Linköping Studies in Science and Technology. Dissertations
No. 716

Residual Generation for Fault Diagnosis

Erik Frisk

Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2001

Residual Generation for Fault Diagnosis

frisk@isy.liu.se
http://www.fs.isy.liu.se
Department of Electrical Engineering,
Linköping University,
SE–581 83 Linköping,
Sweden.

ISBN 91-7373-115-3 ISSN 0345-7524

Printed by Linus & Linnea AB, Linköping, Sweden 2001

Abstract

The objective when supervising technical processes is to alarm an operator when a fault is detected and also identify one, or possibly a set of components, that may have been the cause of the alarm. Diagnosis is an expansive subject, partly due to the fact that nowadays, more applications have more embedded computing power and more available sensors than before.

A fundamental part of many model-based diagnosis algorithms are so called *residuals*. A residual is a signal that reacts to a carefully chosen subset of the considered faults and by generating a suitable set of such residuals, fault detection and isolation can be achieved.

A common thread is the development of systematic design and analysis methods for residual generators based on a number of different model classes, namely deterministic and stochastic linear models on state-space, descriptor, or transfer function form, and non-linear polynomial systems. In addition, it is considered important that there exist readily available computer tools for all design algorithms.

A key result is the minimal polynomial basis algorithm that is used to parameterize all possible residual generators for linear model descriptions. It also, explicitly, finds those solutions of minimal order. The design process and its numerical properties are shown to be sound. The algorithms and its principles are extended to descriptor systems, stochastic systems, nonlinear polynomial systems, and uncertain linear systems. New results from these extensions include: increased robustness by introduction of a reference model, a new type of whitening filters for residual generation for stochastic systems both on state-space form and descriptor form, and means to handle algorithmic complexity for the non-linear design problem.

In conclusion, for the four classes of models studied, new methods have been developed. The methods fulfills requirements generation of all possible solutions, availability of computational tools, and numerical soundness. The methods also provide the diagnosis system designer with a set of tools with well specified and intuitive design freedom.

Acknowledgments

This work was performed at the department of Electrical Engineering, division of Vehicular Systems, Linköping University. I would like to thank my professor and supervisor, Lars Nielsen for letting me join his group and for our research discussions. I would also like to thank the staff at Vehicular Systems for creating a positive atmosphere.

Foremost, I would like to thank my colleague Mattias Nyberg for all our discussions, collaborative work, and for proofreading the manuscript. Dr. Didier Henrion is gratefully acknowledged for patiently answering my questions and writing some of the Matlab code used in this work. Marcus Klein, Mattias Krysander, and Niten Olofsson are also acknowledged for proofreading early versions of the manuscript.

This work has been supported by the Swedish Foundation for Strategic Research through the graduate school ECSEL (Excellence Center in Computer Science and Systems Engineering in Linköping).

Linköping, October 2001

Erik Frisk

Contents

1

Introduction

Modern processes use more and more embedded computers and sensors to, among other things, increase performance and introduce new functionality. At the same time, the sensors combined with on-line computing power provide means for on-line supervision of the process itself. In such more autonomous operation, it is of important to detect faults before the fault seriously affects system performance. Faults in a control loop are particularly important since feedback from a faulty sensor very quickly can result in instability causing a complete failure of the plant. Such faults might need to be detected within a few samples (Blanke et al., 1997). Therefore it is important that faults are detected during normal operation of the plant, without the need to perform certain tests to perform the diagnosis.

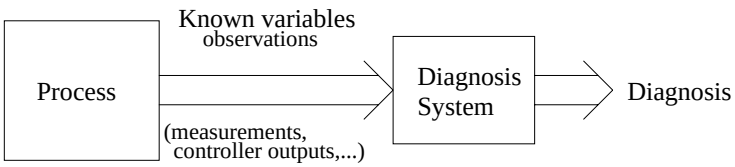


Figure 1.1: Diagnosis application

Here, the word diagnosis means detection and location (isolation) of a faulty component. A general structure of a technical diagnosis application is shown in Figure 1.1, where the diagnosis system is fed all available knowledge (also called observations) of the process. Such knowledge include measured variables,

controller outputs and any other information that is available to the diagnosis system. The diagnosis system processes the observations and produce a *diagnosis*, which is a list of possible faults affecting the process. Often the process is regulated by a controller and the known variables consist of controller outputs and sensor data. Such a situation is depicted in Figure 1.2 which also illustrates a fundamental complication the diagnosis system designer faces. Disturbances, also called unknown inputs, not considered faults also influence the process. The diagnosis system must thereby separate the influence caused by these unknown inputs and the faults.

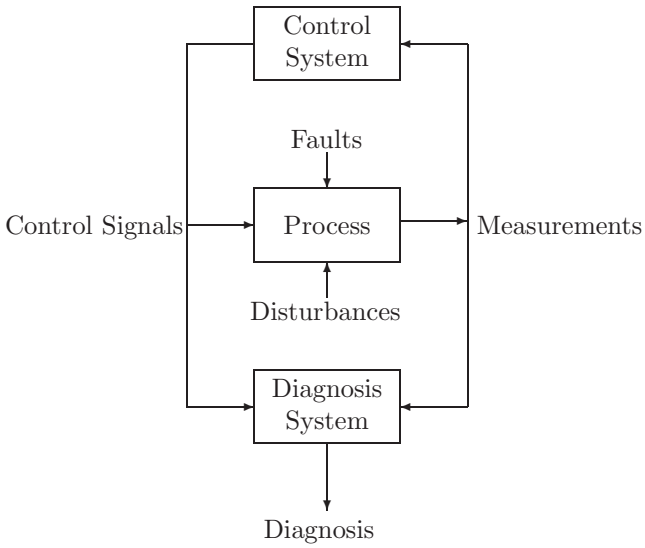


Figure 1.2: Control oriented diagnosis application.

To detect and isolate faulty components, some sort of *redundancy* is needed. The redundancy is used to make consistency checks between related variables. In applications with very high security demands such as aircraft control-systems, redundancy can be supplied by extra hardware, hardware redundancy. A critical component, for example a sensor, is then duplicated or triplicated and voting schemes can be used to monitor signal levels and trends to detect and locate faulty sensors. Hardware redundancy has the advantage of being reliable and gives high performance, but the approach has drawbacks, e.g. extra hardware costs, space and weight consideration, and some components can not be duplicated.

Instead of using hardware redundancy, *analytical* redundancy can be used where the redundancy is supplied by a process model instead of extra hardware. Then the process can be validated by comparing a measured variable with an

estimate, produced using the process model, of the same variable. Diagnosis systems based on analytical redundancy are also called *model based diagnosis* systems which is further described in Chapter 2.

1.1 Outline and contributions of the thesis

Often, a fundamental part of model based diagnosis systems is a residual generator. A residual is a computable quantity that is used to alarm if a fault is present in the supervised process or not. They can also, if designed properly, provide means for isolation of the faults. How to design residual generators for different model descriptions is the topic of this thesis.

Chapter 2 gives an overview of the model-based diagnosis problem. First, the diagnosis problem is defined and it is showed how residual generators fit into a complete supervision system, performing both fault detection and fault isolation. Then, residual generator design based on consistency relations[1] is described for linear and non-linear systems. Finally, the problems studied further in the chapters to follow are indicated and motivated.

In Chapter 3, residual generation for deterministic linear systems is presented. A key contribution is the minimal polynomial basis approach to residual generation. A main property of the algorithm is that it can, in a straightforward and numerically sound way, utilize models on transfer function form, state-space form, and also applies to a more general class of linear systems described by differential-algebraic equations, descriptor systems. The algorithm is thoroughly exemplified on a linearized aircraft model to show basic properties of the algorithm. A large, 24 state model of a jet engine is also included to show numerical properties of the approach.The theoretical parts are are mainly based on (Frisk and Nyberg, 2001) and the examples on (Frisk and Nyberg, 1999).

Chapter 4 continues with a derivation of a design procedure for non-linear systems, mainly based on (Frisk, 2000b). The algorithm has considerable similarities with the linear design procedure, and free design variables in the non-linear case has direct counterparts in the linear case. The algorithm has strong computational support in modern computer algebra systems like Mathematica and Maple. A major concern is the computational complexity of the design algorithm and it is shown how structural analysis of the model equations can be used to manage the complexity.

The linear design problem is revisited in Chapter 5, where stochastic linear model descriptions is considered. To systematically select the free design variables available after a deterministic design, additional modeling and additional constraints on the residual generators are needed which reduces the available design freedom. The design algorithm from Chapter 3, with all its merits on simplicity and numerical stability, is extended to the stochastic design problem. Finally, the design procedure is exemplified on both state-space and descriptor systems. This chapter is mainly based on (Frisk, 2001).

[1] Other commonly used words are parity or analytical redundancy relations

Chapter 6 provides an approach to make the residual generator as robust as possible to parametric uncertainties in the model description. An optimization procedure, based on $\mathcal{H}_\infty$-filtering theory, is used and a main contribution is the systematic procedure to form the the optimization criterion. A key observation is how a, at first sight natural, criterion can result in unnecessary poor performance of the residual generator. A systematic procedure, based on the nominal design problem, to form a feasible optimization criterion to synthesize residual generators is then outlined. The main objective with the procedure is to utilize the design freedom as much as possible to make the residual optimally robust against parametric uncertainties. The algorithm is mainly based on (Frisk and Nielsen, 1999).

Finally, Chapter 7 provides the conclusions.

1.2 Publications

In the research work, leading to this thesis, the author has published the following conference and journal papers (in reversed chronological order):

- E. Frisk (2001). Residual generation in linear stochastic systems - a polynomial approach. To appear in proc. IEEE Conf. on Decision and Control, Orlando, USA.

- E. Frisk and M. Nyberg (2001). A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems. *Automatica* 37(9), September, pp. 1417–1424.

- I. Andersson and E. Frisk (2001). Diagnosis of evaporative leaks and sensor faults in a vehicle fuel system. In proc. IFAC Workshop: advances in automotive control, Karlsruhe, Germany.

- E. Frisk (2000a). Order of residual generators - bounds and algorithms. In proc. IFAC Safeprocess, Budapest, Hungary, pp. 599–604.

- E. Frisk (2000b). Residual generator design for non-linear, polynomial systems – a Gröbner basis approach. In proc. IFAC Safeprocess, Budapest, Hungary, pp. 979–984.

- E. Frisk and M. Nyberg (1999). Using minimal polynomial bases for fault diagnosis. In proc. European Control Conference, Karlsruhe, Germany.[2]

- E. Frisk and L. Nielsen (1999). Robust residual generation for diagnosis including a reference model for residual behavior. In proc. IFAC World Congress, Beijing, P.R. China, Vol. P, pp. 55–60.

[2] Awarded the Polyx prize for best paper on polynomial methods in 1999.

- M. Nyberg and E. Frisk (1999). A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems. In proc. IFAC World Congress, Beijing, P.R. China, Vol. P, pp. 61–66.
- E. Frisk, M. Nyberg, and L. Nielsen (1997). FDI with adaptive residual generation applied to a DC-servo. In proc. IFAC Safeprocess, Hull, United Kingdom, pp. 438–442.

