Chapter 2
Unknown Input Observers and Filters

As can be observed in the literature, observers (or filters in a stochastic framework) are commonly used in both control and fault diagnosis schemes of non-linear systems (see, e.g., [1–6] and the references therein). Undoubtedly, the most common approach is to use robust observers, such as the Unknown Input Observer (UIO) [2, 7], which can tolerate a degree of model uncertainty and hence increase the reliability of fault diagnosis. Although the origins of UIOs can be traced back to the early 1970s (cf. the seminal work of Ref. [8]), the problem of designing such observers is still of paramount importance both from the theoretical and practical viewpoints. A large amount of knowledge on using these techniques for model-based fault diagnosis has been accumulated through the literature for the last three decades (see Ref. [2] and the references therein). A large number of approaches to non-linear fault diagnosis and fault-tolerant control was published during the last two decades. For example, in Ref. [9] the high gain observer for Lipschitz systems was applied for the purpose of fault diagnosis. One of the standard methods of observer design consists in using a non-linear change of coordinates to turn the original system into a linear one (or a pseudo linear one). As indicated in the literature, such approaches can be applied for fault diagnosis and FTC [10, 11]. It should also be noted that when the feasibility condition regarding the non-linear change of coordinates is not matched, then the celebrated Extended Kalman Filter (EKF) can be applied in both stochastic and deterministic context (see, e.g., [2]).

Generally, design problems regarding the UIOs for non-linear systems can be divided into three distinct categories:

- *Non-linear state transformation-based techniques*: Apart from a relatively large class of systems for which they can be applied, even if the non-linear transformation is possible it leads to another non-linear system and hence the observer design problem remains open (see Ref. [7] and the references therein).
- *Linearisation-based techniques*: Such approaches are based on a similar strategy like that for the EKF [1]. In Ref. [2] the author proposed an extended unknown input observer for non-linear systems. He also proved that the proposed observer is convergent under certain conditions.

M. Witczak, *Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems*, Lecture Notes in Electrical Engineering 266, DOI: 10.1007/978-3-319-03014-2_2,

- *Observers for particular classes of non-linear systems*: For example UIOs for polynomial and bilinear systems or for Lipschitz systems [2, 12–14].

In the light of the above discussion, it is clear that accurate state estimation is extremely important for fault detection and control applications. However, estimation under noise and unknown inputs is very difficult.

In order to face the above-mentioned challenges, the design problems regarding UIOs (undertaken within the framework of this chapter) are divided into three distinct categories:

1. How to determine the unknown input distribution matrix, which will not decouple the effect of faults from the residual?
2. How to develop a possibly simple and reliable design procedure of UIO for both non-linear stochastic and deterministic systems?
3. How to extend the approach developed for the constant unknown input distribution matrix into a set of predefined unknown input distribution matrices?

Concerning the first question, a partial answer can be found in Ref. [15]. Indeed, the authors concentrate on the determination of the unknown input distribution matrix for linear systems but they do not answer the question when this matrix will cause the fault decoupling effect. Apart from the fact that there are approaches that can be used for designing UIOs for non-linear systems (listed above), the problem of determining the unknown input distribution matrix for this class of systems remains untouched. In other words, the authors assume that this matrix is known, which apart from relatively simple cases is never the truth. It should also be mentioned that it is usually assumed that disturbance decoupling will not cause a decrease in fault diagnosis sensitivity or fault decoupling in the worst scenario. To tackle this problem within the framework of this chapter, a numerical optimisation-based approach is proposed that can be used to estimate the unknown input distribution matrix which does not cause the fault decoupling effect. As an answer to the second question, this work presents an alternative Unknown Input Filter (UIF) for non-linear systems, which is based on the general idea of the Unscented Kalman Filter (UKF) [16, 17]. This approach is based on an idea similar to that proposed in Refs. [2, 18], but the structure of the scheme is different and instead of the EKF the UKF is employed. To tackle the third problem, it is shown that the Interacting Multiple Model (IMM) algorithm can be employed for selecting an appropriate unknown input distribution matrix from a predefined set. The proposed solutions can be perceived as an alternative to the Takagi–Sugeno-based approach presented, e.g., in Ref. [19], which will be the subject of Chap. 6. Finally, it should be mentioned that some of the results portrayed in this chapter were originally presented in Refs. [18, 20, 21].

2.1 Unknown Input Decoupling

Let us consider a non-linear stochastic system given by the following equations:

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{h}(\boldsymbol{u}_k) + \boldsymbol{E}\boldsymbol{d}_k + \boldsymbol{L}\boldsymbol{f}_k + \boldsymbol{w}_k, \tag{2.1}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{C}\boldsymbol{x}_{k+1} + \boldsymbol{v}_{k+1}. \tag{2.2}$$

Note that the unknown input and fault distribution matrices, denoted by $\boldsymbol{E}$ and $\boldsymbol{L}$, are assumed (for the sake of simplicity) constant in this section. Such an assumption will be relaxed in Sect. 2.6, where a set of predefined matrices will be used instead. Moreover, it should be mentioned that this chapter focuses on faults that can influence the state equation (2.1), such as actuator faults. The case of sensor faults is beyond the scope of this section and will be investigated in the subsequent part of the book.

The main problem is to design a filter which is insensitive to the influence of the unknown input (external disturbances and modeling errors) while being sensitive to faults. The necessary condition for the existence of a solution to the unknown input decoupling problem is as follows:

$$\operatorname{rank}(\boldsymbol{CE}) = \operatorname{rank}(\boldsymbol{E}) = q \tag{2.3}$$

(see Ref. [2] for a comprehensive explanation). If the condition (2.3) is satisfied, then it is possible to calculate $\boldsymbol{H} = (\boldsymbol{CE})^{+} = \left[(\boldsymbol{CE})^T \boldsymbol{CE}\right]^{-1} (\boldsymbol{CE})^T$. Thus, by inserting (2.1) into (2.2) and then multiplying (2.2) by $\boldsymbol{H}$ it is straightforward to show that

$$\boldsymbol{d}_k = \boldsymbol{H}\Big[\boldsymbol{y}_{k+1} - \boldsymbol{C}\left[\boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{h}(\boldsymbol{u}_k) + \boldsymbol{L}\boldsymbol{f}_k + \boldsymbol{w}_k\right] - \boldsymbol{v}_{k+1}\Big]. \tag{2.4}$$

Substituting (2.4) into (2.1) for $\boldsymbol{d}_k$ gives

$$\boldsymbol{x}_{k+1} = \bar{\boldsymbol{g}}(\boldsymbol{x}_k) + \bar{\boldsymbol{h}}(\boldsymbol{u}_k) + \bar{\boldsymbol{E}}\boldsymbol{y}_{k+1} + \bar{\boldsymbol{L}}\boldsymbol{f}_k + \bar{\boldsymbol{w}}_k, \tag{2.5}$$

where

$$\bar{\boldsymbol{g}}(\cdot) = \boldsymbol{G}\boldsymbol{g}(\cdot), \quad \bar{\boldsymbol{h}}(\cdot) = \boldsymbol{G}\boldsymbol{h}(\cdot),$$
$$\bar{\boldsymbol{E}} = \boldsymbol{E}\boldsymbol{H}, \quad \bar{\boldsymbol{w}}_k = \boldsymbol{G}\boldsymbol{w}_k - \boldsymbol{E}\boldsymbol{H}\boldsymbol{v}_{k+1},$$

and

$$\boldsymbol{G} = \boldsymbol{I} - \boldsymbol{E}\boldsymbol{H}\boldsymbol{C}.$$

Consequently, the general observer structure is

$$\hat{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{g}}\left(\hat{\boldsymbol{x}}_k\right) + \bar{\boldsymbol{h}}(\cdot) + \bar{\boldsymbol{E}}\boldsymbol{y}_{k+1} + \boldsymbol{K}(\cdot), \tag{2.6}$$

where $\boldsymbol{K}(\cdot)$ is the state correction term. In order to make further deliberations more general, no particular form of $\boldsymbol{K}(\cdot)$ is assumed in the present and subsequent section.

Let us define a residual as a difference between the output of the system and its estimate:

$$\begin{aligned} \boldsymbol{z}_{k+1} &= \boldsymbol{y}_{k+1} - \boldsymbol{C}\hat{\boldsymbol{x}}_{k+1} \\ &= \boldsymbol{C}(\bar{\boldsymbol{g}}(\boldsymbol{x}_k) - \bar{\boldsymbol{g}}(\hat{\boldsymbol{x}}_k) - \boldsymbol{K}(\cdot)) + \bar{\boldsymbol{f}}_k + \boldsymbol{C}\bar{\boldsymbol{w}}_k + \boldsymbol{v}_{k+1}, \end{aligned} \tag{2.7}$$

where

$$\bar{\boldsymbol{f}}_k = \boldsymbol{C}\bar{\boldsymbol{L}}\boldsymbol{f}_k = \boldsymbol{C}\left[\boldsymbol{I}_n - \boldsymbol{E}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\boldsymbol{C}\right]\boldsymbol{L}\boldsymbol{f}_k. \tag{2.8}$$

A natural question arises: Is it possible that the fault will be decoupled from the residual? If so the proposed strategy seems to be useless as it will lead to undetected faults, which may have serious consequences regarding the performance of the system being diagnosed. An answer is provided in the subsequent section.

2.2 Preventing Fault Decoupling

It is usually assumed that a disturbance decoupling will not cause a decrease in fault diagnosis sensitivity or a fault decoupling in the worst scenario. But such an assumption is a rather unpractical tool in serious applications. Thus, to overcome such a challenging problem, the following theorem provides a simple rule for checking if the proposed unknown input observer will not decouple the effect of a fault from the residual. It relates the fault and unknown input distribution matrices denoted by $\boldsymbol{L}$ and $\boldsymbol{E}$, respectively. Moreover, let us assume that the following rank condition is satisfied:

$$\operatorname{rank}(\boldsymbol{CL}) = \operatorname{rank}(\boldsymbol{L}) = s. \tag{2.9}$$

Theorem 2.1 *The fault $\boldsymbol{f}_k$ will not be decoupled from the residual (2.7) if and only if the matrix*

$$[\boldsymbol{CE}\ \boldsymbol{CL}] \tag{2.10}$$

is a full-rank one.

Proof Let us suppose (theoretically) that $\operatorname{rank}(\boldsymbol{C}\bar{\boldsymbol{L}}) = s$. Then it can be shown that

$$\boldsymbol{f}_k = (\boldsymbol{C}\bar{\boldsymbol{L}})^{+}\bar{\boldsymbol{f}}_k, \tag{2.11}$$

which means that there exists a unique relationship between $\boldsymbol{f}_k$ and $\bar{\boldsymbol{f}}_k$ and hence the fault will not be decoupled from the residual. Unfortunately, the subsequent part of the proof shows that this is not always possible to attain. Indeed, (2.8) can be written into an equivalent form

$$\bar{\boldsymbol{f}}_k = \left[\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right]\boldsymbol{CL}\boldsymbol{f}_k. \tag{2.12}$$

Moreover, it can be observed that

$$\begin{aligned}&\left[\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right]^2\\ &= \boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T,\end{aligned} \tag{2.13}$$

which means that $\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T$ is an idempotent matrix. One of the fundamental properties of an idempotent matrix is that its rank is equal to the trace, i.e.,

$$\begin{aligned}&\text{rank}\left(\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right)\\ &= \text{trace}\left(\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right)\\ &= \text{trace}\,(\boldsymbol{I}_m) - \text{trace}\left(\boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right)\\ &= m - \text{trace}\left(\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\boldsymbol{CE}\right) = m - q.\end{aligned} \tag{2.14}$$

Thus, from (2.9) it is clear that

$$\begin{aligned}&\text{rank}\left(\left[\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right]\boldsymbol{CL}\right)\\ &\leq \min(m-q, s).\end{aligned} \tag{2.15}$$

On the other hand,

$$\begin{aligned}&\text{rank}\left(\left[\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right]\boldsymbol{CL}\right)\\ &\geq \text{rank}\left(\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right) + \text{rank}\,(\boldsymbol{CL}) - m\\ &= s - q.\end{aligned} \tag{2.16}$$

Finally,

$$\begin{aligned}&\max(s-q, 0) \leq \text{rank}\left(\left[\boldsymbol{I}_m - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\right]\boldsymbol{CL}\right)\\ &\leq \min(m-q, s).\end{aligned} \tag{2.17}$$

Thus, it is necessary to find an alternative condition under which

$$\begin{aligned}\bar{\boldsymbol{f}}_k &= \boldsymbol{CLf}_k - \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\boldsymbol{CLf}_k \\ &= \boldsymbol{CLf}_k - \boldsymbol{CLf}_k = \boldsymbol{0}.\end{aligned} \tag{2.18}$$

Indeed, any vector $\boldsymbol{CLf}_k \in \text{col}(\boldsymbol{CE})$, where

$$\text{col}(\boldsymbol{CE}) = \left\{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} = \boldsymbol{CE\beta} \quad \text{for some} \quad \boldsymbol{\beta} \in \mathbb{R}^q\right\}, \tag{2.19}$$

can be written as

$$\boldsymbol{CLf}_k = \boldsymbol{CE}\tilde{\boldsymbol{f}}_k, \tag{2.20}$$

for some non-zero vector $\tilde{\boldsymbol{f}}_k$. As a consequence,

$$\begin{aligned}&\boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\boldsymbol{CLf}_k \\ &= \boldsymbol{CE}\left[(\boldsymbol{CE})^T\boldsymbol{CE}\right]^{-1}(\boldsymbol{CE})^T\boldsymbol{CE}\tilde{\boldsymbol{f}}_k = \boldsymbol{CE}\tilde{\boldsymbol{f}}_k = \boldsymbol{CLf}_k.\end{aligned} \tag{2.21}$$

From the above discussion, it is clear that the proposed unknown input observer will not decouple the fault effect from the residual iff $\boldsymbol{CLf}_k \notin \text{col}(\boldsymbol{CE})$, which is equivalent to

$$\text{rank}\left(\left[\boldsymbol{CE}\ \boldsymbol{CLf}_k\right]\right) = q + 1 \tag{2.22}$$

for all $f_{i,k} \neq 0,\ i = 1, \ldots, s$. It is clear that (2.22) is equivalent to the fact that the only solution to (for all $f_{i,k} \neq 0,\ i = 1, \ldots, s$)

$$\alpha_1(\boldsymbol{CE})_1 + \alpha_2(\boldsymbol{CE})_2 + \cdots + \alpha_q(\boldsymbol{CE})_q + \alpha_{q+1}\boldsymbol{CLf}_k = 0, \tag{2.23}$$

is for $\alpha_i = 0,\ i = 1, \ldots, q + 1$. By further expansion of (2.23) to

$$\alpha_1(\boldsymbol{CE})_1 + \cdots + \alpha_q(\boldsymbol{CE})_q + \alpha_{q+1}f_{1,k}(\boldsymbol{CL})_1 + \cdots + \alpha_{q+1}f_{s,k}(\boldsymbol{CL})_s = 0, \tag{2.24}$$

it can be seen that the zero-valued solution to (2.24) is equivalent to the existence of a full-rank matrix (2.10), which completes the proof.

