cx(t).
\]

This model is identical to the nominal plant provided that \( x_{\Delta 0} = 0 \) holds.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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 $M$ and $N$ of the virtual actuator.

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

9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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 $d$ and command input $y_{\text{ref}}$. If the transformed model is used, the reconfigured closed-loop system is described by

$$
\frac{d}{dt} \begin{pmatrix}
\dot{x}(t) \\
x_\Delta(t)
\end{pmatrix} = \begin{pmatrix}
A - BK C & O \\
-B_\Delta KC & A - B_f M
\end{pmatrix} \begin{pmatrix}
\dot{x}(t) \\
x_\Delta(t)
\end{pmatrix} + 
\begin{pmatrix}
x_0 + x_{\Delta 0} \\
x_{\Delta 0}
\end{pmatrix}.
$$

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 $\sigma$ 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):

$$
\sigma = \sigma\{A - BK C\} \cup \sigma\{A - B_f M\}.$$

9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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 $M$ and $N$ 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 $x'$ has been introduced as before:

\[
\begin{aligned}
\frac{d}{dt} \begin{pmatrix} \dot{x}(t) \\ x_\Delta(t) \end{pmatrix} &= \begin{pmatrix} A - BK & O \\ -B_\Delta KC & A - B_f M \end{pmatrix} \begin{pmatrix} x(t) \\ x_\Delta(t) \end{pmatrix} \\
&\quad + \begin{pmatrix} BV \\ B_\Delta V \end{pmatrix} y_{ref}(t) + \begin{pmatrix} E \\ O \end{pmatrix} d(t) \\
\begin{pmatrix} \dot{x}(0) \\ x_\Delta(0) \end{pmatrix} &= \begin{pmatrix} x_0 + x_{\Delta 0} \\ x_{\Delta 0} \end{pmatrix}
\end{aligned}
\]  

(9.67)

\[
y_c(t) = (C & O) \begin{pmatrix} \dot{x}(t) \\ x_\Delta(t) \end{pmatrix}
\]

(9.68)

\[
y_f(t) = (C & - C) \begin{pmatrix} \dot{x}(t) \\ x_\Delta(t) \end{pmatrix}
\]

(9.69)

9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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 $e = V y_{ref} - y_c$ is fed into the “difference system”

\[
\begin{aligned}
\dot{x}_\Delta(t) &= (A - B_f M)x_\Delta(t) + B_\Delta e(t), \quad x_\Delta(0) = x_{\Delta 0} \\
y_\Delta(t) &= C x_\Delta(t)
\end{aligned}
\]

(9.70)

(9.71)

whose name results from its output $y_\Delta$, which is the difference between the output $y_c$ of the nominal closed-loop system and the output $y_f$ of the reconfigured closed-loop system. Hence, $y_\Delta$ shows how the reconfigured closed-loop system differs from the nominal loop.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

This model yields two corollaries:

- The I/O-behaviour with respect to the disturbance input \( d \) or the command input \( y_{ref} \), respectively, and to the output \( y_c \) is identical to the corresponding I/O behaviour of the nominal closed-loop system.

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

**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 \( M \) and \( N \) of the virtual actuator in order to get a small difference \( y_d \) 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 \( N \) can be chosen such that the matrix \( B_d \) vanishes.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

**Corollary 9.2** If the matrix $N$ can be chosen such that

$$B_\Delta = B - B_f N = O$$  \hspace{1cm} (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 $d$ and the command input $y_{ref}$. Furthermore, if

$$x_\Delta(0) = 0$$  \hspace{1cm} (9.73)

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

---

9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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

$$u_f(t) = (B_f^T B_f)^{-1} B_f^T Bu_c(t)$$  \hspace{1cm} (9.74)

$$y_c(t) = y_f(t),$$  \hspace{1cm} (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 $B_\Delta \neq O$ holds and the dynamical part of the virtual actuator becomes active.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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

\[ G(s) = C(sI - A + B_f M)^{-1}(B - B_f N) = 0, \quad (9.76) \]