2

Model Based Diagnosis

The aim of this chapter is twofold, first to give a brief introduction to the field of model based diagnosis and second to introduce problem formulations and notation. It is intended to form a, both notational and conceptual, basis for the chapters to follow, not to give a complete view of the field.

Section 2.1 provides an introduction to model based diagnosis so that the subsections of Section 2.1 give brief presentations of the concepts and subsystems of residual-based diagnosis systems. This is done by first introducing an important concept, analytical redundancy. Then, in Subsection 2.1.1 fault models are described. The introductory presentation proceeds by discussing topics of central importance in this work, residuals and residual generators, in Subsection 2.1.2. The basics of fault isolation is described in Subsection 2.1.3 and finally, Subsection 2.1.4 describes how residual generators fit into a complete model-based diagnosis system. The introduction so far does not mention how to design and implement the residual generators. This topic is approached in Section 2.2 by exploring consistency relations, a concept that is central in the chapters to follow. Finally, in Section 2.3 problems studied in the remaining parts of the dissertation is discussed and motivated.

2.1 Introduction to model based diagnosis

Model based diagnosis methods has been developed for many model domains, e.g. models from the AI-field which are often logic based (Hamscher et al., 1992), or Discrete Event Dynamic Systems for which automata descriptions are common (Larsson, 1999; Sampath et al., 1995, 1996). A third model domain

that is commonly considered are models typically found in the field of signals and systems, i.e. models involving continuous variables in continuous or discrete time. Typical model formulations are differential/difference equations, transfer functions, and/or static relations. From now on in this work, only models from this domain are used.

In Chapter 1 it was discussed how model-based diagnosis is used when redundancy is supplied by a model instead of additional hardware. Redundancy supplied by a model is called analytical redundancy and can be defined more formally as:

Definition 2.1 (Analytical Redundancy). *There exists analytical redundancy if there exists two or more different ways to determine a variable x by only using the observations $z(t)$, i.e. $x = f_1(z(t))$ and $x = f_2(z(t))$, and $f_1(z(t)) \not\equiv f_2(z(t))$.*

Thus, the existence of analytical redundancy makes it possible to check the validity of the assumptions made to ensure that $f_1(z(t)) = f_2(z(t))$.

Example 2.1
Assume two sensors measure the variable x according to

$$y_1 = \sqrt{x} \quad \wedge \quad y_2 = x$$

The integrity of the two sensors can then be validated by ensuring that the relation, represented by the equation $y_1^2 - y_2 = 0$, holds.

In Example 2.1 it was easy to see that a malfunction in any of the two sensors would invalidate the relation and a fault could be detected. In more general cases, and to facilitate not only fault detection but also fault isolation, there is a need to describe fault influence on the process more formally, i.e. fault models of some sort is needed.

2.1.1 Fault modeling

A fault model is a formal representation of the knowledge of possible faults and how they influence the process. More specific, the term *fault* means that component behavior has deviated from its normal behavior. It does not mean that the component has stopped working altogether. The situation where a component has stopped working is, in the diagnosis community, called a *failure*. So, one goal is to detect faults before they cause a failure. In general, utilizing better fault models (assuming good and valid fault models) implies better diagnosis performance, i.e. smaller faults can be detected and more different types of faults can be isolated. Here fault modeling is illustrated using an example. For more elaborate discussions on fault modeling the reader is referred to e.g. (Nyberg, 1999b; Gertler, 1998; Chen and Patton, 1999).

A common fault model is to model faults as deviations of constant parameters from their nominal value. Typical faults that are modeled in this way are

"gain-errors" and "biases" in sensors, process faults modeled as a deviation of a physical parameter. In cases with constant parameter fault models, methods and theory from parameter estimation have shown useful also for fault diagnosis, see for example (Isermann, 1993). However, other more elaborate fault models exists e.g. fault models that utilizes the change-time characteristic of the process(Basseville and Nikiforov, 1993).

The fault models used in the chapters to follow are typically time-varying fault signals or constant parameter changes. An advantage with using fault-signals when modeling faults is the simplicity and relatively few assumptions made in modeling. A disadvantage with such a general fault model is that fault isolability may be lost compared to more detailed fault models. A small example is now included to illustrate fault modeling principles.

Example 2.2
A nonlinear state-space description including fault models can be written

$$\dot{x} = g(x, u, f)$$
$$y = h(x, u, f)$$

where x, u and y are the state, control signals, and measurements respectively. The signal f represents the fault, which in the fault-free case is $f \equiv 0$ and non-zero in a faulty case. The signal f here represents an arbitrary fault that can for example be a fault in an actuator or a sensor fault.
To illustrate fault modeling more concretely, consider a small idealized first principles model of the arm of an industrial robot. Linearized dynamics around one axis can be described by equations looking something like the following equations:

$$J_m\ddot{\varphi}_m = -F_{v,m}\dot{\varphi}_m + k_T u + \tau_{\text{spring}} \tag{2.1a}$$
$$\tau_{\text{spring}} = k(\varphi_a - \varphi_m) + c(\dot{\varphi}_a - \dot{\varphi}_m) \tag{2.1b}$$
$$J_a\ddot{\varphi}_a = -\tau_{\text{spring}} \tag{2.1c}$$
$$y = \varphi_m \tag{2.1d}$$

where the model variables are:

Symbol	Description
J_m	moment of inertia: motor
J_a	moment of inertia: arm
φ_m	motor position
φ_a	arm position
$F_{v,m}$	viscous friction, motor
k	stiffness coefficient, gear box
c	damping coefficient, gear box
k_T	torque constant, is 1 when torque control-loop is working
u	torque reference value, fed to the torque controller

Now, fault models are illustrated by modeling the following faults

1. A faulty torque-controller
2. Faulty arm position sensor, resulting in increased signal to noise ratio in the sensor signal
3. The robot has a load attached to the tip of the robot arm which is dropped
4. Collision of the robot arm with the environment

Associate a fault-variable f_1 to f_4 with the faults above. Introducing fault models in (2.1) gives for example

$$\begin{aligned} J_m\ddot{\varphi}_m &= -F_{v,m}\dot{\varphi}_m + (k_T + f_1(t))u + \tau_{\text{spring}} \\ \tau_{\text{spring}} &= k(\varphi_a - \varphi_m) + c(\dot{\varphi}_a - \dot{\varphi}_m) \\ (J_a + f_3)\ddot{\varphi}_a &= -\tau_{\text{spring}} + f_4(t) \\ y &= \varphi_m + \epsilon(f_2) \\ \epsilon(f_2) &= \begin{cases} N(0, \sigma_1^2) & f_2 = 0, \text{fault-free case} \\ N(0, \sigma_1^2 + f_2^2) & f_2 \neq 0, \text{faulty sensor} \end{cases} \\ \dot{f}_2 &= 0 \\ \dot{f}_3 &= 0 \end{aligned}$$

Here it is seen that faults f_2 and f_3 are assumed constant. Faults f_1 and f_4 is however not assumed constant. Such an assumption for f_4 would of course lead to a highly unrealistic fault model since f_4 is the torque exercised on the robot arm by the environment which naturally would not be constant in a collision situation. The time variability assumption of f_1 and f_4 is emphasized in the model by adding explicit time dependence.

2.1.2 Residuals and residual generators

The second step in the introductory presentation of model based diagnosis is a presentation of residuals and residual generators. Residuals is often a fundamental component in a diagnosis system. A residual is an, often time-varying, signal that is used as a fault detector. Normally, the residual is designed to be zero (or small in a realistic case where the process is subjected to noise and the model is uncertain) in the fault-free case and deviate significantly from zero when a fault occurs. Note however that other cases exist. In case of a likelihood-function based residual generator where the residual indicates how "likely" it is that the observed data is generated by a fault-free process, the residual is large in the fault-free case and small in a faulty case. But for the remainder of this text it is assumed, without loss of generality, that a residual is 0 in the fault-free case.

A residual generator is a filter that filters known signals to produce the residual. A linear residual generator is thus a proper MISO (Multiple Input

Single Output) filter $Q(s)$, filtering known signals y and u (measurements and control signals) producing an output r

$$r = Q(s) \begin{pmatrix} y \\ u \end{pmatrix}$$

Introduction to linear residual generator design is given in Section 2.2.

A more general non-linear residual generator on state-space form is given by two non-linear functions g and h and the filter

$$\begin{aligned} \dot{z} &= g(z, y, u) \\ r &= h(z, y, u) \end{aligned}$$

A main difficulty when designing residual generators is to achieve the disturbance decoupling, i.e. to ensure that the residual r is not influenced by unknown inputs that is not considered faults. This is was illustrated by Figure 1.2.

The main topic of the chapters to follow is procedures to design and analyze residual generators, i.e. the transfer function $Q(s)$ in the linear case and the non-linear functions g and h in the non-linear case.

2.1.3 Fault isolation

Before it is described how residuals and residual generators fit into a diagnosis system in Subsection 2.1.4, basic fault isolation strategies is described. Since fault isolation is not the topic of this thesis, this section illustrates fault isolation mainly by example.

To achieve isolation, several principles exists. For methods originating from the area of automatic control, at least three different approaches can be distinguished: *fixed direction residuals*, *structured residuals*, and *structured hypothesis tests*.

The idea of *fixed direction residuals* (Beard, 1971) is to design a residual vector such that the residual responds in different directions depending on what fault that acts on the system. Fault isolation is then achieved by studying and classifying the direction of the residual. This approach has not been so much used in the literature, probably because the problems associated with designing a residual vector with desired properties.

The idea of *structured residuals* (Gertler, 1991) is to have a set of residuals, in which each individual residual is sensitive to a subset of faults. By studying which residuals that respond, fault isolation can be achieved. Structured residuals have been widely used in the literature, in both theoretical and practical studies. The basic idea is quite simple and many methods for constructing suitable residuals have been presented both for linear and non-linear systems.

As a generalization of structured residuals, *structured hypothesis tests* has been proposed where the isolation method is formally defined. This formal definition makes it possible to use any possible fault models (Nyberg, 1999a),

I	f_1	f_2	f_3
r_1	1	1	0
r_2	X	0	1
r_3	1	1	1

II	f_1	f_2	f_3
r_1	1	1	0
r_2	1	0	1
r_3	1	1	1

III	f_1	f_2	f_3
r_1	0	X	X
r_2	X	0	X
r_3	X	X	0

IV	f_1	f_2	f_3
r_1	1	0	0
r_2	0	1	0
r_3	0	0	1

Figure 2.1: Examples of influence structures.

and perform deeper and further analysis of isolation properties of fault diagnosis systems. However, isolation issues are addressed very briefly here and the isolation procedure is mainly illustrated by example.

Residuals, as described in Section 2.1.2, can not only be used for fault detection, they can also be used for fault isolation in a structured residual/hypothesis-test isolation framework. The following example illustrates briefly how the isolation procedure works.