Since the fault decoupling prevention problem is solved, then it is possible to provide a set of approaches for designing unknown input observers and filters.

2.3 First- and Second-order Extended Unknown Input Observers

The approach presented in this section is dedicated for deterministic systems. The proposed strategy is based on the general framework of the second-order EKF. In particular, the section will show the design of both first- and second-order EUIO. Moreover, to make the presentation more general, the unknown input distribution matrix is assumed to be a time-varying one. Let us consider a non-linear discrete-time system described by (the fault free-case will be considered for the convergence analysis purposes)

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}\left(\boldsymbol{x}_k\right) + \boldsymbol{h}(\boldsymbol{u}_k) + \boldsymbol{E}_k \boldsymbol{d}_k, \tag{2.25}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{C}_{k+1}\boldsymbol{x}_{k+1}. \tag{2.26}$$

The problem is to design an observer that is insensitive to the influence of an unknown input. The necessary condition for the existence of a solution to the unknown input decoupling problem is

$$\operatorname{rank}(\boldsymbol{C}_{k+1}\boldsymbol{E}_k) = \operatorname{rank}(\boldsymbol{E}_k) = q \tag{2.27}$$

(see [15, p. 72, Lemma 3.1] for a comprehensive explanation). If the condition (2.27) is satisfied, then it is possible to calculate $\boldsymbol{H}_{k+1} = (\boldsymbol{C}_{k+1}\boldsymbol{E}_k)^{+}$, where $(\cdot)^{+}$ stands for the pseudo-inverse of its argument. Thus, by multiplying (2.26) by $\boldsymbol{H}_{k+1}$ and then inserting (2.25), it is straightforward to show that

$$\boldsymbol{d}_k = \boldsymbol{H}_{k+1}\left[\boldsymbol{y}_{k+1} - \boldsymbol{C}_{k+1}\left[\boldsymbol{g}\left(\boldsymbol{x}_k\right) + \boldsymbol{h}(\boldsymbol{u}_k)\right]\right]. \tag{2.28}$$

Substituting (2.28) into (2.25) gives

$$\boldsymbol{x}_{k+1} = \bar{\boldsymbol{g}}\left(\boldsymbol{x}_k\right) + \bar{\boldsymbol{h}}\left(\boldsymbol{u}_k\right) + \bar{\boldsymbol{E}}_k \boldsymbol{y}_{k+1}, \tag{2.29}$$

where

$$\begin{aligned} &\bar{\boldsymbol{g}}\left(\cdot\right) = \bar{\boldsymbol{G}}_k \boldsymbol{g}\left(\cdot\right), \quad \bar{\boldsymbol{h}}\left(\cdot\right) = \bar{\boldsymbol{G}}_k \boldsymbol{h}(\cdot) \\ &\bar{\boldsymbol{G}}_k = \boldsymbol{I} - \boldsymbol{E}_k \boldsymbol{H}_{k+1}\boldsymbol{C}_{k+1}, \quad \bar{\boldsymbol{E}}_k = \boldsymbol{E}_k \boldsymbol{H}_{k+1}. \end{aligned} \tag{2.30}$$

Thus, the unknown input observer for (2.25) and (2.26) is given as follows:

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1/k} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{C}_{k+1}\hat{\boldsymbol{x}}_{k+1/k}),$$

where

$$\hat{\boldsymbol{x}}_{k+1/k} = \bar{\boldsymbol{g}}\left(\hat{\boldsymbol{x}}_k\right) + \bar{\boldsymbol{h}}\left(\boldsymbol{u}_k\right) + \bar{\boldsymbol{E}}_k \boldsymbol{y}_{k+1}. \tag{2.31}$$

As a consequence, the second-order extended Kalman filter algorithm used for the state estimation of (2.25) and (2.26) can be given as follows:

$$\hat{\boldsymbol{x}}_{k+1/k} = \bar{\boldsymbol{g}}\left(\hat{\boldsymbol{x}}_k\right) + \bar{\boldsymbol{h}}\left(\boldsymbol{u}_k\right) + \bar{\boldsymbol{E}}_k \boldsymbol{y}_{k+1} + \boldsymbol{s}_k, \tag{2.32}$$

$$\boldsymbol{P}_{k+1/k} = \bar{\boldsymbol{A}}_k \boldsymbol{P}_k \bar{\boldsymbol{A}}_k^{\,T} + \boldsymbol{Q}_k, \tag{2.33}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^T \cdot \left(\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^T + \boldsymbol{R}_{k+1}\right)^{-1}, \tag{2.34}$$

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1/k} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{C}_{k+1}\hat{\boldsymbol{x}}_{k+1/k}), \tag{2.35}$$

$$\boldsymbol{P}_{k+1} = \left[\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{C}_{k+1}\right]\boldsymbol{P}_{k+1/k}, \tag{2.36}$$

where

$$\begin{aligned} \bar{\boldsymbol{A}}_k &= \left.\frac{\partial \bar{\boldsymbol{g}}\left(\boldsymbol{x}_k\right)}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k} = \bar{\boldsymbol{G}}_k \left.\frac{\partial \boldsymbol{g}\left(\boldsymbol{x}_k\right)}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k} \\ &= \bar{\boldsymbol{G}}_k \boldsymbol{A}_k, \end{aligned} \tag{2.37}$$

and

$$s_{i,k} = \frac{1}{2}\text{trace}\left[\boldsymbol{P}_k \left.\frac{\partial \bar{\boldsymbol{g}}_i\left(\boldsymbol{x}_k\right)^2}{\partial \boldsymbol{x}_k^2}\right|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k}\right], \quad i = 1, \ldots, n. \tag{2.38}$$

The algorithm (2.32)–(2.36) can be perceived as the second-order EKF for non-linear systems with an unknown input. It should also be pointed out that when $\boldsymbol{s}_k = \boldsymbol{0}$ then the algorithm (2.32)–(2.36) reduces to the first-order EUIO.

2.3.1 Convergence Analysis

An important property is the fact that the proposed algorithm is used for the deterministic systems (2.25) and (2.26), and hence there exists a design freedom regarding matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ that can be exploited for increasing the convergence rate of the EUIO. To tackle this challenging problem, the convergence conditions of the EUIO related to the matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are developed and carefully analysed.

Using (2.35), the state estimation error can be given as:

$$\boldsymbol{e}_{k+1} = \boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1} = \left[\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{C}_{k+1}\right]\boldsymbol{e}_{k+1/k}, \tag{2.39}$$

where

$$\boldsymbol{e}_{k+1/k} = \boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1/k} \approx \bar{\boldsymbol{A}}_k \boldsymbol{e}_k - \boldsymbol{s}_k. \tag{2.40}$$

Assuming that $\boldsymbol{e}_k \neq \boldsymbol{0}$ and defining an unknown diagonal matrix:

$$\boldsymbol{\beta}_k = \text{diag}(\beta_{1,k}, \ldots, \beta_{n,k}) \tag{2.41}$$

such that

$$-\boldsymbol{\beta}_k \boldsymbol{e}_k = \boldsymbol{s}_k, \tag{2.42}$$

it is possible to write

$$\begin{aligned} \boldsymbol{e}_{k+1/k} &= \boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1/k} \\ &= \boldsymbol{\alpha}_k \left[\bar{\boldsymbol{A}}_k + \boldsymbol{\beta}_k\right] \boldsymbol{e}_k = \boldsymbol{\alpha}_k \boldsymbol{Z}_k \boldsymbol{e}_k, \end{aligned} \tag{2.43}$$

where $\boldsymbol{\alpha}_k = \text{diag}(\alpha_{1,k}, \ldots, \alpha_{n,k})$ is an unknown diagonal matrix. Thus, using (2.43), the Eq. (2.39) becomes:

$$\boldsymbol{e}_{k+1} = \left[\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{C}_{k+1}\right] \boldsymbol{\alpha}_k \boldsymbol{Z}_k \boldsymbol{e}_k. \tag{2.44}$$

It is clear from (2.43) that $\boldsymbol{\alpha}_k$ represents the lineariztion error. This means that the convergence of the proposed observer is strongly related to the admissible bounds of the diagonal elements of $\boldsymbol{\alpha}_k$. Thus, the main objective of further deliberations is to show that these bounds can be controlled with the use of the instrumental matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$.

First let us start with the convergence conditions, which require the following assumptions:

Assumption 1 Following Ref. [22], it is assumed that the system given by (2.26) and (2.29) is locally uniformly rank observable. This guaranties that (see Ref. [22] and the references therein) that the matrix $\boldsymbol{P}_k$ is bounded, i.e., there exist positive scalars $\bar{\theta} > 0$ and $\underline{\theta} > 0$ such that

$$\underline{\theta} \boldsymbol{I} \preceq \boldsymbol{P}_k^{-1} \preceq \bar{\theta} \boldsymbol{I}. \tag{2.45}$$

Assumption 2 The matrix $\boldsymbol{A}_k$ is uniformly bounded and there exists $\boldsymbol{A}_k^{-1}$.

Moreover, let us define

$$\bar{\alpha}_k = \max_{j=1,\ldots,n} |\alpha_{j,k}|, \quad \underline{\alpha}_k = \min_{j=1,\ldots,n} |\alpha_{j,k}|. \tag{2.46}$$

where $\underline{\sigma}(\cdot)$ and $\bar{\sigma}(\cdot)$ denote the minimum and the maximum singular value of their arguments, respectively.

Theorem 2.2 *If*

$$\bar{\alpha}_k \leq \left[\underline{\alpha}_k^2 \frac{\underline{\sigma}(\boldsymbol{Z}_k)^2 \, \underline{\sigma}(\boldsymbol{C}_{k+1})^2 \, \underline{\sigma}\left(\boldsymbol{Z}_k \boldsymbol{P}_k \boldsymbol{Z}_k^T + \boldsymbol{Q}_k\right)}{\bar{\sigma}\left(\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1/k} \boldsymbol{C}_{k+1}^T + \boldsymbol{R}_{k+1}\right)} \right.$$

$$+ \frac{(1-\zeta)\underline{\sigma}\left(\boldsymbol{Z}_k\boldsymbol{P}_k\boldsymbol{Z}_k^T+\boldsymbol{Q}_k\right)}{\bar{\sigma}\left(\boldsymbol{Z}_k\right)^2\bar{\sigma}\left(\boldsymbol{P}_k\right)}\Bigg]^{\frac{1}{2}}, \tag{2.47}$$

where $0 < \zeta < 1$, *then the proposed extended unknown input observer is locally asymptotically convergent.*

Proof The main objective of further deliberations is to determine conditions for which the sequence $\{V_k\}_{k=1}^{\infty}$, defined by the Lyapunov candidate function

$$V_{k+1} = \boldsymbol{e}_{k+1}^T\boldsymbol{P}_{k+1}^{-1}\boldsymbol{e}_{k+1}, \tag{2.48}$$

is a decreasing one. Substituting (2.44) into (2.48) gives

$$\begin{aligned} V_{k+1} = &\boldsymbol{e}_k^T\boldsymbol{Z}_k^T\boldsymbol{\alpha}_k\left[\boldsymbol{I}-\boldsymbol{C}_{k+1}^T\boldsymbol{K}_{k+1}^T\right]\boldsymbol{P}_{k+1}^{-1} \\ &\times\left[\boldsymbol{I}-\boldsymbol{K}_{k+1}\boldsymbol{C}_{k+1}\right]\boldsymbol{\alpha}_k\boldsymbol{Z}_k\boldsymbol{e}_k. \end{aligned} \tag{2.49}$$

Using (2.36), it can be shown that

$$\left[\boldsymbol{I}-\boldsymbol{C}_{k+1}^T\boldsymbol{K}_{k+1}^T\right] = \boldsymbol{P}_{k+1/k}^{-1}\boldsymbol{P}_{k+1}. \tag{2.50}$$

Inserting (2.34) into $\left[\boldsymbol{I}-\boldsymbol{K}_{k+1}\boldsymbol{C}_{k+1}\right]$ yields

$$\begin{aligned} \left[\boldsymbol{I}-\boldsymbol{K}_{k+1}\boldsymbol{C}_{k+1}\right] = &\boldsymbol{P}_{k+1/k}\left[\boldsymbol{P}_{k+1/k}^{-1}-\boldsymbol{C}_{k+1}^T\right. \\ &\left.\times\left(\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^T+\boldsymbol{R}_{k+1}\right)^{-1}\boldsymbol{C}_{k+1}\right]. \end{aligned} \tag{2.51}$$

Substituting (2.50) and (2.51) into (2.49) gives

$$\begin{aligned} V_{k+1} = &\boldsymbol{e}_k^T\boldsymbol{Z}_k^T\boldsymbol{\alpha}_k\left[\boldsymbol{P}_{k+1/k}^{-1}-\boldsymbol{C}_{k+1}^T\right. \\ &\left.\times\left(\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^T+\boldsymbol{R}_{k+1}\right)^{-1}\boldsymbol{C}_{k+1}\right]\boldsymbol{\alpha}_k\boldsymbol{Z}_k\boldsymbol{e}_k. \end{aligned} \tag{2.52}$$