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

\[ \dot{x}_\Delta(t) = Ax_\Delta(t) + Bu_c(t) + B_f u_f(t), \quad x_\Delta(0) = x_\Delta 0 \quad (9.77) \]
\[ u_\Delta(t) = Mx_\Delta(t) + Nu_c(t) + Q\tilde{u}(t) \quad (9.78) \]

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

9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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

\[ G(0) = -C(A - B_f M)^{-1}(B - B_f N) = 0. \quad (9.79) \]

**Approximate solution.** The generalised virtual actuator has the property that the effect of the virtual actuator “disappears” if the matrix \( B_\Delta \) can be made very small by choosing the matrix \( N \) appropriately.

**Corollary 9.3** For \( \|B_\Delta\| \to 0 \), the behaviour of the reconfigured closed-loop system approaches that of the nominal loop:

\[ \|y_c(t) - y_f(t)\| \to 0. \]

Hence, if \( \|B_\Delta\| \) is sufficiently small it is reasonable to use the static reconfiguration block only.
9.2 Control Reconfiguration for Actuator or Sensor Failures

9.2.3 Virtual Actuator

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.

This model uses the following parameters:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Physical meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1, A_2$</td>
<td>Cross section areas of the two tanks</td>
</tr>
<tr>
<td>$Q_{1\text{max}}$</td>
<td>Maximum flow through the pump</td>
</tr>
<tr>
<td>$h_v$</td>
<td>Height of the upper pipe above the tank bottom</td>
</tr>
<tr>
<td>$S$</td>
<td>Constant of the valves</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravity constant</td>
</tr>
<tr>
<td>$k_I, k_P$</td>
<td>Controller parameters</td>
</tr>
</tbody>
</table>

The tank system is described by the nonlinear state-space model:

\[
\begin{align*}
\dot{h}_1(t) &= \frac{Q_{1\text{max}}}{A_1} (-k_I x_T(t) - k_P (h_1(t) - u_1(t))) \\
& \quad - \frac{Q_{1\text{max}}}{S} \sqrt{2g(h_1(t) - h_v)} u_2(t) - \frac{Q_{1\text{max}}}{S} \sqrt{2gh_1(t)} u_3(t) \\
\dot{x}_T(t) &= h_1(t) - u_1(t) \\
\dot{h}_2(t) &= \frac{1}{A_2} \left( S \sqrt{2g(h_1(t) - h_v)} u_2(t) + S \sqrt{2gh_1(t)} u_3(t) - S \sqrt{2gh_2(t)} d(t) \right) \\
y_c(t) &= h_2(t) 
\end{align*}
\]

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

\[
\begin{align*}
\dot{x}_T(t) &= h_1(t) - u_1(t) \\
\tilde{u}_1(t) &= -k_I x_T(t) - k_P (h_1(t) - u_1(t)).
\end{align*}
\]
9.2 Control Reconfiguration for Actuator or Sensor Failures

Example 9.4 Reconfiguration of the two-tank system

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

\[
A = \begin{pmatrix}
-0.0478 & -0.0004 & 0 \\
1.0000 & 0 & 0 \\
0.0058 & 0 & -0.0058 
\end{pmatrix}
\]

\[
B = \begin{pmatrix}
0.0406 & -0.0058 & -0.0092 \\
-1.0000 & 0 & 0 \\
0 & 0.0046 & 0.0073 
\end{pmatrix}
\]

\[
C = (0 \ 0 \ 1)
\]

\[
E = \begin{pmatrix}
0 \\
0 \\
-0.0454 
\end{pmatrix}
\]

\[
B_f = \begin{pmatrix}
0.0406 & 0 & -0.0092 \\
-1.0000 & 0 & 0 \\
0 & 0 & 0.0073 
\end{pmatrix}
\]

It 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 \( B \) has to be set to zero to obtain the matrix \( B_f \)