Example 2.3

To achieve isolation, in addition to fault detection, a *set* of residuals need to be designed where different residuals are sensitive to different subsets of faults. Which residuals that are sensitive to what faults, can be described by the *influence structure*[1]. Four examples of influence structures are shown in Figure 2.1. A number 1 in the i:th row and the j:th column represents the fact that residual r_i is sensitive to fault j. A number 0 in the i:th row and the j:th column represents the fact that residual r_i is not sensitive to fault j. An X in the i:th row and the j:th column represents the fact that residual r_i is *sometimes* sensitive to fault j. For example in structure I, it can be seen that residual r_2 is sometimes sensitive to fault f_1, not sensitive to fault f_2, and always sensitive to fault f_3. The isolation can ideally be performed by matching fault columns to the actual values of the residuals. Consider for example influence structure II in Figure 2.1, and assume that residuals r_1 and r_3 have signaled, but not r_2. The conclusion is then that fault f_2 has occurred.

In light of this illustration, it is convenient to introduce some notation. Consider residual r_2 in influence structure I which is completely insensitive to fault f_2 and sensitive to faults $\{f_1, f_3\}$, i.e. two sets of faults are considered in each residual generator design. The faults that the residual should be sensitive to are called *monitored* faults and the faults the residual should be insensitive to are called

[1] Also the method structured residuals uses influence structures but under different names. Names that have been used are for example *incidence matrix*(Gertler and Singer, 1990),*residual structure*(Gertler, 1998), and *coding set*(Gertler, 1991).

non-monitored faults. The non-monitored faults are said to be decoupled in the residual. Thus, the residual generator design problem is, in a structured residuals/hypothesis-test framework, essentially a decoupling problem.

It is worth noting that in general, the more faults that are decoupled in each residual, the greater is the possibility to isolate multiple faults. It is for example easy to see that influence structure IV facilitates isolation of 3 multiple faults while influence structure III only can handle single faults. The price to pay for this increased isolation performance is that more sensors are needed and the residual generators become more complex and model-dependent. These issues, among others, are explored in detail in the chapters to follow.

2.1.4 Model based diagnosis using residuals

This section describes how residuals is used in a structured residuals based diagnosis system. To be able to perform the fault isolation task, the residuals must react to faults according to an isolating influence structure. Thus, a design procedure would follow a procedure looking something like

1. Select a desired isolating influence structure. See (Gertler, 1998) for details on how, for example desired isolability properties restricts possible influence structures.

2. For each residual, collect unknown inputs and non-monitored faults, i.e. faults corresponding to zeros in the current row of the influence structure, in a vector d. The rest of the faults, the monitored faults, are collected in a vector f.

3. Design a residual that decouples d and verify what faults the residual is sensitive to. Ideally it is sensitive to all monitored faults, but it is possible that when decoupling d in the residual, also some of the monitored faults are decoupled.

4. If, when all residuals are designed, the resulting influence structure does not comply with design specifications, return to step 1 and re-design the desired influence structure.

It is clear from the procedure above that assuming a fault to be non-monitored is equal to introducing a zero at the corresponding location in the influence structure. Thus, by moving faults between monitored and non-monitored faults, the influence structure becomes a design choice made by the designer. Note that, for example the number of sensors and structural properties of the model both restricts the available design freedom when forming the influence structure. Thus, the influence structure is not entirely free for the designer to choose.

2.2 Consistency relations and residual generator implementation

A consistency relation is any relation between known or measured variables that, in the fault free case always holds. This section is intended to provide a background on consistency relations, how they can be used to form a residual generator and also indicate fundamental differences between how linear and non-linear consistency relations can be used for implementation. Consistency relations is not the only term used in fault diagnosis literature. Previously the words parity relations and parity equations was most common, but lately other words have appeared e.g. analytical redundancy relations (ARR) (Staroswiecki and Comtet-Varga, 2001). Here, the word consistency relation is used.

A consistency relation is an analytical relation between known signals z (and higher order derivatives) that is satisfied in the fault-free case. In case that the known signals are measurements and control signals then the known signals is the vector $z = [y\ u]$. Thus, g is a consistency relation if the following holds for all z that satisfy the original system equations (the model) when $f \equiv 0$:

$$g(z, \dot{z}, \ddot{z}, \dots) = 0 \tag{2.2}$$

In case of a time-discrete system, the time derivatives are substituted for time-delays. For time-continuous linear systems, a consistency relation can always (in the frequency domain) be written as

$$F(s)z = 0$$

where $F(s)$ is a polynomial vector (or matrix if multidimensional consistency relations are considered) in s. Such linear consistency relations are studied in detail in Chapter 3. Note that this holds only if all initial states are zero. Details on consistency relations for the case of non-zero initial states are discussed in Section 3.7.3.

Now, clearly these consistency relations are interesting for fault diagnosis since they describe a relation that is satisfied in the fault-free case and (possibly) not satisfied in case of a fault. If all variables included in the consistency relation (2.2) are known, a residual could be generated by letting

$$r = g(z, \dot{z}, \ddot{z}, \dots)$$

For dynamic systems, the relation g in general contains time-differentiated measurements and control signals, i.e. $\dot{u}$ and $\dot{y}$. Since these are not normally known, it is usually not possible to use the consistency relation directly in an implementation of a residual generator. In the linear case, this implementation problem is easily circumvented which is illustrated by the next example and described in detail in Chapter 3.

Example 2.4
Consider the linear model

$$y = \frac{1}{s^2 + as + b}u + f$$

The time domain interpretation of the transfer function is (with zero initial conditions):

$$\ddot{y} + a\dot{y} + by - u - \ddot{f} - a\dot{f} - bf = 0 \tag{2.3}$$

Equation (2.3) directly gives us a consistency relation, by examining the fault free case, i.e. by setting $f \equiv 0$ ($f = \dot{f} = \ddot{f} = 0$):

$$\ddot{y} + a\dot{y} + by - u = 0$$

and an equivalent frequency domain description of the relation:

$$(s^2 + as + b)y - u = 0$$

It is clear that if $\ddot{y}$ and $\dot{y}$ were known, we could calculate $r = \ddot{y} + a\dot{y} + by - u$ which would be 0 in the fault free case and deviate from 0 when $f \not\equiv 0$. However, the higher order derivatives are usually not known and one way to circumvent this complication is to add, e.g. low-pass dynamics to the consistency relation. That is, instead of computing the residual like $r = \ddot{y} + a\dot{y} + by - u$, compute the residual according to the differential equation

$$\ddot{r} + c_1\dot{r} + c_2 r = \ddot{y} + a\dot{y} + by - u \tag{2.4}$$

where constants c_1 and c_2 has been chosen to ensure a stable residual generator. In the frequency domain the residual generator transforms to

$$r = \frac{s^2 + as + b}{s^2 + c_1 s + c_2}y - \frac{1}{s^2 + c_1 s + c_2}u$$

which can be realized on explicit state-space form, i.e. higher order derivatives of y and u need not be used. The filter still has the property that $r = 0$ in the fault free case.

Consistency relations are frequently used for linear systems, but are equally applicable in the nonlinear case. However, in the example above it was straightforward to add dynamics to form a residual generator based on the linear consistency relation. The main property used was the linearity property. The following small example illustrates non-linear consistency relations and the problem that arise when using non-linear consistency relations to form a residual generator.

Example 2.5
A non-linear consistency relation is best represented in the time domain since no straightforward frequency domain description is possible. Consider the non-linear system described by state-space equations (inspired by flow equations in water-tank systems):

$$\dot{x} = -\sqrt{x} + ku$$
$$y = \sqrt{x}$$

A consistency relation for the model above can be derived by using the measurement equation which gives that

$$y - \sqrt{x} = 0$$

Differentiating both sides, another equation is obtained

$$\dot{y} - \frac{1}{2\sqrt{x}}(-\sqrt{x} + ku) = 0$$

Using these two equations, the state-variable x can easily be eliminated and

$$2y\dot{y} + y - ku = 0 \tag{2.5}$$

is obtained which is a consistency relation for the example model.
Unfortunately, it is not as easy as in the linear case to use the derived consistency relation to form a realizable residual generator. Adding linear dynamics like in (2.4) is in general not sufficient to be able to state the residual generator on state-space form. Further discussions on this topic is found Chapter 4.

The example also illustrates close links with elimination theory when deriving consistency relations. To obtain the consistency relation, unknown variables such as the state x and possibly other unknown inputs have to be eliminated from a set of equations derived from the original model equations. A well known method for linear residual generator design is the Chow-Willsky scheme, first described in (Chow and Willsky, 1984) and later extended to include unknown inputs in (Frank, 1990). This method is very similar to the non-linear example above where the model equations are differentiated a number of times until a set of equations is obtained where unknown variables can be eliminated[2]. A non-linear extension of this approach is investigated in Chapter 4 with an approach that is also closely related to obtaining input-output descriptions of a system described on state-space form (Jirstrand, 1998).

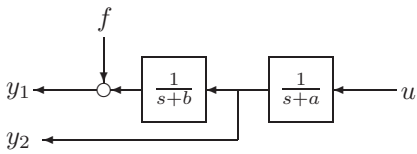
2.3 Problem motivation and discussion

Having now introduced a few basic principles of model based diagnosis, some background to the problems studied further in the chapters to follow are now illustrated and motivated.

[2]See Section 3.5.1 for more details on this approach.

2.3.1 The linear problem

For linear models on state-space form (or proper transfer functions), any consistency relation based design can be performed in an observer framework and vice versa. See for example (Ding et al., 1998, 1999a) for a recent description of these connections and (Patton, 1994) for a more historic view. However, this does not mean that the design algorithms are equivalent or have equal properties. To illustrate consistency based residual generator design, consider a small example system with two sensors, one actuator, and one fault, given by the block-diagram:



On analytical matrix form, the model description consists of the following linear equations:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{(s+a)(s+b)} \\ \frac{1}{(s+a)} \end{pmatrix} u + \begin{pmatrix} 1 \\ 0 \end{pmatrix} f \tag{2.6}$$

This system consists of two linear equations, i.e. we can expect to find two linearly independent consistency relations. Two linearly independent consistency relations are directly given, in the frequency domain, by the two model equations as:

$$(s+a)(s+b)y_1 - u = 0 \tag{2.7a}$$
$$(s+a)y_2 - u = 0 \tag{2.7b}$$

which are both satisfied in the fault-free case. Any consistency relation for the system can now be written as a linear combination of these two. However, the block diagram gives that a first order relationship exists between y_1 and y_2 since they are only separated with first order dynamics. This gives that also the following two consistency relations spans all consistency relations for the system,

$$(s+b)y_1 - y_2 = 0 \tag{2.8a}$$
$$(s+a)y_2 - u = 0 \tag{2.8b}$$

These two equations clearly captures the "most local" relationships in the model and reflects the structure of the process. Thus, a consistency relation based design algorithm should parameterize all solutions in these two relations. The design variables, free for the designer to choose, are then which linear-combination, with rational coefficients, of the two relations that should form the residual-generator. Since the consistency relations are closely related to the process model, this gives a natural interpretation of the design variables.

It is desirable to find a unified design/analysis procedure that is applicable to all linear model descriptions and all design problems. Of course, such an algorithm need good numerical performance to be able to cope with large or ill-conditioned model descriptions. In Chapter 3, a design algorithm for the decoupling problem is developed based on these considerations for systems described by proper transfer functions (or linear state-space descriptions) and in Section 3.7 it is shown how the algorithm also covers descriptor systems. The algorithm finds minimal order relations that span all possible consistency relations like (2.8). It is worth noting that a design method not considering the order easily results in a residual generator of the same order as the process model, and the difference can be significant. For example, with the 26:th order jet-engine model studied in Section 3.6.2, it was possible to design a residual generator based on a 4:th order relationship with fault sensitivity according to design specification.