The sequence $\{V_k\}_{k=1}^{\infty}$ is decreasing when there exists a scalar ζ, $0 < \zeta < 1$, such that

$$V_{k+1} - (1-\zeta)V_k \leq 0. \tag{2.53}$$

Using (2.48) and (2.52), the inequality (2.53) can be written as

$$\begin{aligned} \boldsymbol{e}_k^T\Big[&\boldsymbol{Z}_k^T\boldsymbol{\alpha}_k\left[\boldsymbol{P}_{k+1/k}^{-1}-\boldsymbol{C}_{k+1}^T\right. \\ &\cdot\left.\left(\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^T+\boldsymbol{R}_{k+1}\right)^{-1}\boldsymbol{C}_{k+1}\right]\boldsymbol{\alpha}_k\boldsymbol{Z}_k \end{aligned}$$

$$-(1-\zeta)\boldsymbol{P}_k^{-1}\Big]\boldsymbol{e}_k \le 0. \tag{2.54}$$

Using the bounds of the Rayleigh quotient for $\boldsymbol{X} \succeq 0$, i.e., $\underline{\sigma}(\boldsymbol{X}) \le \frac{\boldsymbol{e}_k^T \boldsymbol{X} \boldsymbol{e}_k}{\boldsymbol{e}_k^T \boldsymbol{e}_k} \le \bar{\sigma}(\boldsymbol{X})$, the inequality (2.54) can be transformed into the following form:

$$\begin{aligned}
&\bar{\sigma}\left(\boldsymbol{Z}_k^T \boldsymbol{\alpha}_k \boldsymbol{P}_{k+1/k}^{-1} \boldsymbol{\alpha}_k \boldsymbol{Z}_k\right) \\
&- \underline{\sigma}\left(\boldsymbol{Z}_k^T \boldsymbol{\alpha}_k \boldsymbol{C}_{k+1}^T \left(\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1/k} \boldsymbol{C}_{k+1}^T + \boldsymbol{R}_{k+1}\right)^{-1}\right. \\
&\left.\times \boldsymbol{C}_{k+1} \boldsymbol{\alpha}_k \boldsymbol{Z}_k\right) - (1-\zeta)\underline{\sigma}\left(\boldsymbol{P}_k^{-1}\right) \le 0.
\end{aligned} \tag{2.55}$$

It is straightforward to show that

$$\begin{aligned}
&\bar{\sigma}\left(\boldsymbol{Z}_k^T \boldsymbol{\alpha}_k \boldsymbol{P}_{k+1/k}^{-1} \boldsymbol{\alpha}_k \boldsymbol{Z}_k\right) \\
&\le \bar{\sigma}(\boldsymbol{\alpha}_k)^2 \, \bar{\sigma}(\boldsymbol{Z}_k)^2 \, \bar{\sigma}\left(\boldsymbol{P}_{k+1/k}^{-1}\right),
\end{aligned} \tag{2.56}$$

and

$$\begin{aligned}
&\underline{\sigma}\left(\boldsymbol{Z}_k^T \boldsymbol{\alpha}_k \boldsymbol{C}_{k+1}^T \left(\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1/k} \boldsymbol{C}_{k+1}^T + \boldsymbol{R}_{k+1}\right)^{-1} \times \boldsymbol{C}_{k+1} \boldsymbol{\alpha}_k \boldsymbol{Z}_k\right) \\
&\ge \underline{\sigma}(\boldsymbol{\alpha}_k)^2 \, \underline{\sigma}(\boldsymbol{Z}_k)^2 \, \underline{\sigma}\left(\bar{\boldsymbol{C}}_{k+1}\right)^2 \\
&\quad \times \underline{\sigma}\left(\left(\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1/k} \boldsymbol{C}_{k+1}^T + \boldsymbol{R}_{k+1}\right)^{-1}\right) \\
&= \frac{\underline{\sigma}(\boldsymbol{\alpha}_k)^2 \, \underline{\sigma}(\boldsymbol{Z}_k)^2 \, \underline{\sigma}\left(\bar{\boldsymbol{C}}_{k+1}\right)^2}{\bar{\sigma}\left(\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1/k} \boldsymbol{C}_{k+1}^T + \boldsymbol{R}_{k+1}\right)}.
\end{aligned} \tag{2.57}$$

Applying (2.56) and (2.57) to (2.55) and then using (2.33), one can obtain (2.47).

Thus, if the condition (2.47) is satisfied, then $\{V_k\}_{k=1}^{\infty}$ is a decreasing sequence and hence, under the local uniform rank observability condition [22], the proposed observer is locally asymptotically convergent.

2.3.2 Design Principles

First-order Case

When first-order expansion is employed, then $\boldsymbol{s}_k$ in (2.32) should be $\boldsymbol{s}_k = \boldsymbol{0}$. This means that $\boldsymbol{\beta}_k = \boldsymbol{0}$ and hence $\boldsymbol{Z}_k = \bar{\boldsymbol{A}}_k$.

Remark 1 As can be observed by a straightforward comparison of (2.47) and (2.58), the convergence condition (2.47) is less restrictive than the solution obtained with the approach proposed in Ref. [22], which can be written as

$$\bar{\alpha}_k \leq \left(\frac{(1-\zeta)\underline{\sigma}\left(\bar{\boldsymbol{A}}_k \boldsymbol{P}_k \bar{\boldsymbol{A}}_k^T + \boldsymbol{Q}_k\right)}{\bar{\sigma}\left(\bar{\boldsymbol{A}}_k\right)^2 \bar{\sigma}\left(\boldsymbol{P}_k\right)} \right)^{\frac{1}{2}}. \tag{2.58}$$

However, (2.47) and (2.58) become equivalent when $\boldsymbol{E}_k \neq \boldsymbol{0}$, i.e., in all cases when unknown input is considered. This is because of the fact that the matrix $\bar{\boldsymbol{A}}_k$ is singular when $\boldsymbol{E}_k \neq \boldsymbol{0}$, which implies that $\underline{\sigma}\left(\bar{\boldsymbol{A}}_k\right) = 0$. Indeed, from (2.37),

$$\begin{aligned} \bar{\boldsymbol{A}}_k = \bar{\boldsymbol{G}}_k \boldsymbol{A}_k = \Big[\boldsymbol{I} - \boldsymbol{E}_k \Big[(\boldsymbol{C}_{k+1}\boldsymbol{E}_k)^T \boldsymbol{C}_{k+1} \boldsymbol{E}_k \Big]^{-1} \\ (\boldsymbol{C}_{k+1}\boldsymbol{E}_k)^T \boldsymbol{C}_{k+1} \Big] \boldsymbol{A}_k, \end{aligned} \tag{2.59}$$

and under Assumption 2, it is evident that $\bar{\boldsymbol{A}}_k$ is singular when

$$\boldsymbol{E}_k \left[(\boldsymbol{C}_{k+1}\boldsymbol{E}_k)^T \boldsymbol{C}_{k+1} \boldsymbol{E}_k \right]^{-1} (\boldsymbol{C}_{k+1}\boldsymbol{E}_k)^T \boldsymbol{C}_{k+1}$$

is singular. The singularity of the above matrix can be easily shown with the use of (2.27), i.e.,

$$\begin{aligned} &\operatorname{rank}\left(\boldsymbol{E}_k \left[(\boldsymbol{C}_{k+1}\boldsymbol{E}_k)^T \boldsymbol{C}_{k+1} \boldsymbol{E}_k \right]^{-1} (\boldsymbol{C}_{k+1}\boldsymbol{E}_k)^T \boldsymbol{C}_{k+1} \right) \\ &\leq \min\left[\operatorname{rank}(\boldsymbol{E}_k), \operatorname{rank}(\boldsymbol{C}_{k+1}) \right] = q. \end{aligned} \tag{2.60}$$

Taking into account the fact that $q < n$, the singularity of $\bar{\boldsymbol{A}}_k$ becomes evident.

Remark 2 It is clear from (2.47) that the bound of $\bar{\alpha}_k$ can be maximised by suitable settings of the instrumental matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$. Indeed, $\boldsymbol{Q}_k$ should be selected in such a way as to maximise

$$\underline{\sigma}\left(\bar{\boldsymbol{A}}_k \boldsymbol{P}_k \bar{\boldsymbol{A}}_k^T + \boldsymbol{Q}_k \right). \tag{2.61}$$

To tackle this problem, let us start with a similar solution to the one proposed in Ref. [23], i.e.,

$$\boldsymbol{Q}_k = \gamma \bar{\boldsymbol{A}}_k \boldsymbol{P}_k \bar{\boldsymbol{A}}_k^T + \delta_1 \boldsymbol{I}, \tag{2.62}$$

where $\gamma \geq 0$ and $\delta_1 > 0$. Substituting, (2.62) into (2.61) and taking into account that $\underline{\sigma}\left(\bar{\boldsymbol{A}}_k\right) = 0$, it can be shown that,

$$(1+\gamma)\underline{\sigma}\left(\bar{\boldsymbol{A}}_k\boldsymbol{P}_k\bar{\boldsymbol{A}}_k^T\right)+\delta_1\boldsymbol{I}=\delta_1\boldsymbol{I}. \tag{2.63}$$

Indeed, singularity of $\bar{\boldsymbol{A}}_k$, causes $\underline{\sigma}\left(\bar{\boldsymbol{A}}_k\boldsymbol{P}_k\bar{\boldsymbol{A}}_k^T\right)=0$, which implies the final result of (2.63). Thus, this solution boils down to the classical approach with constant $\boldsymbol{Q}_k=\delta_1\boldsymbol{I}$. It is, of course, possible to set $\boldsymbol{Q}_k=\delta_1\boldsymbol{I}$ with δ_1 large enough. As has been mentioned, the more accurate (near "true" values) the covariance matrices, the better the convergence rate. This means that, in the deterministic case, both matrices should be zero ones. On the other hand, such a solution may lead to the divergence of the observer. To tackle this problem, a compromise between the convergence and the convergence rate should be established. This can be easily done by setting $\boldsymbol{Q}_k$ as

$$\boldsymbol{Q}_k=(\gamma\boldsymbol{\varepsilon}_k^T\boldsymbol{\varepsilon}_k+\delta_1)\boldsymbol{I},\quad \boldsymbol{\varepsilon}_k=\boldsymbol{y}_k-\boldsymbol{C}_k\hat{\boldsymbol{x}}_k, \tag{2.64}$$

with $\gamma>0$ and $\delta_1>0$ large and small enough, respectively. Since the form of $\boldsymbol{Q}_k$ is established, then it is possible to obtain $\boldsymbol{R}_k$ in such a way as to minimise

$$\bar{\sigma}\left(\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^T+\boldsymbol{R}_{k+1}\right). \tag{2.65}$$

To tackle this problem, let us start with the solution proposed in Refs. [22, 23]:

$$\boldsymbol{R}_{k+1}=\beta_1\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^T+\delta_2\boldsymbol{I}, \tag{2.66}$$

with $\beta_1\geq 0$ and $\delta_2>0$. Substituting (2.66) into (2.65) gives

$$(1+\beta_1)\bar{\sigma}\left(\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^T\right)+\delta_2\boldsymbol{I}. \tag{2.67}$$

Thus, β_1 in (2.66) should be set so as to minimise (2.67), which implies ($\beta_1=0$)

$$\boldsymbol{R}_{k+1}=\delta_2\boldsymbol{I}, \tag{2.68}$$

with δ_2 small enough.

Second-order Case

From Remark 2 it is clear that the matrix $\boldsymbol{R}_k$ should be set according to (2.68) both in the first- and the second-order case. Indeed, it can be easily observed that its derivation does not depend on the form of $\boldsymbol{Z}_k$. A significantly different situation takes place in the case of $\boldsymbol{Q}_k$. Indeed, when (2.64) is employed to set $\boldsymbol{Q}_k$, then from (2.33) and (2.36) it is evident that $\boldsymbol{P}_k$ is large in the sense of its singular values as well as in the trace. Thus, from (2.38) and (2.42) it is clear that the diagonal entries of $\boldsymbol{\beta}_k$ should be relatively large. On the other hand, by observing (2.47) it can be concluded that the upper bound of $\boldsymbol{\alpha}_k$ strongly depends on $\boldsymbol{Z}_k$ while

$$\begin{aligned}\bar{\sigma}\left(\mathbf{Z}_k\right) &\leq \bar{\sigma}\left(\bar{\boldsymbol{A}}_k\right) + \bar{\sigma}\left(\boldsymbol{\beta}_k\right)\\ &= \bar{\sigma}\left(\bar{\boldsymbol{A}}_k\right) + \max_{i=1,\ldots,n} |\beta_{i,k}|. \end{aligned} \tag{2.69}$$

The above-described situation results in the fact that the upper bound of $\boldsymbol{\alpha}_k$ can be very small (while using (2.64)), which may lead to the divergence of the observer. Thus, the only solution is to set $\boldsymbol{Q}_k$ in a conventional way, i.e.,

$$\boldsymbol{Q}_{k+1} = \delta_3 \boldsymbol{I}, \tag{2.70}$$

with $\delta_3 > 0$ small enough.

2.4 Unscented Kalman Filter

The main objective of this section is to provide a general framework for designing the UIF for non-linear stochastic systems, which is based on the UKF.

As has already been mentioned, state estimation for non-linear stochastic systems is a difficult and important problem for modern fault diagnosis and control systems (see the recent books in the subject area for a complete survey and explanations [2, 3, 24–27]). As can be observed in the literature, the most frequently used approach to state estimation of non-linear stochastic systems is to use the celebrated EKF. However, the linearised non-linear transformations of the state and/or output are reliable only if there is no excessive difference between the local behaviour compared to the original non-linear transformation. If this is not the case, then the EKF will suffer from divergence. However, in the preceding part of this chapter the process and measurement noise matrices are used as instrumental matrices that can significantly improve the convergence performance (see Refs. [2, 18] for a comprehensive survey). Unfortunately, in the stochastic case, $\boldsymbol{Q}$ and $\boldsymbol{R}$ have to play their primary role as covariance matrices.

As indicated in Ref. [16], *it is easier to approximate a probability distribution than it is to approximate an arbitrary non-linear function or transformation.*

Bearing in mind this sentence, the idea of an Unscented Transform (UT) was applied along with the celebrated Kalman filter in order to form the UKF. To make the chapter self-contained, the subsequent points will describe the UT and the algorithm of the UKF.

Finally, it should be underlined that the reader is referred to Ref. [16] (and the references therein) for a large number of practical examples showing the superiority of the UKF over the conventional EKF. Thus, the subsequent part of the chapter is focused on developing a new UKF-based scheme rather than showing its superiority over the EKF.