Robustness

Low order relationships can also imply robustness against model uncertainty. Consider the following two residual generators for detecting f in (2.6) where the first is based on relation (2.7a) and the second on (2.8a):

$$r_1 = y_1 - \frac{1}{(s+a)(s+b)}u \qquad r_2 = y_1 - \frac{1}{s+b}y_2$$

Examining the expressions gives that both r_1 and r_2 has the same fault-response but r_1 relies on the accuracy of both model parameters a and b while r_2 only on parameter b. Thus, the lower order residual generator r_2 is less dependent on the model accuracy compared to r_1. This is not a general result; model dependency does not always decrease with the order. However, if the model has such a property, systematic utilization of low-order residual generators is desirable.

Uncertainty models

The last step in a residual generator design is to select the free design variables. To guide the selection, or at least restrict the design freedom, additional modeling/requirements on the residual generator is needed. A common way to introduce such extra requirements is to consider uncertain models. Two natural ways to model this uncertainty are parametric uncertainties in the model or subjecting the model to stochastic noise and investigate what available design freedom that is available with these extended models.

For stochastic linear systems, i.e. noise affected linear systems, there is not much work published. A common approach for such systems is to use Kalman-filters as residual generators which then produces residuals that are zero-mean and white with known covariance. However, for fault diagnosis, faults must be

decoupled in the residuals to facilitate fault isolation which means that the diagnosis decision should not be based on any residuals that are dependent on these unknown signals, i.e. they should be decoupled in the residual. Unknown input decoupling is not handled directly using basic, straightforward Kalman filtering theory. The nominal design algorithm from Chapter 3 handles decoupling and Chapter 5 extends the nominal design algorithm to also address disturbance decoupling in stochastic linear systems.

When the model is subjected to parametric uncertainties, it is a common approach to first state an optimization criterion reflecting diagnosis performance. Synthesis of residual generators is then performed by minimizing[3] the influence from worst-case uncertainties. For the optimization to produce a useful result, the criterion must be stated such that influence from both control signals and disturbances are attenuated while fault sensitivity is kept. A main difficulty is how to state the desired fault sensitivity without violating structural properties of the model. An algorithm, based on the nominal design algorithm, to form the optimization criterion is developed in Chapter 6.

2.3.2 The non-linear problem

When approaching the full non-linear problem, it quite naturally gets more difficult to derive complete solutions similar to what is available in the linear case. In Chapter 4, systematic methods, with strong computational support, to derive consistency relations for non-linear systems is pursued. Deriving consistency relations is closely related to variable elimination, which in a general non-linear system of equations is difficult. Therefore, only a class of non-linear systems is considered, namely models consisting of a set of polynomial differential-algebraic equations. For this class of systems, a design algorithm that finds polynomial consistency relations is derived. The algorithm then produces non-linear versions of relations like (2.7). The available design freedom is then similar to the design freedom in the linear case, i.e. which combination of the relations that should form the residual-generator. Here, a fundamental difference between the linear and non-linear case appears in the difficulty of using a non-linear consistency relation for residual generation which was discussed in Section 2.2.

The computational support provided by symbolic computer algebra packages such as Mathematica and Maple enables highly automated design procedures. However, the high computational complexity of these algorithms forces us to, if anything but small sized systems are considered, take additional measures to handle the complexity. In Chapter 4, structural analysis of the model equations is used to handle such complexity problems.

[3] assuming a small value of the criterion indicates good performance

3

Residual Generation Based on Linear Models

In this chapter, design and analysis tools for residual generators based on deterministic linear dynamic models are developed. For this linear design problem, a plethora of design methods for designing linear residual generators have been proposed in literature, see for example (Chen and Patton, 1999; Wünnenberg, 1990; Massoumnia et al., 1989; Nikoukhah, 1994; Chow and Willsky, 1984; Nyberg and Nielsen, 2000). However there still exists important topics that have not been resolved. Based on the discussion in Chapter 2, focus of the approach described here is a number of natural questions. For example

- Does the method find all possible residual generators?
- Does the method find residual generators of minimal order?
- What types of model descriptions can the method cope with? Due to the simple nature of linear systems, a design method for linear residual generators should be able to cope with any linear description, i.e. transfer functions, state-space descriptions or descriptor models.
- What are the numerical properties of the design algorithm?

Based on these fundamental questions, a design methodology is developed. Although the results are quite straightforward, the details proofs requires theory for polynomial matrices, rational vector spaces, and polynomial bases for these spaces (Kailath, 1980; Forney, 1975; Chen, 1984). Basic definitions and theorems used are, for the sake of convenience, collected in Appendix 3.A and 3.B.

3.1 The minimal polynomial basis approach

This section introduces the *minimal polynomial basis approach* to the design of linear residual generators. All derivations are performed in the time-continuous case but the same results for the time-discrete case can be obtained by substituting s by z and *improper* by *non-causal*. In the stochastic case, additional differences exist between the discrete and continuous-time case. This topic is further investigated in Chapter 5.

3.1.1 A general problem formulation

First, a general problem formulation is presented which has been used in many papers, e.g. (Gertler, 1991).

The class systems studied are assumed to be modeled by

$$y = G_u(s)u + G_d(s)d + G_f(s)f \tag{3.1}$$

where $y \in \mathbb{R}^m$ is the measurements, $u \in \mathbb{R}^{k_u}$ the known inputs to the system, $d \in \mathbb{R}^{k_d}$ the disturbances including non-monitored faults, and $f \in \mathbb{R}^{k_f}$ the monitored faults. The transfer functions $G_u(s)$, $G_d(s)$, and $G_f(s)$ are all assumed to be proper and of suitable dimensions.

Since we are considering linear systems also linear residual generators are considered, i.e. the residual is produced by linear filtering of measurements and control signals. For a system (3.1), linear residual generators can be defined as follows:

Definition 3.1 (residual generator for deterministic systems). *A stable and proper linear filter $Q(s)$ is a residual generator for (3.1) if and only if when $f \equiv 0$ it holds that*

$$r = Q(s)\begin{pmatrix} y \\ u \end{pmatrix} \equiv 0 \tag{3.2}$$

for all u, d.

From now on, all initial conditions is assumed 0. Since only strictly stable residual generators are considered, influence from these unknown initial states will vanish exponentially. If the transfer functions is non-proper, this is not generally true. This will be discussed further in Section 3.7 for so called descriptor systems. Note that for the residual to be useful for fault detection it must also hold that the transfer function from faults to the residual is non-zero. Sometimes this requirement is also included in the definition of the residual generator. Also, the requirement that the residual should be zero in the fault-free case is too strict in the general case. This since perfect decoupling is not always possible even in the deterministic case and we have to resort to *approximate* decoupling of disturbances. Such issues is further explored in Chapter 6. From now on, without loss of generality, r is assumed to be a *scalar* signal.

3.1.2 Derivation of design methodology

Inserting (3.1) into (3.2) gives

$$r = Q(s)\begin{bmatrix} G_u(s) & G_d(s) \\ I_{k_u} & 0_{k_u \times k_d} \end{bmatrix}\begin{bmatrix} u \\ d \end{bmatrix} + Q(s)\begin{bmatrix} G_f(s) \\ 0_{k_u \times k_f} \end{bmatrix} f \tag{3.3}$$

To make $r = 0$ when $f = 0$, it is required that disturbances and the control signal are *decoupled*, i.e. for $Q(s)$ to be a residual generator, it must hold that

$$Q(s)\begin{bmatrix} G_u(s) & G_d(s) \\ I_{k_u} & 0_{k_u \times k_d} \end{bmatrix} = 0$$

This implies that $Q(s)$ must belong to the left null-space of

$$M(s) = \begin{bmatrix} G_u(s) & G_d(s) \\ I_{k_u} & 0_{k_u \times k_d} \end{bmatrix} \tag{3.4}$$

This null-space is denoted $\mathcal{N}_L(M(s))$. The matrix $Q(s)$ need to fulfill two requirements to form a good residual generator: belong to the left null-space of $M(s)$ *and* provide good fault sensitivity properties in the residual. This filter $Q(s)$ can be, and has been, designed by observer methodology or by numerous other methods. Here however, the design method directly considers the null-space of $M(s)$ which will show to lead to appealing analysis possibilities and a straightforward and numerically good design algorithm. If, in the first step of the design, *all* $Q(s)$ that fulfills the first requirement is found and parameterized, then in a second step a single $Q(s)$ with good fault sensitivity properties can be selected. Thus, in a first step of the design, f or $G_f(s)$ do not need to be considered. The problem is then to find and parameterize *all* rational $Q(s) \in \mathcal{N}_L(M(s))$. Of special interest are residual generators of low order for reasons discussed in Chapter 2.

Finding all $Q(s) \in \mathcal{N}_L(M(s))$ can be done by finding a *minimal polynomial basis* for the rational vector-space $\mathcal{N}_L(M(s))$. Algorithms for computing such a basis for $\mathcal{N}_L(M(s))$ will be described in Section 3.2. For now, assume that such a basis has been found and is formed by the rows of a matrix denoted $N_M(s)$. By inspection of (3.4), it can be realized that the dimension of $\mathcal{N}_L(M(s))$ (i.e. the number of rows of $N_M(s)$) is

$$\dim\ \mathcal{N}_L(M(s)) = m - \text{rank}\ G_d(s) \overset{*}{=} m - k_d \tag{3.5}$$

The last equality, marked $\overset{*}{=}$, holds only if rank $G_d(s) = k_d$, but this should be the normal, or *generic*, case. More formal thoughts on genericity can be found in (Wonham, 1979).

Forming the residual generator

The second and final design-step is to use the polynomial basis $N_M(s)$ to form the residual generator. A decoupling polynomial vector is a polynomial row-vector $F(s)$ for which it holds that $F(s) \in \mathcal{N}_L(M(s))$. It is immediate that such

a row-vector corresponds to a consistency relation

$$F(s)\begin{pmatrix} y \\ u \end{pmatrix} = 0$$

The minimal polynomial basis $N_M(s)$ is irreducible according to Theorem 3.B.1, and then according to Theorem 3.B.2, all decoupling *polynomial* vectors $F(s)$ can be parameterized as

$$F(s) = \phi(s)N_M(s) \tag{3.6}$$

where $\phi(s)$ is a polynomial vector of suitable dimension. Thus, all consistency relations can be parameterized by a polynomial row-vector $\phi(s)$. Since $N_M(s)$ is a basis, the parameterization vector $\phi(s)$ have minimal number of elements, i.e. $N_M(s)$ gives a minimal parameterization of *all* decoupling polynomial vectors, not only minimal order.