2.4.1 Unscented Transform

The unscented transform boils down to approximating the mean and covariance of the so-called sigma points after the non-linear transformation $\boldsymbol{h}(\cdot)$. The mean and covariance of sigma points are given as $\bar{\boldsymbol{x}}$ and $\boldsymbol{P}$, while the UT procedure is Ref. [16]

1. Generate k sigma points,

$$\boldsymbol{X}_i, \quad i = 1, \ldots, k, \tag{2.71}$$

   with the mean $\bar{\boldsymbol{x}}$ and covariance $\boldsymbol{P}$.
2. Obtain a non-linear transformation of each sigma point (cf. Fig. 2.1),

$$\boldsymbol{X}_i^t = \boldsymbol{h}(\boldsymbol{X}_i), \quad i = 1, \ldots, k. \tag{2.72}$$

3. Calculate the weighted mean of the transformed points,

$$\bar{\boldsymbol{x}}^t = \sum_{i=1}^{k} W^i \boldsymbol{X}_i^t. \tag{2.73}$$

4. Calculate the covariance of the transformed points,

$$\boldsymbol{P}^t = \sum_{i=1}^{k} W^i \left[\boldsymbol{X}_i^t - \bar{\boldsymbol{x}}^t\right] \cdot \left[\boldsymbol{X}_i^t - \bar{\boldsymbol{x}}^t\right]^T. \tag{2.74}$$

Note that the sigma points can be generated with various scenarios [16, 17], and one of them will be described in the subsequent point. It should also be mentioned that, in order to provide an unbiased estimate [16], the weights should satisfy

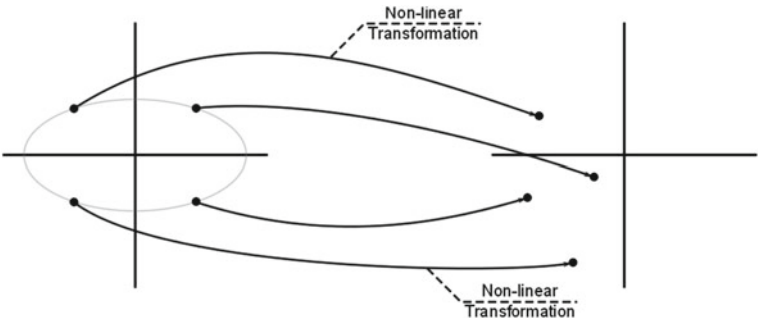

Fig. 2.1 Unscented transform

$$\sum_{i=1}^{k} W^i = 1. \tag{2.75}$$

2.4.2 Principle of the UKF-based UIF

Let us consider a non-linear, discrete-time fault-free system, i.e., (2.1) and (2.2) for $\boldsymbol{f}_k = \boldsymbol{0}$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{h}(\boldsymbol{u}_k) + \boldsymbol{E}\boldsymbol{d}_k + \boldsymbol{w}_k, \tag{2.76}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{C}\boldsymbol{x}_{k+1} + \boldsymbol{v}_{k+1}. \tag{2.77}$$

The UKF [17] can be perceived a derivative-free alternative to the extended Kalman filter in the framework of state estimation. The UKF calculates the mean and covariance of a random variable, which undergoes a non-linear transformation by utilising a deterministic "sampling" approach. Generally, $2n + 1$, *sigma* points are chosen based on a square-root decomposition of the prior covariance. These sigma points are propagated through true nonlinearity, without any approximation, and then a weighted mean and covariance are taken, as described in Sect. 2.4.1.

The presented form of the UKF is based on the general structure of the unknown input observer (2.6) and by taking into account the fact that the output equation (2.77) is linear.

The UKF involves a recursive application of these sigma points to state-space equations. The standard UKF implementation for state estimation uses the following variable definitions:

- $\lambda = 2n(\alpha^2 - 1)$,
- $W_0^m = \frac{\lambda}{n+\lambda}$,
- $W_0^c = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta$,
- $W_i^m = W_i^c = \frac{1}{2(n+\lambda)}$,
- $\eta = \sqrt{n+\lambda}$,

where W_i^m is a set of scalar weights, and λ and η are scaling parameters. The constant α determines the spread of the sigma points around $\hat{\boldsymbol{x}}$ and is usually set to $10^{-4} \leq \alpha \leq 1$. The constant β is used to incorporate prior knowledge of the distribution (for the Gaussian distribution, $\beta = 2$ is an optimal choice). The UKF algorithm is as follows:

Initialise with

$$\hat{\boldsymbol{x}}_0 = \mathcal{E}[\boldsymbol{x}_0], \quad \boldsymbol{P}_0 = \mathcal{E}[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T], \tag{2.78}$$

For $k \in \{1, \ldots, \infty\}$

Calculate $2n + 1$ sigma points:

$$\hat{\boldsymbol{X}}_{k-1} = [\hat{\boldsymbol{x}}_{k-1} \ \hat{\boldsymbol{x}}_{k-1} + \eta S(1), \ldots, \hat{\boldsymbol{x}}_{k-1} + \eta S(n), \\ \hat{\boldsymbol{x}}_{k-1} - \eta S(1), \ldots, \hat{\boldsymbol{x}}_{k-1} - \eta S(n)], \tag{2.79}$$

where $\boldsymbol{S} = \sqrt{\boldsymbol{P}_{k-1}}$ and $\boldsymbol{S}(j)$ stands for the jth column of $\boldsymbol{S}$.
Time update equations:

$$\hat{\boldsymbol{X}}_{i,k|k-1} = \bar{\boldsymbol{g}}\left(\hat{\boldsymbol{X}}_{i,k-1}\right) + \bar{\boldsymbol{h}}(\boldsymbol{u}_k) + \bar{\boldsymbol{E}}\boldsymbol{y}_{k+1}, \quad i = 0, \ldots, 2n, \tag{2.80}$$

$$\hat{\boldsymbol{x}}_{k,k-1} = \sum_{i=0}^{2n} W_i^{(m)} \hat{\boldsymbol{X}}_{i,k|k-1}, \tag{2.81}$$

$$\boldsymbol{P}_{k,k-1} = \sum_{i=0}^{2n} W_i^{(c)} [\hat{\boldsymbol{X}}_{i,k|k-1} \\ - \hat{\boldsymbol{x}}_{k,k-1}][\hat{\boldsymbol{X}}_{i,k|k-1} - \hat{\boldsymbol{x}}_{k,k-1}]^T + \boldsymbol{Q}. \tag{2.82}$$

Measurement update equations:

$$\boldsymbol{P}_{y_k y_k} = \boldsymbol{C}\boldsymbol{P}_{k,k-1}\boldsymbol{C}^T + \boldsymbol{R},$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k,k-1}\boldsymbol{C}^T \boldsymbol{P}_{y_k y_k}^{-1}, \tag{2.83}$$

$$\hat{\boldsymbol{y}}_{k,k-1} = \boldsymbol{C}\hat{\boldsymbol{x}}_{k,k-1}, \tag{2.84}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k,k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k,k-1}), \tag{2.85}$$

$$\boldsymbol{P}_k = [\boldsymbol{I}_n - \boldsymbol{K}_k\boldsymbol{C}]\boldsymbol{P}_{k,k-1}. \tag{2.86}$$

2.5 Determination of an Unknown Input Distribution Matrix

As a result of the deliberations presented in the preceding sections, the matrix $\boldsymbol{E}$ should satisfy the following conditions:

$$\operatorname{rank}(\boldsymbol{CE}) = \operatorname{rank}(\boldsymbol{E}) = q, \tag{2.87}$$

where

$$[\boldsymbol{CE} \ \boldsymbol{CL}] \tag{2.88}$$

should be a full rank one, which means that

$$\operatorname{rank}([\boldsymbol{CE} \ \boldsymbol{CL}]) = \min(m, s + q). \tag{2.89}$$

Thus, the set of matrices $\boldsymbol{E}$ satisfying (2.87) and (2.89) is given by

$$\begin{aligned}\mathbb{E} = \big\{\boldsymbol{E} \in \mathbb{R}^{n\times q} : \operatorname{rank}(\boldsymbol{CE}) = q \wedge \operatorname{rank}(\boldsymbol{E}) = q \wedge \operatorname{rank}([\boldsymbol{CE}\ \boldsymbol{CL}]) \\ = \min(m, s+q)\big\}. \end{aligned} \tag{2.90}$$

It should be strongly underlined that $\mathbb{E}$ is not convex, which significantly complicates the problem and limits the spectrum of possible approaches that can be used for settling the determination of the unknown input distribution matrix.

The subsequent part of this section presents a numerical algorithm that can be used for estimating the unknown input distribution matrix $\boldsymbol{E}$ based on a set of input–output measurements $\{(\boldsymbol{u}_k, \boldsymbol{y}_k)\}_{k=1}^{n_t}$.

To settle the problem of numerical estimation of $\boldsymbol{E}$, the following optimisation criterion is selected:

$$\hat{\boldsymbol{E}} = \arg\min_{\boldsymbol{E}\in\mathbb{E}} J(\boldsymbol{E}) \tag{2.91}$$

with

$$J(\boldsymbol{E}) = \frac{1}{mn_t}\sum_{k=1}^{n_t} z_k^T z_k, \tag{2.92}$$

where z_k stands for the residual defined by (2.7) and $\hat{\boldsymbol{E}}$ is an estimate of $\boldsymbol{E}$.

It is important to underline that the computation of (2.92) requires the run of the proposed UIF for a given instance of the unknown input distribution matrix $\boldsymbol{E}$. The computation of the cost function (2.92) is definitely the most time-consuming part of the proposed algorithm. On the other hand, the computation time and the resulting computational burden are not of paramount importance since the proposed algorithm performs off-line. Indeed, only the result of the proposed algorithm, being an estimate of the unknown input distribution matrix $\boldsymbol{E}$, is utilised on-line for the unknown input decoupling.

The outline of the proposed algorithm is as follows:

Step 1: Obtain the fault-free input–output data set from the system $\{(\boldsymbol{u}_k, \boldsymbol{y}_k)\}_{k=1}^{n_t}$.

Step 2: Initialise the algorithm with some initial value of $\boldsymbol{E}$ satisfying (2.87) and (2.88).

Step 3: Use an optimisation strategy to find an estimate of $\boldsymbol{E}$ for which (2.92) reaches its minimum and conditions (2.87) and (2.90) are satisfied.

Similarly as in the case of (2.8), i.e., by following with $\tilde{\boldsymbol{d}}_k$ in a similar way as with $\bar{\boldsymbol{f}}_k$ in (2.8), it can be shown that the fault-free residual is

$$\begin{aligned} z_{k+1} &= \boldsymbol{y}_{k+1} - \boldsymbol{C}\hat{\boldsymbol{x}}_{k+1} \\ &= \boldsymbol{C}\left(\bar{\boldsymbol{g}}\left(\boldsymbol{x}_k\right) - \bar{\boldsymbol{g}}\left(\hat{\boldsymbol{x}}_k\right) - \boldsymbol{K}(\cdot)\right) + \tilde{\boldsymbol{d}}_k + \boldsymbol{C}\bar{\boldsymbol{w}}_k + \boldsymbol{v}_{k+1}, \end{aligned} \tag{2.93}$$

where

$$\tilde{\boldsymbol{d}}_k = \boldsymbol{C}\left[\boldsymbol{I}_n - \hat{\boldsymbol{E}}\left[(\boldsymbol{C}\hat{\boldsymbol{E}})^T\boldsymbol{C}\hat{\boldsymbol{E}}\right]^{-1}(\boldsymbol{C}\hat{\boldsymbol{E}})^T\boldsymbol{C}\right]\bar{\boldsymbol{d}}_k. \tag{2.94}$$

Alternatively, assuming $\bar{\boldsymbol{d}}_k = \boldsymbol{E}\boldsymbol{d}_k$, it can be expressed by

$$\tilde{\boldsymbol{d}}_k = \boldsymbol{C}\left[\boldsymbol{I}_n - \hat{\boldsymbol{E}}\left[(\boldsymbol{C}\hat{\boldsymbol{E}})^T\boldsymbol{C}\hat{\boldsymbol{E}}\right]^{-1}(\boldsymbol{C}\hat{\boldsymbol{E}})^T\boldsymbol{C}\right]\boldsymbol{E}\boldsymbol{d}_k. \tag{2.95}$$

Following the same line of reasoning as in the proof of Theorem 2.1, it can be shown that for any vector $\boldsymbol{C}\boldsymbol{E}\boldsymbol{d}_k \in \mathrm{col}(\boldsymbol{C}\hat{\boldsymbol{E}})$, the effect of an unknown input $\tilde{\boldsymbol{d}}_k$ will be decoupled from the residual, i.e., $\tilde{\boldsymbol{d}}_k = \boldsymbol{0}$.

Based on the above deliberations, it seems that an alternative approach is:

Step 0: Obtain the fault-free input–output data set from the system $\{(\boldsymbol{u}_k, \boldsymbol{y}_k)\}_{k=1}^{n_t}$.

Step 1: Estimate $\bar{\boldsymbol{d}}_k$ for $k = 1, \ldots, n_t$ with, e.g., an augmented UKF.

Step 2: Find a basis of $[\bar{\boldsymbol{d}}_1, \ldots, \bar{\boldsymbol{d}}_{n_t}]$ (e.g. an orthonormal basis), which will constitute an estimate of $\boldsymbol{E}$.

Apart from the unquestionable appeal of the above algorithm, it does not take into account that the conditions (2.87) and (2.90) must be satisfied. On the other hand, it was empirically proven that, due to the process and measurement noise, accurate estimation of $\bar{\boldsymbol{d}}_k$ (for $k = 1, \ldots, n_t$) is impossible, and hence Step 2 of the above algorithm cannot be realised with expected results.

Thus, the only fruitful conclusion is that an optimal estimate of $\boldsymbol{E}$ is not unique, which will undoubtedly facilitate the performance of the optimisation-based approach presented in the subsequent part of this section.

Taking into account all the above-mentioned difficulties, it is proposed to use the adaptive random search algorithm [2, 28] to solve (2.91). The algorithm has proven to be very reliable in various global optimisation problems, which also justifies its application for this particular task.

The search process of the ARS can be split into two phases. The first phase (variance-selection phase) consists in selecting an element from the sequence

$$\{\sigma^{(i)}\}, \quad i = 1, \ldots, i_{\max}, \tag{2.96}$$

where $\sigma^{(1)}$ stands for an initial standard deviation selected by the designer (forming the covariance matrix $\Sigma = \sigma\boldsymbol{I}_{n\times q}$, where $n \times q$ is the number of elements of $\boldsymbol{E}$), and

$$\sigma^{(i)} = 10^{(-i+1)}\sigma^{(1)}. \tag{2.97}$$

In this way, the range of σ ensures both proper exploration properties over the search space and sufficient accuracy of optimum localisation. Larger values of σ decrease

the possibility of getting stuck in a local minimum. The second phase (variance-exploration phase) is dedicated to exploring the search space with the use of σ obtained from the first phase and consists in repetitive random perturbation of the best point obtained in the first phase. The scheme of the ARS algorithm is as follows:

0. Input data

- $\sigma^{(1)}$: The initial standard deviation,
- $j_{\max}$: The number of iterations in each phase,
- $i_{\max}$: The number of standard deviations (σ^i) changes,
- $k_{\max}$: The global number of algorithm runs,
- $\boldsymbol{E}^{(0)}$: The initial value of the unknown input distribution matrix.

1. Initialise

(1.1) Generate $\boldsymbol{E}_{\text{best}} \rightarrow \boldsymbol{E}^0$, satisfying (2.87) and (2.90), $k \rightarrow 1$, $i \rightarrow 1$.

2. Variance-selection phase

(2.1) $j \rightarrow 1, \boldsymbol{E}^{(j)} \rightarrow \boldsymbol{E}^{(0)}$ and $\sigma^{(i)} \rightarrow 10^{(-i+1)}\sigma^{(1)}$.
(2.2) Perturb $\boldsymbol{E}^{(j)}$ to get a new trial point $\boldsymbol{E}_+^{(j)}$ satisfying (2.87) and (2.90).
(2.3) If $J(\boldsymbol{E}_+^{(j)}) \leq J(\boldsymbol{E}^{(j)})$ then $\boldsymbol{E}^{(j+1)} \rightarrow \boldsymbol{E}_+^{(j)}$
else $\boldsymbol{E}^{(j+1)} \rightarrow \boldsymbol{E}^{(j)}$.
(2.4) If $J(\boldsymbol{E}_+^{(j)}) \leq J(\boldsymbol{E}_{\text{best}})$ then
$\boldsymbol{E}_{\text{best}} \rightarrow \boldsymbol{E}_+^{(j)}$, $i_{\text{best}} \rightarrow i$.
(2.5) If $(j \leq j_{\max}/i)$ then $j \rightarrow j + 1$ and go to (2.2).
(2.6) If $(i < i_{\max})$ then set $i \rightarrow i + 1$ and go to (2.1).

3. Variance-exploration phase

(3.1) $j \rightarrow 1, \boldsymbol{E}^{(j)} \rightarrow \boldsymbol{E}_{\text{best}}, i \rightarrow i_{\text{best}}$
and $\sigma^{(i)} \rightarrow 10^{(-i+1)}\sigma^{(1)}$.
(3.2) Perturb $\boldsymbol{E}^{(j)}$ to get a new trial point $\boldsymbol{E}_+^{(j)}$ satisfying (2.87) and (2.90).
(3.3) If $J(\boldsymbol{E}_+^{(j)}) \leq J(\boldsymbol{E}^{(j)})$ then $\boldsymbol{E}^{(j+1)} \rightarrow \boldsymbol{E}_+^{(j)}$
else $\boldsymbol{E}^{(j+1)} \rightarrow \boldsymbol{E}^{(j)}$.
(3.4) If $J(\boldsymbol{E}_+^{(j)}) \leq J(\boldsymbol{E}_{\text{best}})$ then $\boldsymbol{E}_{\text{best}} \rightarrow \boldsymbol{E}_+^{(j)}$.
(3.5) If $(j \leq j_{\max})$ then $j \rightarrow j + 1$ and go to Step 3.2.
(3.6) If $(k \rightarrow k_{\max})$ then STOP.
(3.7) $k \rightarrow k + 1, \boldsymbol{E}^{(0)} \rightarrow \boldsymbol{E}_{\text{best}}$ and resume from (2.1).

The perturbation phase (the points (2.2) and (3.2) of the algorithm) is realised according to

$$\boldsymbol{E}_+^{(j)} = \boldsymbol{E}^{(j)} + \boldsymbol{Z}, \tag{2.98}$$

where each element of $\boldsymbol{Z}$ is generated according to $\mathcal{N}(\boldsymbol{0}, \sigma^i \boldsymbol{I})$. When a newly generated $\boldsymbol{E}^{(j)}$ does not satisfy (2.87) and (2.90), then the perturbation phase (2.98) is repeated.

It should also be noted that for some $\boldsymbol{E}^{(j)}$ the proposed UIF may diverge, e.g., due to the loss of observability or a large mismatch with the real system. A simple remedy is to impose a bound (possibly large) ζ on $J(\boldsymbol{E}^{(j)})$, which means that when this bound is exceeded then the UIF is terminated and $J(\boldsymbol{E}^{(j)}) = \zeta$.

2.6 Design of the UIF with Varying Unknown Input Distribution Matrices

The UIF proposed in this section is designed in such a way that it will be able to tackle the problem of automatically changing (or mixing) the influence of unknown input distribution matrices according to system behaviour. In other words, the user can design a number of such matrices in order to cover different operating conditions. Thus, having such a set of matrices, it is possible to design a bank of UIFs and the algorithm should use them to obtain the best unknown input decoupling and state estimation. In order to realise this task, the Interacting Multiple-Model (IMM) approach [29] is used. The subsequent part of this section shows a comprehensive description of the UIF and the IMM.

The IMM solution consists of a filter for each disturbance matrix (corresponding to a particular model of the system), an estimate mixer at the input of the filters, and an estimate combiner at the output of the filters. The IMM works as a recursive estimator. In each recursion it has four steps:

1. Interacting or mixing of model-conditional estimates, in which the input to the filter matched to a certain mode is obtained by mixing the estimates of all filters from the previous time instant under the assumption that this particular mode is in effect at the present time;
2. Model-conditional filtering, performed in parallel for each mode;
3. Model probability update, based on model-conditional innovations and likelihood functions;
4. Estimate combination, which yields the overall state estimate according to the probabilistically weighted sum of updated state estimates of all the filters.

The probability of a mode plays a crucial role in determining the weights in the combination of the state estimates and covariances for the overall state estimate. Figure 2.2 shows the block diagram of the classic IMM algorithm [29], where

- $\hat{\boldsymbol{x}}_{k+1|k+1}$ is the state estimate for time k using measurements through time $(k+1|k+1)$ based on N models;
- $\hat{\boldsymbol{x}}^{j}_{k+1|k+1}$ is the state estimate for time k using measurements through time $(k+1|k+1)$ based on model j;
- Λ^{j}_{k} is the model likelihood at time k based on model j;
- μ_k is the vector of model probabilities at time k when all the likelihoods Λ^{j}_{k} have been considered during model probability update.

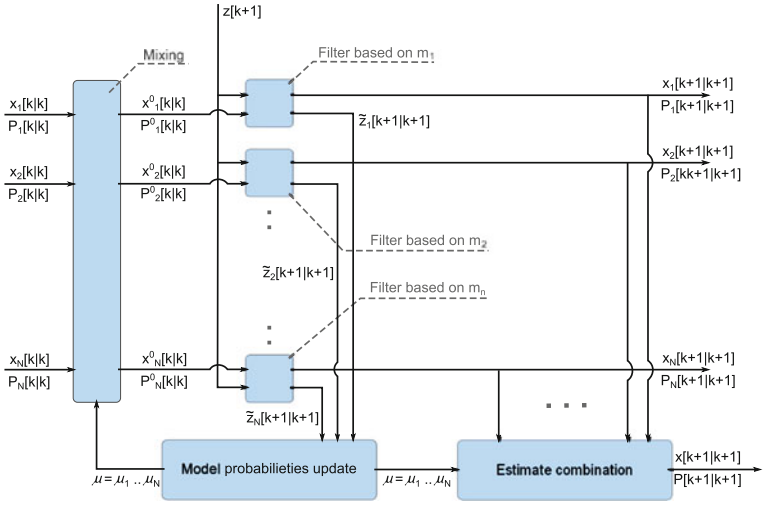

Fig. 2.2 IMM algorithm

With the assumption that model switching is governed by an underlying Markov [29] chain, an interacting mixer at the input of the N filters uses the model probabilities μ_k and the model switching probabilities p_{ij} to compute a mixed (initial or a priori) estimate $\hat{\boldsymbol{X}}^{0j}_{k|k}$ for N filters. The interacting mixer blends the previous state estimates based on N models to obtain new state estimates. The mixing gains $\mu^{i|j}_{k-1|k-1}$ are computed from the preceding model probabilities μ^{i}_{k-1} and the model switching probabilities p_{ij} in the model probability update.

At the beginning of a filtering cycle, all filters use an a priori mixed estimate $\hat{\boldsymbol{X}}^{0j}_{k-1|k-1}$ and the current measurement $\boldsymbol{y}_k$ to compute a new estimate $\hat{\boldsymbol{X}}^{j}_{k|k}$ and the likelihood Λ^{j}_{k} for the jth model filter. The likelihoods, prior model probabilities, and model switching probabilities are then used by the model probability update to compute the new model probabilities. The overall state estimate $\hat{\boldsymbol{X}}_{k|k}$ is then computed by an estimate combiner with the new state estimates and their probabilities.

The algorithm presented below is a combination of the UIF and the IMM and constitutes a solution to the challenging problem of designing the UIF for a set of predefined unknown input distribution matrices $\{\boldsymbol{E}_j\}_{j=1}^{N}$.

Step 1: Mixing state estimates

The filtering process starts with "a priori" state estimates $\hat{\boldsymbol{X}}^{j}_{k-1|k-1}$, state error covariances $\boldsymbol{P}_{k-1|k-1}$ and the associated probabilities μ^{j}_{k-1} for each jth filter model corresponding to the jth unknown input distribution matrix. The initial or mixed state estimate and covariance for the jth model at time k is computed as

$$\bar{c_j} = \sum_{i=1}^{N} p_{ij}\mu_{k-1}^{i}, \tag{2.99}$$

$$\mu_{k-1|k-1}^{i|j} = \frac{1}{\bar{c_j}} p_{ij}\mu_{k-1}^{i}, \tag{2.100}$$

$$\hat{\boldsymbol{X}}_{k-1|k-1}^{0j} = \sum_{i=1}^{N} \hat{\boldsymbol{X}}_{k-1|k-1}^{i}\mu_{k-1|k-1}^{i|j}, \tag{2.101}$$

$$\begin{aligned}\boldsymbol{P}_{k-1|k-1}^{0j} = \sum_{i=1}^{N} \big[&\boldsymbol{P}_{k-1|k-1}^{i} \\ &+ \big(\hat{\boldsymbol{X}}_{k-1|k-1}^{i} - \hat{\boldsymbol{X}}_{k-1|k-1}^{0j}\big) \\ &\cdot \big(\hat{\boldsymbol{X}}_{k-1|k-1}^{i} - \hat{\boldsymbol{X}}_{k-1|k-1}^{0j}\big)^{T}\big]\mu_{k-1|k-1}^{i|j}.\end{aligned} \tag{2.102}$$

p_{ij} is the assumed transition probability for switching from model i to model j, and $\bar{c_j}$ is a normalisation constant. For every state estimate $\hat{\boldsymbol{X}}_{k|k}^{i}$ and $\hat{\boldsymbol{X}}_{k-1|k-1}^{i}$, there is a corresponding covariance $\boldsymbol{P}_{k|k}^{i}$ and $\boldsymbol{P}_{k-1|k-1}^{i}$.

Step 2: Model-conditioned update

Calculate sigma points (for each jth model):

$$\begin{aligned}\hat{\boldsymbol{X}}_{k-1}^{j} = \big[&\hat{\boldsymbol{X}}_{k-1|k-1}^{0j} \;\; \hat{\boldsymbol{X}}_{k-1|k-1}^{0j} + \eta\sqrt{\boldsymbol{P}_{k-1|k-1}^{0j}} \\ &\hat{\boldsymbol{X}}_{k-1|k-1}^{0j} - \eta\sqrt{\boldsymbol{P}_{k-1|k-1}^{0j}}\big].\end{aligned} \tag{2.103}$$

Time update (for each jth model):

$$\hat{\boldsymbol{X}}_{i,k|k-1}^{j} = \bar{\boldsymbol{g}}\left(\hat{\boldsymbol{X}}_{i,k-1}^{j}\right) + \bar{\boldsymbol{h}}(\boldsymbol{u}_k) + \bar{\boldsymbol{E}}\boldsymbol{y}_{k+1},\ i = 0, \ldots, 2n, \tag{2.104}$$

$$\hat{\boldsymbol{x}}_{k,k-1}^{j} = \sum_{i=0}^{2n} W_i^{(m)}\hat{\boldsymbol{X}}_{i,k|k-1}^{j}, \tag{2.105}$$

$$\begin{aligned}\boldsymbol{P}_{k,k-1}^{j} = \sum_{i=0}^{2n} W_i^{(c)}\big[&\hat{\boldsymbol{X}}_{i,k|k-1}^{j} \\ &- \hat{\boldsymbol{x}}_{k,k-1}^{j}\big]\big[\hat{\boldsymbol{X}}_{i,k|k-1}^{j} - \hat{\boldsymbol{x}}_{k,k-1}^{j}\big]^{T} + \boldsymbol{Q}.\end{aligned} \tag{2.106}$$

Measurement update equations:

$$\begin{aligned}\boldsymbol{P}_{y_k y_k}^{j} &= \boldsymbol{C}\boldsymbol{P}_{k,k-1}^{j}\boldsymbol{C}^{T} + \boldsymbol{R}, \\ \boldsymbol{K}_{k}^{j} &= \boldsymbol{P}_{k,k-1}^{j}\boldsymbol{C}^{T}\boldsymbol{P}_{y_k y_k}^{-1\,(j)}, \\ \hat{\boldsymbol{y}}_{k,k-1}^{j} &= \boldsymbol{C}\hat{\boldsymbol{x}}_{k,k-1}^{j},\end{aligned} \tag{2.107, 2.108}$$

$$z_k^j = y_k - \hat{y}_{k,k-1}^j, \tag{2.109}$$

$$\hat{x}_{k|k}^j = \hat{x}_{k,k-1}^j + K_k^j z_k^j \tag{2.110}$$

$$P_{k|k}^j = [I_n - K_k C]\, P_{k|k-1}^j. \tag{2.111}$$

Step 3: Model likelihood computations

The likelihood of the jth model is computed with the filter residuals z_k^j, the covariance of the filter residuals $P_{y_k y_k}^j$ and the assumption of the Gaussian statistics. The likelihood of the jth model and model probabilities update are as follows:

$$\Lambda_k^j = \frac{1}{\sqrt{|2\pi P_{y_k y_k}^j|}} \exp\big[-0.5(z_k^j)^T (P_{y_k y_k}^j)^{-1} z_k^j\big],$$

$$c = \sum_{i=1}^{N} \Lambda_k^i \bar{c}_i,$$

$$\mu_k^j = \frac{1}{c} \Lambda_k^j \bar{c}_j.$$

Step 4: Combination of state estimates

The state estimate $\hat{x}_{k|k}$ and the covariance $P_{k|k}$ for the IMM filter are obtained from a probabilistic sum of the individual filter outputs,

$$\hat{x}_{k|k} = \sum_{i=1}^{N} \hat{x}_{k|k}^i \mu_k^i,$$

$$P_{k|k} = \sum_{i=1}^{N} \mu_k^i \big[P_{k|k}^i + (\hat{x}_{k|k}^i - \hat{x}_{k|k})(\hat{x}_{k|k}^i - \hat{x}_{k|k})^T\big].$$

2.7 Illustrative Examples

The objective of the subsequent part of this section is to examine the proposed approaches with two sample systems, i.e., an induction motor and a two-tank system. In particular, the way of determining unknown input distribution matrix and the "switching" of these matrices will be illustrated with an induction motor. The two-tank system will be employed to show the performance of the proposed approach with respect to fault detection and isolation. The final part of this section shows a comparison between the first- and the second-order EUIO. An empirical comparison is realised with the model of an induction motor.