It is also straightforward to show that since $N_M(s)$ is a minimal polynomial basis, one of the rows corresponds to a decoupling polynomial vector of minimal row-degree, see proof of Lemma 3.2 on page 31. Consistency relations was discussed thoroughly in Section 2.2 where also Example 2.4 indicated how, in the linear case, such a consistency relation could be used to design a residual generator. A general formulation of that example gives that a realizable rational transfer function $Q(s)$, i.e. the residual generator, can be found as

$$Q(s) = c^{-1}(s)F(s) \tag{3.7}$$

where the scalar polynomial $c(s)$ has greater or equal degree compared to the row-degree of $F(s)$. The degree constraint is the only constraint on $c(s)$ apart from a stability constraint. This means that the dynamics, i.e. poles, of the residual generator $Q(s)$ can be chosen freely as long as the roots of $c(s)$ lies in the open left half-plane. Therefore, $\phi(s)$ and $c(s)$ includes all design freedom that can be used to shape the fault-to-residual response. This also means that the minimal order of a realization of a residual generator is determined by the row-degree of the polynomial vector $F(s)$.

This design freedom can be used in many ways, e.g. can the poles of the residual generator be selected to impose a low-pass characteristic of the residual generator to filter out noise or high frequency uncertainties. However, if the residual generator problem is stated as in Definition 3.1 and the model is given by (3.1), any choice of $\varphi(s)$ and $c(s)$ are theoretically equally good. In practice this is of course not true, but to be able to form a systematic design procedure where the parameterization matrices are determined, additional modeling/requirements need to be introduced. Two such natural extensions are explored in subsequent chapters; the first is introduction of stochastic noise in the model and the second is introduction of parametric uncertainties in the model equations.

3.2 Methods to find a minimal polynomial basis for $\mathcal{N}_{\mathbf{L}}(\mathbf{M(s)})$

The problem of finding a minimal polynomial basis to the left null-space of the rational matrix $M(s)$ can be solved by a transformation to the problem of finding a polynomial basis for the null-space of a *polynomial* matrix. This latter problem is then a standard problem in linear systems theory where standard algorithms can be applied (Kailath, 1980).

The transformation from a rational problem to a polynomial problem can be done in different ways. In this section, two methods are demonstrated, where one is used if the model is given on transfer function form and the other if the model is given in state-space form. If there are no disturbances d, the problem of finding a basis to the left null-space of $M(s)$ is simplified and a method applicable in this case will also be described. Altogether, the results in this section will give us a computationally simple, efficient, and numerically stable method to find a polynomial basis for the left null-space of $M(s)$.

3.2.1 Frequency domain solution

When the system model is given on transfer function form (3.1), the transformation from the rational problem to a polynomial problem can be done by performing a right MFD (Kailath, 1980) of $M(s)$, i.e.

$$M(s) = \widetilde{M}_1(s)\widetilde{D}^{-1}(s) \tag{3.8}$$

By finding a polynomial basis for the left null-space of the *polynomial* matrix $\widetilde{M}_1(s)$, a basis is found also for the left null-space of $M(s)$ since $\widetilde{D}(s)$ is full rank. Thus the problem of finding a minimal polynomial basis for $\mathcal{N}_L(M(s))$ has been transformed into finding a minimal polynomial basis for $\mathcal{N}_L(\widetilde{M}_1(s))$.

3.2.2 State-space solution

When the system model is available in state-space form, it is here shown how the *system matrix* in state-space form can be used to find the left null-space of $M(s)$. The system matrix has been used before in the context of fault diagnosis, see for example (Nikoukhah, 1994; Magni and Mouyon, 1994).

Assume that the fault-free system is described in state-space form by,

$$\dot{x} = Ax + B_u u + B_d d \tag{3.9a}$$
$$y = Cx + D_u u + D_d d \tag{3.9b}$$

To be able to obtain a basis that is irreducible, it is required that the state x is controllable from $[u^T \; d^T]^T$.

Denote the system matrix $M_s(s)$, describing the system with disturbances as inputs:

$$M_s(s) = \begin{bmatrix} C & D_d \\ -sI_n + A & B_d \end{bmatrix} \tag{3.10}$$

Define a matrix P as

$$P = \begin{bmatrix} I_m & -D_u \\ 0_{n\times m} & -B_u \end{bmatrix} \tag{3.11}$$

The rationale of these definitions is that the Laplace transform of the model equations (3.9) can then be written as

$$M_s(s) \begin{pmatrix} x \\ d \end{pmatrix} = P \begin{pmatrix} y \\ u \end{pmatrix}$$

Then the following theorem gives a direct method on how to find a minimal polynomial basis to $\mathcal{N}_L(M(s))$ via the system matrix.

Theorem 3.1. *If the pair $\{A, [B_u\ B_d]\}$ is controllable and the rows of a polynomial matrix $V(s)$ form a minimal polynomial basis for $\mathcal{N}_L(M_s(s))$, then the rows of $W(s) = V(s)P$ form a minimal polynomial basis for $\mathcal{N}_L(M(s))$.*

Before this theorem can be proven, a lemma is needed:

Lemma 3.1.

$$\dim \mathcal{N}_L(M(s)) = \dim \mathcal{N}_L(M_s(s))$$

Proof. In this proof, controllability of (3.9) is assumed. See (Nyberg, 1999b) for the general proof.

The dimension of $\mathcal{N}_L(M(s))$ can immediately be seen as

$$\dim \mathcal{N}_L(M(s)) = m + k_u - \text{rank } M(s) = m - \text{rank } G_d(s) \tag{3.12}$$

since rank $M(s) = k_u +$ rank $G_d(s)$. By utilizing generalized Bezout-identities like in (Kailath, 1980, Sec. 6.4.2), it is seen that

$$M_s(s) \overset{\text{S}}{\sim} \begin{bmatrix} I_n & 0 \\ 0 & N(s) \end{bmatrix}$$

where $N(s)$ is the numerator in a right MFD $G_d(s) = N(s)D^{-1}(s)$ and $\overset{\text{S}}{\sim}$ represents Smith-form similarity. This gives that

$$\text{rank } M_s(s) = n + \text{rank } N(s) = n + \text{rank } G_d(s)$$

Then, $\dim \mathcal{N}_L(M_s(s))$ can be written as

$$\dim \mathcal{N}_L(M_s(s)) = n + m - \text{rank } M_s(s) = m - \text{rank } G_d(s)$$

which is equal to (3.12) which gives the theorem. ■

Now, return to the proof of Theorem 3.1:

Proof. In the fault free case, i.e. $f = 0$, consider the following relation between the matrices $M(s)$ and $M_s(s)$:

$$P \begin{pmatrix} y \\ u \end{pmatrix} = PM(s) \begin{pmatrix} u \\ d \end{pmatrix} = M_s(s) \begin{pmatrix} x \\ d \end{pmatrix}$$

If $V(s)M_s(s) = 0$, then the above expression is zero for all x and d, which implies that $W(s)M(s) = V(s)PM(s) = 0$, i.e. $W(s) \in \mathcal{N}_L(M(s))$. It is also immediate that if $V(s)$ is polynomial, $W(s) = V(s)P$ is also polynomial. Also, Lemma 3.1 gives that that $\dim \mathcal{N}_L(M_s(s)) = \dim \mathcal{N}_L(M(s))$. Then since both $V(s)$ and $W(s)$ has the same number of rows, the rows of $W(s)$ must span the whole null-space $\mathcal{N}_L(M(s))$, i.e. $W(s)$ must be a basis for $\mathcal{N}_L(M(s))$.

To now show that $W(s)$ is a *minimal* polynomial basis, it is according to Theorem 3.B.1 sufficient to prove that $W(s)$ is irreducible and row-reduced. It is clear that the following relation must hold:

$$V(s)[P \; M_s(s)] = V(s) \begin{bmatrix} I & -D_u & C & D_d \\ 0 & -B_u & -(sI - A) & B_d \end{bmatrix} = [W(s) \; 0] \tag{3.13}$$

Since the state x is controllable from u and d, the PBH test (Theorem 3.B.3) implies that the lower part of the matrix $[P \; M_s(s)]$ has full rank for all s, i.e. it is irreducible. Now assume that $W(s)$ is not irreducible. This means that there exists a s_0 and a $\gamma \neq 0$ such that $\gamma V(s_0)[P \; M_s(s_0)] = \gamma[W(s_0) \; 0] = 0$. Since $[P \; M_s(s_0)]$ has full row-rank it must hold that $\gamma V(s_0) = 0$. However, this contradicts the fact that $V(s)$ is a minimal polynomial basis. This contradiction implies that $W(s)$ must be irreducible.

Now, partition $V(s) = [V_1(s) \; V_2(s)]$ according to the partition of $M_s(s)$. Since $V(s) \in \mathcal{N}_L(M_s(s))$, it holds that

$$V_1(s)C = V_2(s)(sI - A) = sV_2(s) - V_2(s)A$$

Also, since each row-degree of $sV_2(s)$ is strictly greater than the corresponding row-degree of $V_2(s)A$, it holds that for each row i

$$\text{row-deg}_i \; sV_2(s) = \text{row-deg}_i \; V_2(s) + 1 = \text{row-deg}_i \; V_1(s)C$$

The above equation can be rearranged into the inequalities

$$\text{row-deg}_i \; V_2(s) < \text{row-deg}_i \; V_1(s)C \leq \text{row-deg}_i \; V_1(s)$$

This implies that $V_{hr} = [V_{1,hr} \; 0]$ where V_{hr} and $V_{1,hr}$ are the highest-row-degree coefficient matrices of $V(s)$ and $V_1(s)$ respectively. Since $V(s)$ is a minimal polynomial basis V_{hr} has full row rank from which it follows that $V_{1,hr}$ has full row rank.

From the definition of P it follows that

$$[W_1(s) \; W_2(s)] = [V_1(s) \quad (-V_1(s)D_u - V_2(s)B_u)] \tag{3.14}$$

From (3.14) it follows that the highest-row-degree coefficient matrix of $W(s)$ looks like $W_{hr} = [V_{1,hr} \star]$ where $\star$ is any constant matrix. Since $V_{1,hr}$ has full row-rank so has W_{hr}, i.e. $W(s)$ is row reduced. ■

What happens if the controllability assumption in Theorem 3.1 is dropped is directly given by the following corollary.

Corollary 3.1. *Let the polynomial matrix $V(s)$ form a minimal polynomial basis for $\mathcal{N}_L(M_s(s))$, then the rows of $W(s) = V(s)P$ form a polynomial basis, not necessarily irreducible, for $\mathcal{N}_L(M(s))$.*

Proof. Following the proof of Theorem 3.1 it is seen that if the realization is not controllable from $\left[u^T\ d^T\right]^T$, then the matrix $[P\ M_s(s)]$ in (3.13) does not have full row-rank for all s. Thus, $W(s) = V(s)P$ is a basis but not necessarily irreducible. This has the implication that all decoupling polynomial vectors $F(s)$ can *not* be parameterized as in (3.6). ■

3.2.3 No disturbance case

If there are no disturbances, i.e. $G_d(s) = 0$, the matrix $M(s)$ becomes $M_{nd}(s) = [G_u^T(s)\ I]^T$. A minimal basis is then given directly by the following theorem:

Theorem 3.2. *If $G(s)$ is a proper transfer matrix, $\bar{D}_G(s)$, $\bar{N}_G(s)$ form an irreducible left MFD of $G(s)$, and $\bar{D}_G(s)$ is row-reduced, then*

$$N_{M_{nd}}(s) = [\bar{D}_G(s)\ \ -\bar{N}_G(s)] \tag{3.15}$$

forms a minimal polynomial basis for the left null-space of the matrix $M_{nd}(s)$.

Proof. It is immediate to evaluate

$$[\bar{D}_G(s)\ \ -\bar{N}_G(s)] \begin{bmatrix} G_u(s) \\ I \end{bmatrix} = 0$$

Also, the dimension of the left null-space of $M_{nd}(s)$ has dimension m, i.e. the number of measurements, which equals the number of rows in $N_M(s)$. Since $\bar{D}_G(s)$ and $\bar{N}_G(s)$ is co-prime, $N_M(s)$ will be irreducible. Let

$$\bar{D}_G(s) = S_D(s)D_{hr} + L_D(s)$$
$$\bar{N}_G(s) = S_N(s)N_{hr} + L_N(s)$$

where D_{hr} and N_{hr} be the highest row-degree coefficient matrices for $\bar{D}_G(s)$ and $\bar{N}_G(s)$ where D_{hr} will be of full rank since $\bar{D}_G(s)$ is row-reduced.

Since the transfer function $G(s)$ is proper, i.e. the row degrees of $\bar{N}_G(s)$ is less or equal to the row degrees of $\bar{D}_G(s)$. A high-degree coefficient decomposition of the basis $N_M(s)$ will look like

$$[\bar{D}_G(s)\ \ -\bar{N}_M(s)] = S_D(s)[D_{hr}\ \star] + \tilde{L}(s)$$

where $\star$ is any matrix. Since D_{hr} is full rank, so is $[D_{hr}\ \star]$, which gives that the basis is row-reduced which ends the proof. ■

Note that not just any irreducible MFD will suffice, the row-reducedness property is also needed and an algorithm that provides such an MFD is found in (Strijbos, 1996) and is implemented in (Polynomial Toolbox 2.5 for Matlab 5, 2001). The row-degrees of the minimal polynomial basis for $\mathcal{N}_L(M_{nd}(s))$ are closely related to the observability indices according to the following theorem:

Theorem 3.3. *The set of observability indices of a proper transfer function $G(s)$ is equal to the set of row degrees of $\bar{D}_G(s)$ and also (3.15) in any row-reduced irreducible left MFD $G(s) = \bar{D}_G^{-1}(s)\bar{N}_G(s)$.*

Proof. A proof of the dual problem, controllability indices, can be found in (Chen, 1984, p. 284). ■

Thus, a minimal polynomial basis for the left null-space of matrix $M_{nd}(s)$ is given by a left MFD of $G(s)$ and the order of the basis is the sum of the observability indices of $G(s)$.
Remark 1: Note that, in the general case, the observability indices of the pair $\{A, C\}$ do not give the row-degrees of a minimal polynomial basis for $\mathcal{N}_L(M(s))$. However, as will be shown in Theorem 3.6, the minimal observability index of $\{A, C\}$ does give a lower bound on the minimal row-degree of the basis.
Remark 2: The result (3.15) implies that finding the left null-space of the rational transfer matrix (3.4), in the general case with disturbances included, can be reduced to finding the left null-space of the rational matrix

$$\widetilde{M}_2(s) = \bar{D}_G(s)H(s) \tag{3.16}$$

In other words, this is an alternative to the use of the matrix $\widetilde{M}_1(s)$ in (3.8). This view closely connects with the so called frequency domain methods, which are further examined in Section 3.5.

3.2.4 Finding a minimal polynomial basis for the null-space of a general polynomial matrix

For the general case, with disturbances included, the only remaining problem is how to find a minimal polynomial basis to for the left null-space of a general polynomial matrix. This is a well-known problem in the general literature on linear systems. When numerical performance is considered, a specific algorithm based on the *polynomial echelon form* (Kailath, 1980) has been proven to be both fast and numerically stable. Such an algorithm is implemented in the command `null` in Polynomial Toolbox 2.5 for Matlab 5 (2001).

3.3 Matlab sessions

To illustrate the simplicity of the design algorithm, a complete Matlab-session (requires control and polynomial toolbox) for design of residual generators is

included for both cases when the model is given on transfer function form and state-space form. In both cases, $\phi(s)$ in (3.6) is set to $\phi(s) = [1\ 0]$ and $c(s)$ in (3.7) to $c(s) = (s+1)^3$. First the state-space case:

```
Ms = [C Dd;-(s*eye(nx)-A) Bd];
P = [eye(m) -Du;zeros(nx,m) -Bu];
Nms = null(Ms.').';
fi = [1 0]; c = (s+1)^3;
[Qa,Qb,Qc,Qd] = lmf2ss(fi*Nms*P,c);
Q = ss(Qa,Qb,Qc,Qd);
```

A Matlab-session for the case when the model is given on transfer function form is similar:

```
M = [Gu Gd;eye(ku) zeros(ku,kd)
[M1,D] = ss2rmf(M.a,M.b,M.c,M.d);
Nm = null(M1.').';
fi=[1 0]; c = (s+1)^3;
[Qa,Qb,Qc,Qd] = lmf2ss(fi*Nm,c);
Q = ss(Qa,Qb,Qc,Qd);
```

As shown above, no diagnosis specific code need to be developed and the design procedure solely relies on high performance numerical routines in established Matlab toolboxes. The two main operations are `null` which computes a basis for the null-space of a polynomial matrix and `lmf2ss` which derives a state-space realization of a left MFD description. The simplicity of the design algorithm is possible because of the abstraction made to consider polynomial matrices. The numerical performance in diagnosis applications of the above code is illustrated further in Section 3.6 and Section 3.7.

3.4 Bounds on maximum and minimum row-degree of the basis

In Section 2.3, influence of the order of the residual generator on e.g. robustness properties was discussed. This section will provide a deeper analysis on the residual generator order.

In Section 3.1 it was shown how the row-degrees of a minimal polynomial basis for the matrix $M(s)$ were connected with the order of the residual generator. Now follows an analysis of these row-degrees where upper limits on the maximum and minimum row-degree of a matrix is derived. The notation n is used to denote the number of states in a given state-space representation and n_x will be used to denote the number of states controllable from $\left[u^T\ d^T\right]^T$.

3.4.1 Upper bound for the maximum row-degree of the basis

Theorem 3.4 (Nyberg,1999b). *A matrix whose rows form a minimal polynomial basis for $\mathcal{N}_L(M(s))$ has all row-degrees $\leq n_x$.*

Before Theorem 3.4 can be proven, a few lemmas are needed.

Lemma 3.2. *Let the rows of $F(s)$ form a minimal polynomial basis for a rational vector space $\mathcal{F}$. Denote the row-degrees of $F(s)$ with $\mu_1 \leq \cdots \leq \mu_\alpha$. Then it holds that $\mu_i \leq m_i, i = 1, \ldots, \alpha$ where m_i is the row-degrees of any polynomial basis for $\mathcal{F}$.*

Proof. Let $P(s)$ be a polynomial basis for $\mathcal{F}$ with row-degrees m_i. Let the rows in $P(s)$ be ordered such that $m_1 \leq \cdots \leq m_\alpha$.

The theorem is proved by contradiction. Assume that $\mu_1 \leq m_1, \ldots, \mu_{i-1} \leq m_{i-1}$ but $\mu_i > m_i$. Since $F(s)$ is an irreducible basis, it holds that

$$p_j(s) = \sum_{l=1}^{\alpha} f_l(s) q_l(s) \quad j = 1, \ldots, i \tag{3.17}$$

where $q_l(s)$ is polynomials.

If $i = 1$, then $\deg p_1(s) < \mu_j \quad j = 1, \ldots, \alpha$, i.e. according to Theorem 3.B.5, $p_1(s)$ can not be a linear combination of the rows in the row reduced matrix $F(s)$. However, this contradicts (3.17).

If $i > 1$, according to the assumption, the following relations hold:

$$\deg p_j(s) \leq m_i < \mu_i \quad j = 1, \ldots, i$$

According to the predictable degree property it must hold that in (3.17), $q_l(s) \equiv 0, l = i, \ldots, \alpha$. Thus, the upper summation limit can at maximum be $i - 1$, i.e. equation (3.17) can be rewritten as:

$$p_j(s) = \sum_{l=1}^{i-1} f_l(s) q_l(s) \quad j = 1, \ldots, i$$

This contradicts the linear independence of the $p_1(s), \ldots, p_i(s)$ polynomial row vectors since they are spanned by $f_1(s), \ldots, f_{i-1}(s)$ ending the proof. ■

Lemma 3.3. *Let $P(s)$ be a matrix with maximum row-degree 1. Then the maximum row-degree of a minimal polynomial basis for $\mathcal{N}_L(P(s))$ is less or equal to rank $P(s)$.*

Proof. Since $P(s)$ is a matrix pencil it can be transferred to Kronecker Canonical Form (Theorem 3.B.4) by pre- and post-multiplication with constant non-singular matrices U and V, i.e. $P_{KCF}(s) = UP(s)V$. Also, since all matrices besides $\widetilde{L}_\nu$, in the block-diagonal Canonical Form has full row-rank the

left null-space structure of $P_{KCF}(s)$ is fully characterized by the left Kronecker indices $\nu_1, \ldots, \nu_\beta$.

It is also easy to check that the left null-space of $\widetilde{L}_\nu$ is given by

$$v_\nu(s) = [1 \; s \; \cdots \; s^\nu]$$

i.e, the degree of the left null-space vectors is directly given by the left Kronecker indices. Thus, a basis for the left null-space of $P_{KCF}(s)$ is given by a matrix on the form

$$N_{P_{KCF}}(s) = \begin{bmatrix} 0 & \cdots & v_{\nu_1}(s) & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & v_{\nu_2}(s) & 0 & 0 & \cdots & 0 \\ & & & & \ddots & & & \\ 0 & \cdots & 0 & & & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & v_{\nu_\beta}(s) & \cdots & 0 \end{bmatrix}$$

i.e. the maximum row-degree of $N_{P_{KCF}}(s)$ is $\max_i \nu_i$. A basis for the left null-space of $P(s)$ is given by $N_{P_{KCF}}(s)U$ where U is constant and non-singular which gives that the maximum row-degree its maximum row-degree is $\max_i \nu_i$. It holds that rank $\widetilde{L}_\nu = \nu$ which gives that rank $P(s) \geq \sum_{i=1}^{\beta} \nu_i$, i.e. the maximum row-degree of a polynomial basis (and thereby also a minimal polynomial basis according to Lemma 3.2) is less than rank $P(s)$. ■

Lemma 3.4. *The row-degrees of a minimal polynomial basis for $\mathcal{N}_L(M(s))$ is equal to the row-degrees of a minimal polynomial basis for $\mathcal{N}_L(M_s(s))$, where $M_s(s)$ is a system matrix with the pair $\{A, [B_u \; B_d]\}$ controllable.*

Proof. Let $V(s)$ be a minimal polynomial basis for $\mathcal{N}_L(M_s(s))$ and partition $V(s) = [V_1(s) \; V_2(s)]$ according to the partition of $M_s(s)$. Then, since we know that $V(s) \in \mathcal{N}_L(M_s(s))$, it holds that

$$V_1(s)C = V_2(s)(sI - A) = sV_2(s) - V_2(s)A$$

Also, since each row degree of $sV_2(s)$ is strictly greater than the corresponding row-degree of $V_2(s)A$, it holds that for each row i

$$\text{row-deg}_i \; sV_2(s) = 1 + \text{row-deg}_i \; V_2(s) = \text{row-deg}_i \; V_1(s)C$$

The above equation can be rearranged to

$$\text{row-deg}_i \; V_2(s) < \text{row-deg}_i \; V_1(s)C \leq \text{row-deg}_i \; V_1(s) \tag{3.18}$$

i.e. $\text{row-deg}_i \; V(s) = \text{row-deg}_i \; V_1(s)$. From the definition of P in (3.11) it follows that

$$W(s) = [W_1(s) \; W_2(s)] = V(s)P = [V_1(s) \quad (-V_1(s)D_u - V_2(s)B_u)] \tag{3.19}$$

Equations (3.18) and (3.19) directly give

$$\text{row-deg}_i \; W(s) = \text{row-deg}_i \; V_1(s) = \text{row-deg}_i \; V(s),$$

i.e. the row degrees of $W(s)$ and $V(s)$ are equal. According to Theorem 3.1, $W(s)$ and $V(s)$ are minimal polynomial bases for $\mathcal{N}_L(M(s))$ and $\mathcal{N}_L(M_s(s))$ respectively and the lemma follows immediately. ■

Remark:

Lemma 3.4 implies that the row-degrees of $N_M(s)$ equals the left Kronecker indices of the matrix pencil $M_s(s)$. There exist a lot of literature and algorithms for computing the Kronecker indices of a general matrix pencil, e.g. (Misra et al., 1994; Wang et al., 1975; Aling and Schumacher, 1984; Kågström, 1986).