2.7.1 Estimation of E for an Induction Motor

The purpose of this section is to show the reliability and effectiveness of the proposed EUIO. The numerical example considered here is a fifth-order two-phase non-linear model of an induction motor, which has already been the subject of a large number of various control design applications (see Ref. [22] and the references therein).The complete discrete-time model in a stator-fixed (a, b) reference frame is

$$x_{1,k+1} = x_{1,k} + h\left(-\gamma x_{1k} + \frac{K}{T_r}x_{3k} + Kpx_{5k}x_{4k} + \frac{1}{\sigma L_s}u_{1k}\right), \tag{2.112}$$

$$x_{2,k+1} = x_{2,k} + h\left(-\gamma x_{2k} - Kpx_{5k}x_{3k} + \frac{K}{T_r}x_{4k} + \frac{1}{\sigma L_s}u_{2k}\right), \tag{2.113}$$

$$x_{3,k+1} = x_{3,k} + h\left(\frac{M}{T_r}x_{1k} - \frac{1}{T_r}x_{3k} - px_{5k}x_{4k}\right), \tag{2.114}$$

$$x_{4,k+1} = x_{4,k} + h\left(\frac{M}{T_r}x_{2k} + px_{5k}x_{3k} - \frac{1}{T_r}x_{4k}\right), \tag{2.115}$$

$$x_{5,k+1} = x_{5,k} + h\left(\frac{pM}{JL_r}(x_{3k}x_{2k} - x_{4k}x_{1k}) - \frac{T_L}{J}\right), \tag{2.116}$$

$$y_{1,k+1} = x_{1,k+1},\ y_{2,k+1} = x_{2,k+1}, \tag{2.117}$$

where $\boldsymbol{x}_k = [x_{1,k}, \ldots, x_{n,k}]^T = [i_{\mathrm{sak}}, i_{\mathrm{sbk}}, \psi_{\mathrm{rak}}, \psi_{\mathrm{rbk}}, \omega_k]^T$ represents the currents, the rotor fluxes, and the angular speed, respectively, while $\boldsymbol{u}_k = [u_{\mathrm{sak}}, u_{\mathrm{sbk}}]^T$ is the stator voltage control vector, p is the number of the pairs of poles, and T_L is the load torque. The rotor time constant T_r and the remaining parameters are defined as

$$T_r = \frac{L_r}{R_r},\ \sigma = 1 - \frac{M^2}{L_s L_r},\ K = \frac{M}{\sigma L_s L_r},\ \gamma = \frac{R_s}{\sigma L_s} + \frac{R_r M^2}{\sigma L_s L_r^2}, \tag{2.118}$$

where R_s, R_r and L_s, L_r are stator and rotor per-phase resistances and inductances, respectively, and J is the rotor moment inertia.

The numerical values of the above parameters are as follows: $R_s = 0.18\ \Omega$, $R_r = 0.15\ \Omega$, $M = 0.068$ H, $L_s = 0.0699$ H, $L_r = 0.0699$ H, $J = 0.0586\ \mathrm{kgm}^2$, $T_L = 10$ Nm, $p = 1$, and $h = 0.1$ ms. The input signals are

$$u_{1,k} = 350\cos(0.03k), \quad u_{2,k} = 300\sin(0.03k). \tag{2.119}$$

Let us assume that the unknown input and its distribution matrix have the following form:

$$\boldsymbol{E} = [1.2, 0.2, 2.4, 1, 1.6]^T, \tag{2.120}$$

$$\boldsymbol{d}_k = 0.3\sin(0.5\pi k)\cos(0.03\pi k), \tag{2.121}$$

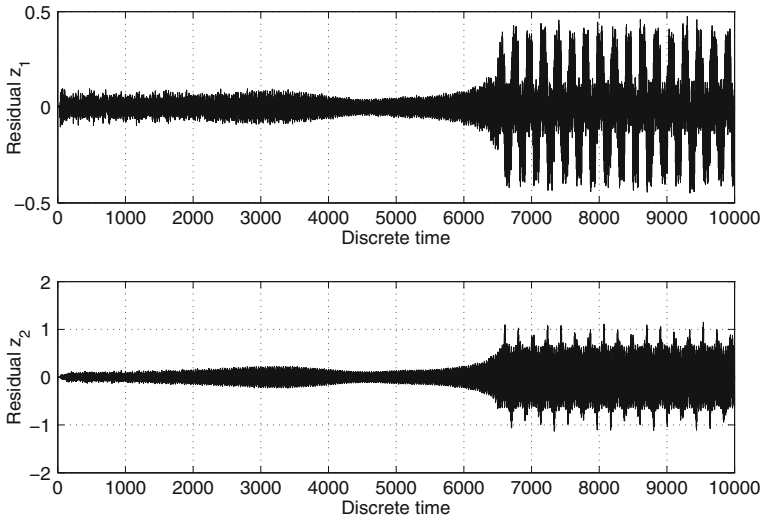

Fig. 2.3 Residuals for a randomly selected $\boldsymbol{E}$

while the noise covariance matrices are $\boldsymbol{Q} = 10^{-5}\boldsymbol{I}$ and $\boldsymbol{R} = 10^{-5}\boldsymbol{I}$. Note that the small values of the process and measurement noise are selected in order to clearly portray the effect of an unknown input. Figure 2.3 shows the residual z_k for randomly selected $\boldsymbol{E}$. From these results, it is evident that the estimation quality is very low and hence the residual is significantly different from zero, which may lead to a decrease in the fault detection abilities.

In order to prevent such a situation, the algorithm presented in Sect. 2.5 was utilised with the following settings:

- $\sigma^{(1)}$: The initial standard deviation,
- $j_{\max} = 20$: The number of iterations in each phase,
- $i_{\max} = 5$: The number of standard deviations (σ^i) changes,
- $k_{\max} = 50$,
- $\boldsymbol{E}^{(0)}$: Randomly selected.

The performance of the algorithm was tested for a set of $\sigma^{(1)}$, i.e., $\{1, 2, 3, 4, 5\}$. Note that $k_{\max} = 50$, which means that each run of the algorithm was performed 50 times. As a result, the mean and the standard deviation of the resulting $J(\boldsymbol{E})$ (cf. (2.92)) for each setting of $\sigma^{(1)}$ was calculated. The mean of $J(\boldsymbol{E})$ is presented in Fig. 2.4, while its standard deviation is portrayed in Fig. 2.5.

From these results, it is evident that the smallest mean and standard deviation are obtained for $\sigma^{(1)} = 3$. This, of course, does not mean that this is a particular value $\sigma^{(1)} = 3$, which should be the best one for each example. However, it can be easily observed that, for other $\sigma^{(1)}$, i.e., $\{1, 2, 4, 5\}$, the mean and standard deviation are

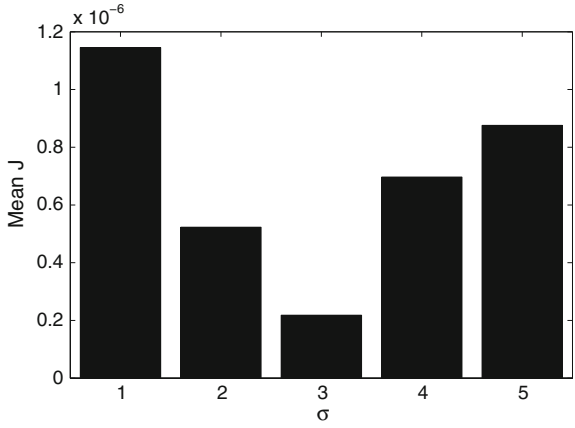

Fig. 2.4 Mean of $J(\boldsymbol{E})$ for $\sigma^{(1)} = 1, \ldots, 5$

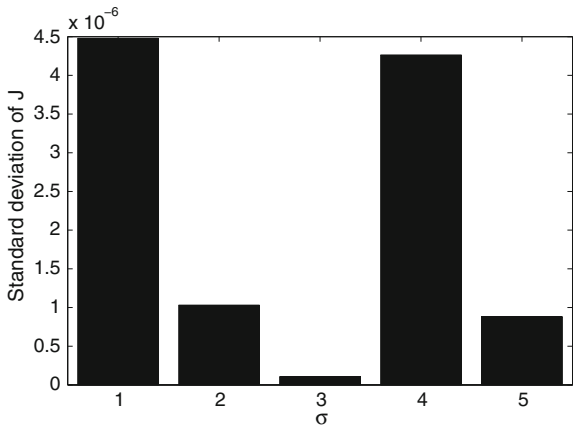

Fig. 2.5 Standard deviation of $J(\boldsymbol{E})$ for $\sigma^{(1)} = 1, \ldots, 5$

also very small. Numerous numerical experiments confirm this property, i.e., this means that the proposed algorithm is not extremely sensitive to the initial value of $\sigma^{(1)}$.

As mentioned in the preceding part of the chapter, the matrix $\boldsymbol{E}$, which is able to decouple the unknown input, is not unique. Indeed, the estimate of $\boldsymbol{E}$, for which $J(\boldsymbol{E})$ reaches its minimum, is

$$\hat{\boldsymbol{E}} = [0.3651, 0.0609, 0.7303, 0.3043, 0.4869]^T. \tag{2.122}$$

Figure 2.6 presents the residual for the obtained estimate. A direct comparison of Figs. 2.3 and 2.6 clearly shows the profits that can be gained while using the proposed algorithm.

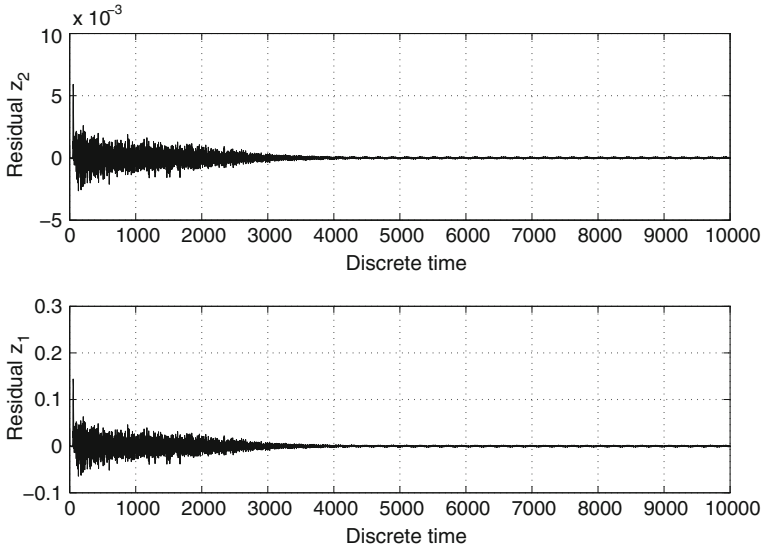

Fig. 2.6 Residuals for the estimated $\boldsymbol{E}$

2.7.2 Varying E Case

Let us reconsider an example presented in the previous section. The unknown input is defined, as previously, by (2.121), but three different settings of the unknown input distribution matrix $\boldsymbol{E}^j$ were employed during system simulation (the simulation time was 10,000 samples):

$$\begin{aligned}
\boldsymbol{E}^1 &= [1.2, 0.2, 2.4, 1, 1.6]^T \text{ for } 0 \leq k < 2,500, \\
\boldsymbol{E}^2 &= [0.2, 1.2, 2.4, 1, 1.6]^T \text{ for } 2,500 \leq k < 5,000 \\
&\quad \text{and } 7,500 \leq k < 10,000, \\
\boldsymbol{E}^3 &= [2.1, 2.1, 2.1, 2.1, 2.1]^T \text{ for } 5,000 \leq k < 7,500.
\end{aligned}$$

Contrary to the above-described simulation scenario, it was assumed that the set of unknown input distribution matrices for the UIF is composed of

$$\begin{aligned}
\boldsymbol{E}^1 &= [0.2, 1.2, 2.4, 1, 1.6]^T, \\
\boldsymbol{E}^2 &= [0, 0.2, 2.4, 1, 0]^T, \\
\boldsymbol{E}^3 &= [2.1, 2.1, 2.1, 2.1, 2.1]^T, \\
\boldsymbol{E}^4 &= [1, 2, 3, 1, 0]^T, \\
\boldsymbol{E}^5 &= [1.2, 0.2, 2.4, 1, 1.6]^T.
\end{aligned}$$

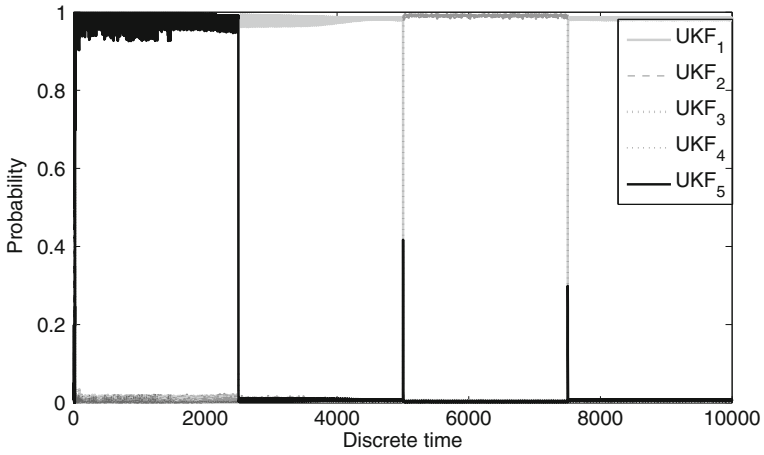

Fig. 2.7 Model probabilities

This means that $\boldsymbol{E}^2$ and $\boldsymbol{E}^4$ should not be used by the UIF, while $\boldsymbol{E}^1$, $\boldsymbol{E}^3$ and $\boldsymbol{E}^5$ should be appropriately switched.