Now return to the proof of Theorem 3.4:

Proof. Let n_x be the order of a minimal state-space realization of (3.9), controllable from $\left[u^T\ d^T\right]^T$. Let $M_s(s)$ be the corresponding system matrix, i.e.

$$M_s(s) = \begin{bmatrix} C & D_d \\ -(sI - A) & B_d \end{bmatrix}$$

and let the rows of N_{DB} be a basis for the left null-space of $[D_d^T\ B_d^T]^T$. Then we have that

$$N_{DB}M_s(s) = \left[N_{DB}\begin{bmatrix} C \\ -(sI - A) \end{bmatrix}, 0\right] \tag{3.20}$$

The left part of the matrix (3.20) has rank $\leq n_x$. From Lemma 3.3 we know that a minimal polynomial basis for (3.20) has row degrees less or equal to n_x. Let the rows of a matrix $Q(s)$ form such a basis.

The matrix $Q(s)N_{DB}$ forms a polynomial basis for $\mathcal{N}_L(M_s(s))$ and since $Q(s)$ has row degrees less or equal to n_x, the row degrees of the basis $Q(s)N_{DB}$ is also less or equal to n_x. Thus, according to Lemma 3.2, a *minimal* polynomial basis for $\mathcal{N}_L(M_s(s))$ has lower or equal row-degrees than the polynomial basis $Q(s)N_{DB}$.

Since a minimal polynomial basis for $\mathcal{N}_L(M_s(s))$ has maximum row-degree $\leq n_x$, Lemma 3.4 implies that also a minimal polynomial basis for $\mathcal{N}_L(M(s))$ has maximum row-degree $\leq n_x$, ending the proof. ■

The result of Theorem 3.4 is important for several reasons, the residual generators obtained directly from the vectors of the minimal basis, are in one sense the only ones needed. All other are filtered versions (i.e. linear combinations) of these residual generators. With this argument, Theorem 3.4 shows that we do not need to consider residual generators of orders greater than n_x.

Remark

Related problems have been investigated in (Chow and Willsky, 1984) and (Gertler et al., 1990). In (Chow and Willsky, 1984), it was shown that, in the no-disturbance case, there exist a parity equation of order $\leq n$. In (Gertler et al., 1990), it was shown that for a restricted class of disturbances, there exist a parity function of order $\leq n$. However the result of Theorem 3.4 is stronger since it includes *arbitrary* disturbances and shows that there exist a *basis* in which the maximum row-degree is $\leq n_x$.

3.4.2 Bounds for the minimal row-degree of the basis

Theorem 3.5 (Frisk,2000a). *An upper bound for the minimal row-degree ρ_{min} of a basis for $\mathcal{N}_L(M(s))$ is given by*

$$\rho_{min} \leq \lfloor \frac{n_x + n_d}{m - n_d} \rfloor$$

where

$$n_d = Rank \begin{pmatrix} B_d \\ D_d \end{pmatrix}$$

is the number of linearly independent disturbances.

Before Theorem 3.5 can be proven, some more results are needed. If $n_d < k_d$, i.e. there exists linear dependencies between disturbances, rewrite the system description with a new set of n_d linearly independent disturbances. That is, find $\widetilde{B}_d$ and $\widetilde{D}_d$ with dimensions $n_x \times n_d$ and $m \times n_d$ respectively such that

$$\text{Im} \begin{pmatrix} B_d \\ D_d \end{pmatrix} = \text{Im} \begin{pmatrix} \widetilde{B}_d \\ \widetilde{D}_d \end{pmatrix}$$

Then, denote

$$\widetilde{M}_\rho = \underbrace{\begin{bmatrix} Q & R & & & \\ & Q & R & & \\ & & & \ddots & \\ & & & Q & R \end{bmatrix}}_{(\rho+2)(n_x+n_d)} \Bigg\} (\rho+1)(m+n_x)$$

where $M_s(s) = Q + sR$. Then, the following lemma can be stated:

Lemma 3.5. *The space $\mathcal{N}_L(M_s(s))$ contains a ρ-degree polynomial vector if and only if $\widetilde{M}_\rho$ does not have full row rank.*

Proof. Let $F(s)$ be a ρ-degree polynomial matrix in $\mathcal{N}_L(M_s(s))$. Then it holds that

$$\begin{aligned} 0 = F(s)M_s(s) &= (F_0 + F_1 s + \cdots + s^\rho F_\rho)M_s(s) = \\ &= [F_0 \; F_1 \cdots F_\rho] \begin{bmatrix} M_s(s) \\ sM_s(s) \\ \vdots \\ s^\rho M_s(s) \end{bmatrix} = \widetilde{F}\widetilde{M}_\rho \begin{bmatrix} I \\ sI \\ \vdots \\ s^\rho I \end{bmatrix} \end{aligned}$$

From the equation above it is clear that a ρ-degree polynomial $F(s)$ is in $\mathcal{N}_L(M_s(s))$ if and only if $\widetilde{F}\widetilde{M}_\rho = 0$. The lemma follows directly because such a $\widetilde{F}$ can only exist if $\widetilde{M}_\rho$ does not have full row-rank. ■

A similar result can also be found in (Karcanias and Kalogeropoulos, 1988). Now, return to the proof of Theorem 3.5.

Proof. Using Lemma 3.4 and 3.5 it is clear that a ρ-degree polynomial vector is in $\mathcal{N}_L(M(s))$ if and only if $\widetilde{M}_\rho$ does not have full row rank. A sufficient condition for $\widetilde{M}_\rho$ not to have full row-rank is that the number of rows is larger than the number of columns, i.e.

$$(\rho+1)(m+n_x) > (\rho+2)(n_x+n_d)$$

Straightforward manipulations of the inequality results in

$$\rho > \frac{n_x+n_d}{m-n_d} - 1$$

Note that the inequality $m > n_d$ always holds if a residual generator exists which can be seen directly in (3.5). Therefore, the smallest integer ρ that fulfills the inequality is $\lfloor \frac{n_x+n_d}{m-n_d} \rfloor$ which completes the proof. ■

Remark: A similar result without disturbance decoupling, i.e. when $n_d = 0$, can be found in (Mironovskii, 1980).

Theorem 3.2 and 3.3 states that in the no-disturbance case, the observability indices of the pair $\{A, C\}$ give exactly the row-degrees of a minimal polynomial basis for $\mathcal{N}_L(M(s))$. In the more general case, including disturbances, this is not true anymore. However, the minimal observability index of $\{A, C\}$ can still be used to obtain a lower bound according to the following theorem

Theorem 3.6. *A lower bound for the minimal row-degree ρ_{min} of a basis for $\mathcal{N}_L(M(s))$ is the minimal observability index μ_{min} of $\{A, C\}$, i.e. $\rho_{min} \geq \mu_{min}$ with equality when $n_d = 0$.*

Proof. Denote the system matrix without disturbances with $M_s^{(nd)}(s)$, i.e.

$$M_s^{(nd)}(s) = \begin{bmatrix} C \\ sI - A \end{bmatrix}$$

A direct consequence of Theorem 3.3 is that the row-degrees of a minimal polynomial basis for the left null-space of $M_s^{(nd)}(s)$ is equal to the observability indices of the pair (A, C). Let $c_{\min}$ be the minimum observability index of (A, C). Then, according to Lemma 3.5, $c_{\min}$ is the lowest ρ such that $\widetilde{M}_\rho^{(nd)}$ does not have full row-rank. Let

$$Q = [Q_1 \; Q_2] = \begin{bmatrix} C & D_d \\ A & B_d \end{bmatrix}$$
$$R = [R_1 \; R_2] = \begin{bmatrix} 0 & 0 \\ -I & 0 \end{bmatrix}$$

Then, by a trivial column reordering, $\widetilde{M}_\rho$ can be written on the form

$$\widetilde{M}_\rho = \begin{bmatrix} Q_1 & R_1 & & & & Q_2 & R_2 & & \\ & Q_1 & R_1 & & & & Q_2 & R_2 & \\ & & \ddots & & & & & \ddots & \\ & & & Q_1 & R_1 & & & Q_2 & R_2 \end{bmatrix} L$$
$$= [\widetilde{M}_\rho^{(nd)} \ \star] L$$

where L is a square, full rank pivoting matrix and $\star$ is a matrix of suitable dimensions who is not of further interest here. From the equation above, it is clear that if $\widetilde{M}_\rho^{(nd)}$ has full row-rank, then also $\widetilde{M}_\rho$ has full row-rank. Also, for all $\rho < c_{\min}$, $\widetilde{M}_\rho^{(nd)}$ and thereby also $\widetilde{M}_\rho$, will have full row-rank. The theorem then follows directly from Lemma 3.5, i.e. there exists no ρ-degree polynomial in $\mathcal{N}_L(M_s(s))$ where $\rho < c_{\min}$. ■

A similar result can also be found in (Ding et al., 1999b) in a parity-space setting and in (Karcanias and Kalogeropoulos, 1988) in a geometrical setting.

3.5 Relation to other residual generator design methods

This section discusses the relation between the minimal polynomial basis approach and two other design methods for linear residual generation, the parity-space approach and the so called frequency domain approach. Relations to these two approaches are included because they both exhibit interesting relations to the minimal polynomial basis approach.