Figure 2.7 shows model probabilities corresponding to the five unknown input distribution matrices. From these results, it is evident that the instrumental matrices $\boldsymbol{E}^1$, $\boldsymbol{E}^3$ and $\boldsymbol{E}^5$ were switched correctly. Moreover, probabilities corresponding to $\boldsymbol{E}^2$ and $\boldsymbol{E}^4$ are very low.

2.7.3 Fault Detection and Isolation of a Two-Tank System

The main objective of this section is to show that the UIF can also be effectively applied for fault detection and isolation. In this case, the unknown input is suitably used for designing the required fault isolation performance. The system considered consists of two cylindrical tanks of the same diameter. They are linked to each other through a connecting cylindrical pipe (Fig. 2.8). The two-tank system can be perceived as a Single-Input Multi-Output (SIMO) system, where the input u is the water flow through the pump, while the outputs y_1 and y_2 are water levels in the first and the second tank, respectively.

It is assumed that the system considered can be affected by the following set of faults:

actuator fault: f_1 pump lost-of-effectiveness or leakage from the pump pipe,
process faults: f_2 clogged connecting cylindrical pipe,
sensor faults: f_3 water level sensor fault of the first tank, f_4 water level sensor fault of the second tank.

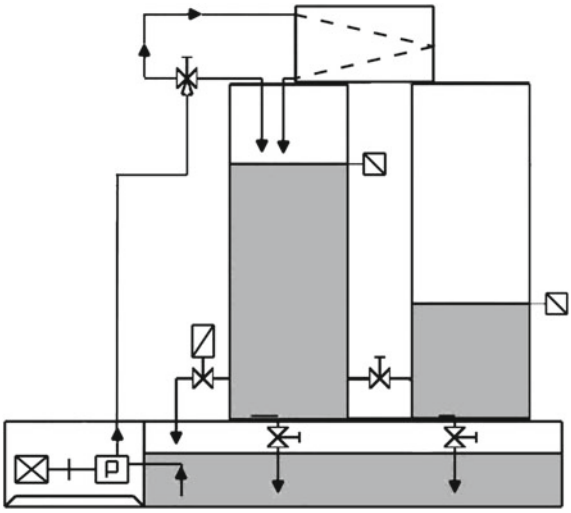

Fig. 2.8 Schematic diagram of a two-tank system

Once the fault description is provided, then a complete system description can be given as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{h}(\boldsymbol{u}_k) + \boldsymbol{L}_1 \boldsymbol{f}_{a,k} + \boldsymbol{w}_k, \tag{2.123}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{C}\boldsymbol{x}_{k+1} + \boldsymbol{L}_2 \boldsymbol{f}_{s,k+1} + \boldsymbol{v}_{k+1}, \tag{2.124}$$

where

$$\boldsymbol{g}(\boldsymbol{x}_k) = \begin{bmatrix} -h\frac{K_1}{A_1}\sqrt{x_{1,k} - x_{2,k}} + x_{1,k} \\ h\frac{K_1}{A_2}\sqrt{x_{1,k} - x_{2,k}} - h\frac{K_2}{A_2}\sqrt{x_{2,k}} + x_{2,k} \end{bmatrix}, \tag{2.125}$$

$$\boldsymbol{h}(\boldsymbol{u}_k) = \left[h\frac{1}{A_1}u_k, 0 \right]^T, \tag{2.126}$$

$$\boldsymbol{L}_1 = \begin{bmatrix} -\frac{h}{A_1} & \frac{h}{A_1} \\ 0 & \frac{-h}{A_2} \end{bmatrix}, \tag{2.127}$$

$$\boldsymbol{L}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{C} = \boldsymbol{I}, \tag{2.128}$$

$$\boldsymbol{f}_{a,k} = \left[f_{1,k}, \sqrt{x_{1,k} - x_{2,k}} f_{2,k} \right]^T, \quad \boldsymbol{f}_{s,k} = \left[f_{3,k}, f_{4,k} \right]^T,$$

where $x_{1,k}$ and $x_{2,k}$ are water levels in the first and the second tank, respectively, A_1, A_2 stand for the cross-sections of the tanks, K_1 denotes the cross-section of the

connecting pipe, K_2 is the cross-section of the outflow pipe from the second tank, and h is the sampling time.

The objective of the subsequent part of this section is to design UIF-based diagnostic filter which will make it possible to detect and isolate the above mentioned faults.

Filter 1: In order to make the residual insensitive to f_1, it is proposed to use the developed UIF with the following settings:

$$\boldsymbol{E} = \boldsymbol{L}_1^1, \quad \boldsymbol{d}_k = f_{1,k}, \quad \boldsymbol{L} = \boldsymbol{L}_1^2, \quad \boldsymbol{f}_k = f_{2,k}, \quad \boldsymbol{C}_k = [1,\ 0], \tag{2.129}$$

where $\boldsymbol{L}_1^i$ stands for the ith column of $\boldsymbol{L}_1$. It is straightforward to examine that the conditions (2.87) and (2.90) are satisfied, which means that the observer while remaining insensitive to $f_{1,k}$ while it will remain sensitive to $f_{2,k}$.

Filter 2: Similarly as in the *Filter 1* case, the residual generated by the *Filter 2* should be insensitive to $f_{2,k}$,

$$\boldsymbol{E} = \boldsymbol{L}_1^2, \quad \boldsymbol{d}_k = f_{2,k}, \quad \boldsymbol{L} = \boldsymbol{L}_1^1, \quad \boldsymbol{f}_k = f_{1,k}, \quad \boldsymbol{C}_k = [1,\ 0]. \tag{2.130}$$

It is straightforward to examine that conditions (2.87) and (2.90) are satisfied, which means that the observer will be insensitive to $f_{2,k}$ while while remaining sensitive to $f_{1,k}$.

Filter 3: The filter should be insensitive to $f_{3,k}$ while sensitive to $f_{4,k}$. This can be realised using the conventional UKF with

$$\boldsymbol{C} = [0,\ 1]. \tag{2.131}$$

Filter 4: The filter should be insensitive to $f_{4,k}$ while it should be sensitive to $f_{3,k}$. This can be realised using the conventional UKF with

$$\boldsymbol{C} = [1,\ 0], \tag{2.132}$$

The main objective of this section is to show the testing results obtained with the proposed approach. To tackle this problem, a Matlab-based simulator of a two-tank system was implemented. The simulator is able to generate the data for normal as well as for all faulty conditions $(f_1, \ldots, f_4)$ being considered. The filter-based fault diagnosis scheme was also implemented using Matlab. As a result, a complete scheme that is able to validate the performance of the proposed fault diagnosis strategy was developed. It should be also pointed out that the simulations were carried out using the following numerical parameters: $u_k = 2.56$, $h = 0.1$, $A_1 = 4.2929$, $A_2 = 4.2929$, $K_1 = 0.3646$, $K_2 = 0.2524$.

All fault scenarios where generated according to the following rule:

$$f_{i,k} = \begin{cases} \neq 0 & k = 300, \ldots, 400 \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \ldots, 4.$$

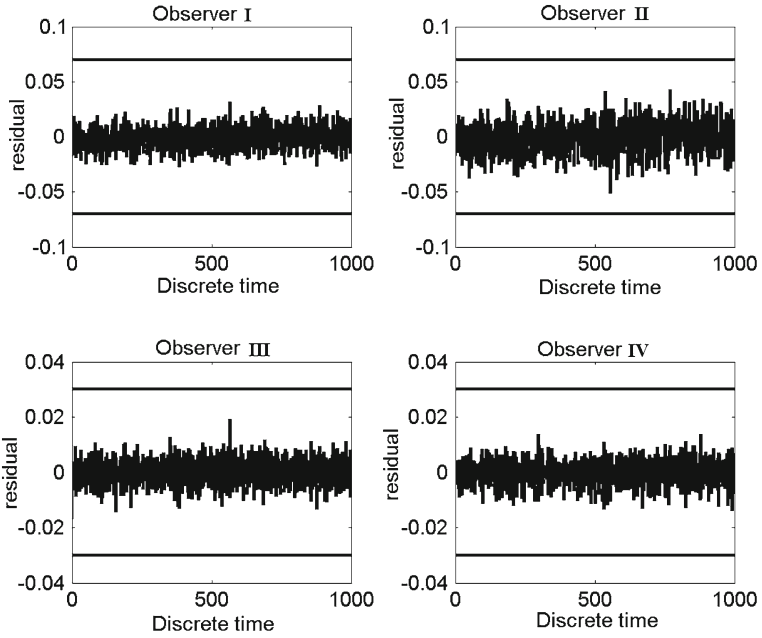

Fig. 2.9 Residuals for the fault-free case

Moreover, y_1 and y_2 were corrupted by measurement noise generated according to the normal distribution, i.e., $\mathcal{N}(\mathbf{0}, \text{diag}(0.01, 0.01))$. Thus, the following settings of the instrumental matrices were employed: $\boldsymbol{R} = 0.1\boldsymbol{I}$ and $\boldsymbol{Q} = 0.1\boldsymbol{I}$.

Figure 2.9 portrays the residual obtained with the four filters for the fault-free case. As can be observed, all of them are very close to zero.

Figures 2.10, 2.11, 2.12 and 2.13 present the residuals for the faults f_1 to f_4 obtained with the four filters.

The results are summarised in the form of a diagnostic table presented as Table 2.1.

It should be noticed that the residuals generated by *Filter 3* and *Filter 4* are insensitive to f_1 and f_2. Such a situation is caused by the fact that observers use feedback from the system output and hence some damping effects may arise. This is the case in the presented situation. On the other hand, it was observed that the results of experiments can be consistent with the theoretical expectations when there is no measurement noise, but this is a rather unreal situation. Irrespective of the presented results, the faults can still be isolated because they have unique signatures.

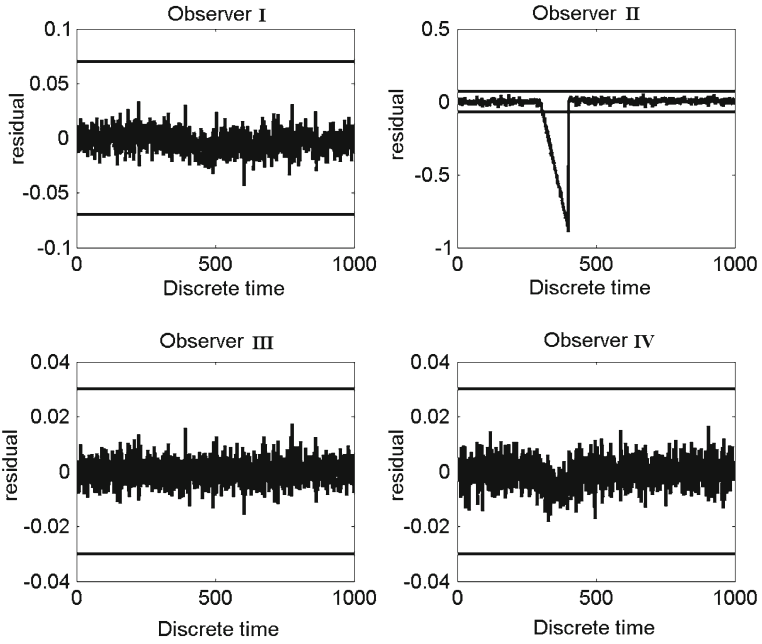

Fig. 2.10 Residuals for the fault f_1

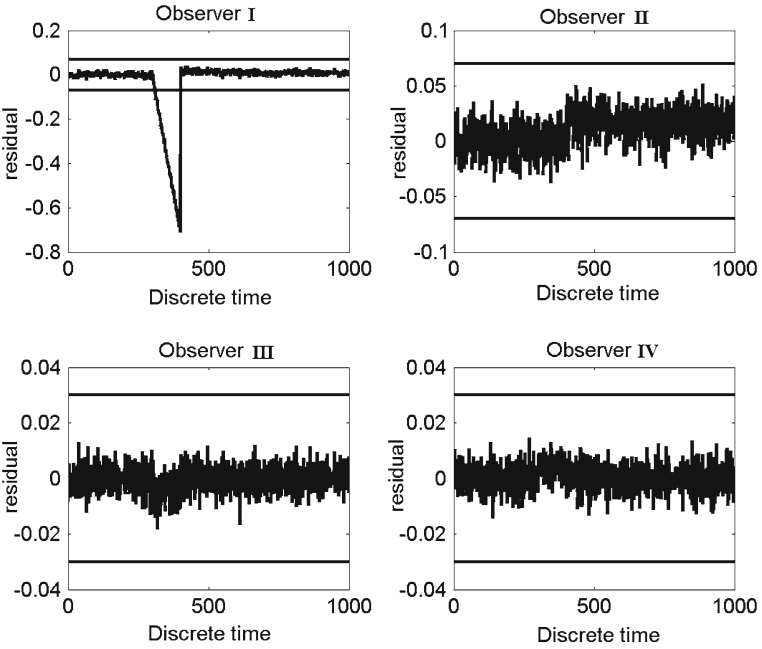

Fig. 2.11 Residuals for the fault f_2

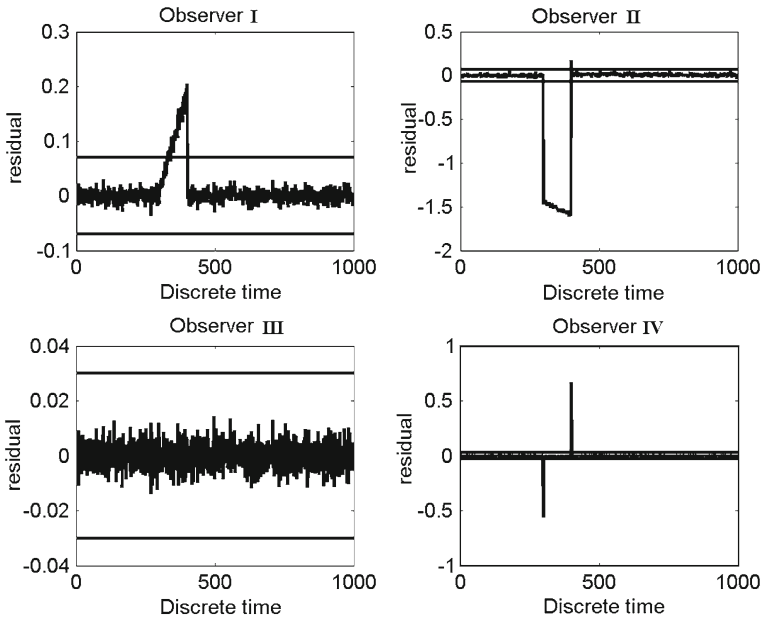

Fig. 2.12 Residuals for the fault f_3

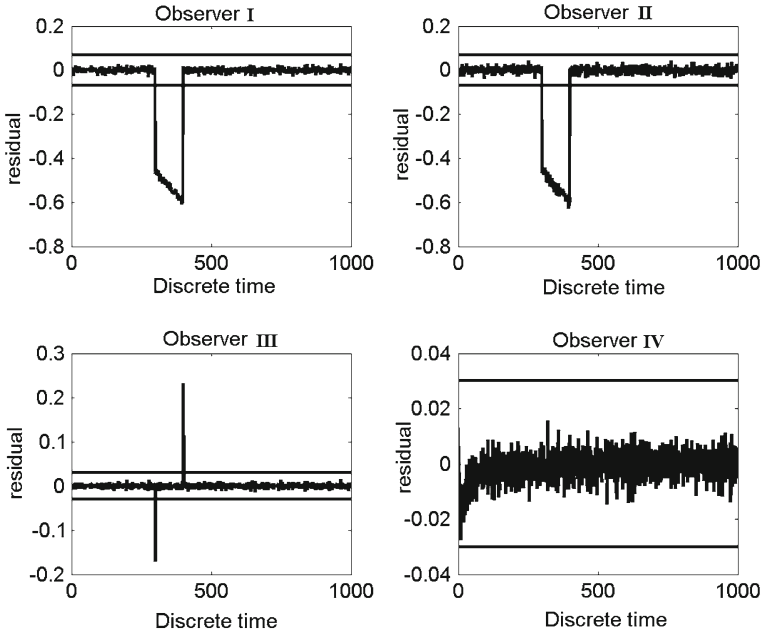

Fig. 2.13 Residuals for the fault f_4

Table 2.1 Diagnostic table

Filter	f_1	f_2	f_3	f_4
Filter 1	0	1	1	1
Filter 2	1	0	1	1
Filter 3	0	0	0	1
Filter 4	0	0	1	0

2.7.4 First- Versus Second-order EUIO

The main objective of this section is to perform a comprehensive study regarding the first- and the second-order EUIO.

Let us reconsider the induction motor described by (2.112)–(2.117). Let $\mathbb{X}$ be a bounded set denoting the space of the possible variations of the initial condition $\boldsymbol{x}_0$:

$$\begin{aligned}\mathbb{X} = \{&[-276, 279] \times [-243, 369] \times S_{15}^{(2)}(\mathbf{0}) \\ &\times [-11, 56]\} \subset \mathbb{R}^5,\end{aligned} \tag{2.133}$$

where $S_r^{(n)}(\boldsymbol{c}) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{c}\|_2 \leq r\}$, $r = 15$. Let us assume that each initial condition of the system $\boldsymbol{x}_0$ is equally probable, i.e.

$$pr(\boldsymbol{x}_0) = \begin{cases} \frac{1}{m(\mathbb{X})} & \text{for } \boldsymbol{x}_0 \in \mathbb{X}, \\ 0 & \text{otherwise,} \end{cases}$$

where $m(\mathbb{A})$ is the Lebesgue measure of the set $\mathbb{A}$. Moreover, the following three observer configurations were considered:

Case 1: First-order EUIO with:

$$\begin{aligned}\boldsymbol{R}_k &= 0.1\boldsymbol{I}_2, \\ \boldsymbol{Q}_k &= 0.1\boldsymbol{I}_5.\end{aligned}$$

Case 2: First-order EUIO with:

$$\begin{aligned}\boldsymbol{R}_k &= 0.1\boldsymbol{I}_2, \\ \boldsymbol{Q}_k &= 10^3\varepsilon_k^T\varepsilon_k\boldsymbol{I}_5 + 0.001\boldsymbol{I}_5.\end{aligned}$$

Case 3: Second-order EUIO with:

$$\begin{aligned}\boldsymbol{R}_k &= 0.1\boldsymbol{I}_2, \\ \boldsymbol{Q}_k &= 0.1\boldsymbol{I}_5.\end{aligned}$$

In order to validate the performance of the observers, each of them was run for $N = 1000$ randomly selected initial conditions $\boldsymbol{x}_0 \in \mathbb{X}$ and then the following quality index was calculated:

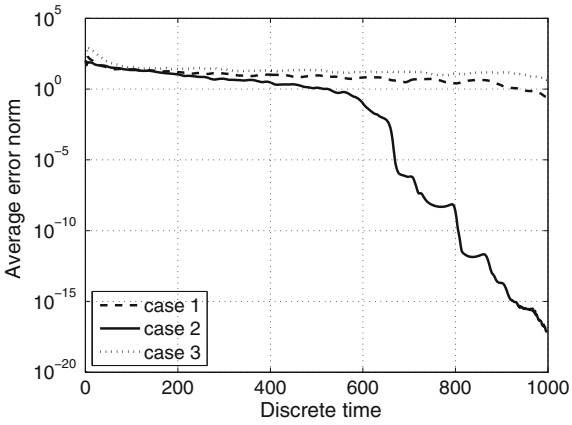

Fig. 2.14 Average norm of the state estimation error

$$Q = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\boldsymbol{e}_{n_t}^i\|_2}{\|\boldsymbol{e}_0^i\|_2}, \tag{2.134}$$

where $n_t = 1000$. This quality index describes an average ration between the norm of the initial state estimation error $\boldsymbol{e}_0$ and the norm of the state estimation estimation error achieved after n_t iterations. The following results were obtained:

Case 1: $Q = 0.0023$,
Case 2: $Q = 9.6935 \cdot 10^{-17}$,
Case 3: $Q = 0.0385$.

It is clear that the observer of Case 2 provides the best results. This can also be observed in Fig. 2.14, which presents the evolution of an average norm of the state estimation error. The main reason why the second-order EUIO does not provide good results is that it is very sensitive to the initial state estimation error and to the initial value of $\boldsymbol{P}_k$. This follows from the fact that $\boldsymbol{s}_k$ is calculated using the approximation (2.38) instead of an exact form:

$$s_{i,k} = \frac{1}{2}\left[\boldsymbol{e}_k^T \left.\frac{\partial \bar{\boldsymbol{g}}_i(\boldsymbol{x}_k)^2}{\partial \boldsymbol{x}_k^2}\right|_{\boldsymbol{x}_k=\hat{\boldsymbol{x}}_k} \boldsymbol{e}_k\right], \quad i = 1, \ldots, n.$$

2.8 Concluding Remarks

The main objective of this chapter was to present three different approaches that can be used for designing unknown input observers and filters for non-linear discrete-time systems. In particular, a system description was provided, which covers a large class of non-linear systems, and a general rule for decoupling the unknown input

was portrayed. The chapter provides a rule for checking if the observer/filter will not decouple the effect of a fault. This unappealing phenomenon may lead to undetected faults, which may have a serious impact on the performance of the system being controlled and diagnosed. Subsequently, an approach for designing the so-called first- and second-order extended unknown input observer was provided. The proposed approach is based on a general extended Kalman filter framework and can be applied for non-linear deterministic systems. It was shown, with the help of the Lyapunov method, that such a linearisation-based technique is convergent under certain conditions. To tackle this task, a novel structure and design procedure of the EUIO were proposed. Another approach was proposed for non-linear stochastic framework, bearing in mind that Ref. [16] *it is easier to approximate a probability distribution than it is to approximate an arbitrary non-linear function or transformation.* The unscented Kalman filter formed a base for the development of an unknown input filter. Based on the UIF, an algorithm for estimating unknown input distribution matrix was proposed. Another important contribution of this chapter was the development of the UIF that is able to switch the unknown input distribution matrices according to the working conditions. This task was realised with the interacting multiple model algorithm. The final part of the chapter presented comprehensive case studies regarding practical application of the proposed approaches. These examples are an induction motor and a two-tank systems. In particular, based on the example with the induction motor, the strategies for determining the unknown input distribution matrix and the case with a set of predefined unknown input distribution matrices were examined. The same example (within a deterministic framework) was utilised to perform a comparison between the first- and second-order extended unknown input observer. The abilities regarding fault detection and isolation were illustrated with the two-tank system. In all the cases, the proposed approaches exhibit their practical usefulness.

References

1. J. Korbicz, J. Kościelny, Z. Kowalczuk, W. Cholewa (eds.), *Fault Diagnosis. Models, Artificial Intelligence, Applications* (Springer-Verlag, Berlin, 2004)
2. M. Witczak, *Modelling and Estimation Strategies for Fault Diagnosis of Non-linear Systems* (Springer-Verlag, Berlin, 2007)
3. S.X. Ding, *Model-Based Fault Diagnosis Techniques: Design Schemes. Algorithms and Tools* (Springer-Verlag, Berlin, 2008)
4. V. Puig, Fault diagnosis and fault tolerant control using set-membership approaches: application to real case studies. Int. J. Appl. Math. Comput. Sci. **20**(4), 619–635 (2010)
5. S. Tong, G. Yang, W. Zhang, Observer-based fault-tolerant control againts sensor failures for fuzzy systems with time delays. Int. J. Appl. Math. Comput. Sci. **21**(4), 617–628 (2011)
6. K. Kemir, F. Ben Hmida, J. Ragot, M. Gossa, Novel optimal recursive filter for state and fault estimation of linear systems with unknown disturbances. Int. J. Appl. Math. Comput. Sci. **21**(4), 629–638 (2011)
7. P.M. Frank, T. Marcu, Diagnosis strategies and systems. principles, fuzzy and neural approaches, in *Intelligent Systems and Interfaces*, ed. by H.N. Teodorescu, D. Mlynek, A. Kandel, H.J. Zimmermann (Kluwer Academic Publishers, Boston, 2000)

8. S.H. Wang, E.J. Davison, P. Dorato, Observing the states of systems with unmeasurable disturbances. IEEE Trans. Autom. Control **20**(5), 716–717 (1975)
9. H. Hammouri, P. Kabore, S. Othman, J. Biston, Failure diagnosis and nonlinear observer. Application to a hydraulic process. J. Franklin Inst. **339**(4–5), 455–478 (2002)
10. H. Hammouri, P. Kabore, S. Othman, J. Biston, Observer-based approach to fault detection and isolation for nonlinear systems. IEEE Trans. Autom. Control **44**(10), 1879–1884 (1999)
11. R. Kabore, H. Wang, Design of fault diagnosis filters and fault tolerant control for a class of nonlinear systems. IEEE Trans. Autom. Control **46**(11), 1805–1809 (2001)
12. W. Chen, A.Q. Khan, M. Abid, S.X. Ding, Integrated design of observer-based fault detection for a class of uncertain non-linear systems. Int. J. Appl. Math. Comput. Sci. **21**(4), 619–636 (2011)
13. D. Koenig, S. Mammar, Design of a class of reduced unknown inputs non-linear observer for fault diagnosis, in *Proceedings of American Control Conference*, ACC, Arlington, USA, 2002
14. A.M. Pertew, H.J. Marquez, Q. Zhao, $\mathcal{H}_\infty$ synthesis of unknown input observers for non-linear lipschitz systems. Int. J. Control **78**(15), 1155–1165 (2005)
15. J. Chen, R.J. Patton, *Robust Model Based Fault Diagnosis for Dynamic Systems* (Kluwer Academic Publishers, London, 1999)
16. S.J. Julier, J.K. Uhlmann, Unscented filtering and estimation. Proc. IEEE **92**(3), 401–422 (2004)
17. R. Kandepu, B. Foss, L. Imsland, Applying the unscented Kalman filter for nonlinear state estimation. J. Process Control **18**(7–8), 753–768 (2008)
18. M. Witczak, P. Pretki, Design of an extended unknown input observer with stochastic robustness techniques and evolutionary algorithms. Int. J. Control **80**(5), 749–762 (2007)
19. F.J. Uppal, R.J. Patton, M. Witczak, A neuro-fuzzy multiple-model observer approach to robust fault diagnosis based on the DAMADICS benchmark problem. Control Eng. Pract. **14**(6), 699–717 (2006)
20. M. Witczak, J. Korbicz, R. Jozefowicz, Design of unknown input observers for non-linear stochastic systems and their application to robust fault diagnosis. Control Cybern. **42**(1), 227–256 (2013)
21. M. Witczak, P. Pretki, J. Korbicz, Design and convergence analysis of first- and second-order extended unknown input observers, in *Methods and Models in Automation and Robotics—MMAR 2007: Proceedings of the 13th IEEE/IFAC International Conference*, Szczecin, Poland, 2007, pp. 833–838 (CD-ROM)
22. M. Boutayeb, D. Aubry, A strong tracking extended Kalman observer for nonlinear discrete-time systems. IEEE Trans. Autom. Control **44**(8), 1550–1556 (1999)
23. L.Z. Guo, Q.M. Zhu, A fast convergent extended Kalman observer for non-linear discrete-time systems. Int. J. Syst. Sci. **33**(13), 1051–1058 (2002)
24. G. Ducard, *Fault-tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles* (Springer-Verlag, Berlin, 2009)
25. R. Iserman, *Fault Diagnosis Applications: Model Based Condition Monitoring, Actuators, Drives, Machinery, Plants, Sensors, and Fault-Tolerant Systems* (Springer-Verlag, Berlin, 2011)
26. M. Mahmoud, J. Jiang, Y. Zhang, *Active Fault Tolerant Control Systems: Stochastic Analysis and Synthesis* (Springer-Verlag, Berlin, 2003)
27. H. Noura, D. Theilliol, J. Ponsart, A. Chamseddine, *Fault-Tolerant Control Systems: Design and Practical Applications* (Springer-Verlag, Berlin, 2003)
28. E. Walter, L. Pronzato, *Identification of Parametric Models from Experimental Data* (Springer, London, 1996)
29. H.A.P. Blom, Y. Bar-Shalom, The interacting multiple model algorithm for systems with markovian switching coefficients. IEEE Trans. Autom. Control **33**(8), 780–783 (1988)

http://www.springer.com/978-3-319-03013-5

Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems
Analytical and Soft Computing Approaches
Witczak, M.
2014, XVI, 229 p. 117 illus., Hardcover
ISBN: 978-3-319-03013-5