3.5.1 The parity-space approach

The *parity space approach* to the design of linear residual generators was presented in (Chow and Willsky, 1984; Frank, 1990). It's basic version is based on a state-space description (3.9). By differentiating the measurement equation ρ times, and substituting in (3.9a) a new formulation of the system equations is obtained. For notational simplicity, let

$$\Psi_\rho(s) = \begin{bmatrix} I_m & sI_m & \dots & s^\rho I_m \end{bmatrix}^T$$

The fault-free system description is then, in the frequency domain,

$$\Psi_\rho(s) y(t) = \mathcal{O}_\rho x(t) + Q_\rho \Psi_\rho(s) u(t) + H_\rho \Psi_\rho d(t)$$

where $\mathcal{O}_\rho$ is the observation matrix up to order ρ and Q_ρ,H_ρ is lower triangular Toeplitz matrices describing fault/disturbance influence on the process. The exact appearance of these matrices is not of importance here, see any of

the referred works cited above for details. A consistency relation can then be obtained as

$$w[\Psi_\rho(s) \quad -Q_\rho\Psi_\rho(s)]\begin{pmatrix} y(t) \\ u(t) \end{pmatrix} = 0$$

where $w \in \mathcal{N}_L([\mathcal{O}_\rho \; H_\rho])$. A relation to the minimal polynomial basis approach is straightforward to verify; $w[\Psi_\rho(s) \quad -Q_\rho\Psi_\rho(s)] \in \mathcal{N}_L(M(s))$ which should be compared to (3.6). Also, Theorem 3.4 gives that with $\rho \geq n$, this is an alternative way to parameterize all consistency relations, i.e. all polynomial row-vectors in $\mathcal{N}_L(M(s))$. This basic approach does not however provide a basis since an over-parameterized solutions is obtained. This issue is clearly illustrated in a high-order design example in Section 3.6.2.

Also, minimal order consistency relations are not explicitly found with this basic version of the algorithm. But, by starting with $\rho = 0$ and iteratively increase ρ until a consistency relation is found, a minimal order consistency relation is found. However, a basis can not be obtained in such a way and several modifications to the original algorithm is necessary to obtain a modified scheme which will produce a minimal polynomial basis for the left null-space of $M(s)$. The modified scheme and the equivalence with the minimal polynomial basis approach is thoroughly described in (Nyberg, 1999b).

3.5.2 Frequency domain approaches

A number of design methods described in literature are called *frequency domain methods* where the residual generators are designed with the help of different transfer matrix factorization techniques. Examples are (Frank and Ding, 1994) for the general case with disturbances and (Ding and Frank, 1990; Viswanadham et al., 1987) in the non-disturbance case. The methods can be summarized as methods where the residual generator is parameterized as

$$r = R(s)[\tilde{D}(s) \quad -\tilde{N}(s)]\begin{pmatrix} y \\ u \end{pmatrix} \tag{3.21}$$

where $\tilde{D}(s)$ and $\tilde{N}(s)$ form a left co-prime factorization of $G(s)$ over $\mathcal{RH}_\infty$. Note the close relationship with Equation (3.15) where the factorization is performed over polynomial matrices instead of over $\mathcal{RH}_\infty$.

This solution however does not generally generate a residual generator with minimal order. In (Ding and Frank, 1990) and (Frank and Ding, 1994), the co-prime factorization is performed via a minimal state-space realization of the complete system, including the disturbances as in Equation (3.9). This results in $\tilde{D}(s)$ and $\tilde{N}(s)$ of a degree that, in the general case, is larger than the lowest possible order of a disturbance decoupling residual generator. Thus, to find a lowest order residual generator extra care is required since "excess" states need to be canceled out.

3.6 Design examples with nominal models

This section includes two design examples, each included to illustrate different properties of the algorithm described. The first example, a model of aircraft dynamics, illustrates design methodology, the available design freedom and advantages of the minimality property. It also shows in principle how a set of residual generators can be designed to fit in a diagnosis system based on structured hypothesis tests/structured residuals. The second example is included to illustrate some numerical properties of the algorithm and also show consequences of the minimality property of the algorithm.

3.6.1 Design example 1: Aircraft dynamics

This model, taken from (Maciejowski, 1989), represents a linearized model of vertical-plane dynamics of an aircraft. The inputs and outputs of the model are

Inputs		Outputs	
u_1:	spoiler angle [tenth of a degree]	y_1:	relative altitude [m]
u_2:	forward acceleration [ms^{-2}]	y_2:	forward speed [ms^{-1}]
u_3:	elevator angle [degrees]	y_3:	Pitch angle [degrees]

The model has state-space matrices:

$$A = \begin{bmatrix} 0 & 0 & 1.132 & 0 & -1 \\ 0 & -0.0538 & -0.1712 & 0 & 0.0705 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.0485 & 0 & -0.8556 & -1.013 \\ 0 & -0.2909 & 0 & 1.0532 & -0.6859 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ -0.12 & 1 & 0 \\ 0 & 0 & 0 \\ 4.419 & 0 & -1.665 \\ 1.575 & 0 & -0.0732 \end{bmatrix}$$

$$C = [I_3 \ 0] \qquad D = 0_{3\times 3}$$

Suppose the faults of interest are sensor-faults (denoted f_1, f_2, and f_3), and actuator-faults (denoted f_4, f_5, and f_6). Also, assume that the faults are modeled with additive fault models. The total model, including fault models then becomes:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = G(s) \left(\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} f_4 \\ f_5 \\ f_6 \end{bmatrix} \right) + \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

where

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

Thus, the transfer function from fault vector f to measurement vector y becomes, $G_{yf}(s) = [I_3 \ G(s)]$.

Decoupling of faults in the elevator angle actuator

The first design example is intended to illustrate the design procedure and also illustrate how available design freedom can be utilized, e.g. when selecting the influence structure from Section 2.1.3, or selecting the dynamics of the residual generator.

The design objective here is to design a residual generator $Q_1(s)$ that decouples faults in the elevator angle actuator, f_6. Then, matrix $G_d(s)$ from Equation 3.1 correspond to all signals that are to be decoupled, i.e. considered disturbances. In this case then, $G_d(s)$ becomes the column in $G_{yf}(s)$ corresponding to f_6. Matrix $G_f(s)$ corresponds to the monitored faults and therefore $G_f(s)$ becomes the other columns.

Without actually deriving the minimal polynomial basis $N_M(s)$ for $\mathcal{N}_L(M(s))$, some properties of the basis $N_M(s)$ can be obtained by using results from Section 3.4. According to formula (3.5), the dimension of the null-space $\mathcal{N}_L(M(s))$ is 2, i.e. there exists exactly two linearly independent polynomial row-vectors that decouples f_6. The maximum row-degree of $N_M(s)$ will, according to Theorem 3.4 be $\leq n = 5$. Theorem 3.5 and Theorem 3.6 give an upper and lower bound on the minimum row-degree ρ_{min} of $N_M(s)$. The minimal observability index is $\mu_{min} = 1$, and thus $1 \leq \rho_{min} \leq \lfloor \frac{5+1}{3-1} \rfloor = 3$. This implies that the minimum degree residual generator will have an order of 1, 2, or 3.

Since the model is given in state-space form, Theorem 3.1 is used to extract $N_M(s)$. Calculations in MATLAB give

$$N_M(s) = \left[\begin{array}{cccccc} 0.0705s & s+0.0538 & \cdots & & & \\ 22.7459s^2+14.5884s & -6.6653 & \cdots & & & \\ & \cdots & 0.091394 & 0.12 & -1 & 0 \\ & \cdots & s^2-0.93678s-16.5141 & 31.4058 & 0 & 0 \end{array} \right]$$

An additive actuator fault is decoupled if and only if the actuator is not used in the calculation of the residual. This is verified by the two 0 in the last column in the basis. This basis also gives that the dimension of the null-space $\mathcal{N}_L(M(s))$ is 2, i.e. there exists exactly two linearly independent numerators that decouples f_6 which was the expected result according to (3.5). The row-degrees of the basis is 1 and 2, i.e. it is a basis of order 3. From this it is clear that the filter of least degree decoupling f_6 is a first order filter corresponding to row 1 in the basis. By setting ϕ in (3.6) to $\phi = [1\ 0]$ and $c(s)$ in (3.7) to $c(s) = 1 + s$ the filter is made realizable and results in the following 1:st order filter

$$Q_1(s) = \frac{1}{1+s} \begin{bmatrix} 0.0705s & s+0.0538 & 0.091394 & 0.12 & -1 & 0 \end{bmatrix} \tag{3.22}$$

The polynomial, here scalar, $c(s)$ need to have degree ≥ 1 to make the filter realizable since the row-degree of the first row in $N_M(s)$ is 1. In this example, a first order polynomial is chosen to get minimal order. The polynomial $c(s)$ is chosen to $1 + s$ to detect faults with energy in frequency ranges up to 1 rad/s.

Let $G_{rd}(s) = Q(s)\begin{pmatrix} G_u(s) & G_d(s) \\ I & 0 \end{pmatrix}$, which should be zero if infinite precision arithmetics were used. Computing the size of $G_{rd}(s)$ using the infinity norm gives $\|G_{rd}(s)\|_\infty \approx -220$ dB which is close to machine precision. Figure 3.1 shows how the monitored faults influence the residual. Maximum gain from fault to residual is ≈ 0 dB, i.e. control signals and the decoupled fault has no significant influence compared to monitored faults. The leftmost plot in

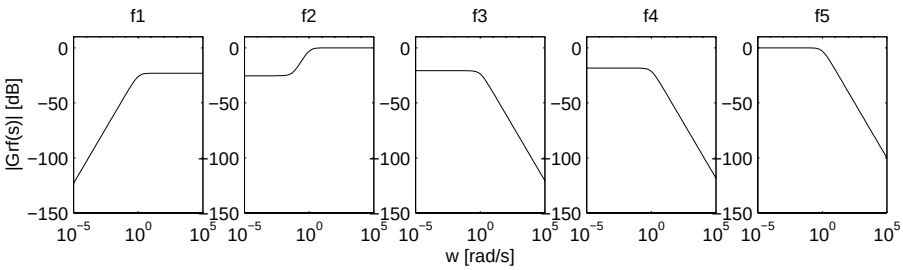


Figure 3.1: Magnitude bode plots for the monitored faults to the residual.

Figure 3.1 also shows that DC-gain from fault 1 to the residual is 0 and f_1 is therefore difficult to detect since the effect in the residual of a constant fault f_1 disappears. If a fault has zero DC-gain for any residual generator, the fault is said to be *weakly detectable*. Since the algorithm from Chapter 3 finds all residual generators, then to investigate if any fault is weakly detectable, evaluate

$$N_M(s)\begin{bmatrix} L(s) \\ 0 \end{bmatrix}\Bigg|_{s=0}$$

If any column is identically zero, then the corresponding fault is weakly detectable. A thorough investigation on fault detectability properties can be found in (Nyberg, 2000).

The design freedom can also be used to avoid non minimum-phase behavior of the fault response in the residual. Such behavior is undesirable since the response to a step fault would be something like Figure 3.2. This is clearly undesirable since the residual would first indicate a fault, then pass below the threshold again, cancelling the alarm, before settling above the threshold. Non minimum-phase behavior can easily be detected by computing zeros of the polynomial elements in the basis $N_M(s)$.

Decoupling several faults

As noted in the example above, the dimension of the null-space when decoupling f_6 was 2. This was expected according to (3.5) and therefore additional freedom exists. This freedom can for example be used to decouple more than one fault in each residual to shape the influence structure, e.g. for multiple fault isolation.

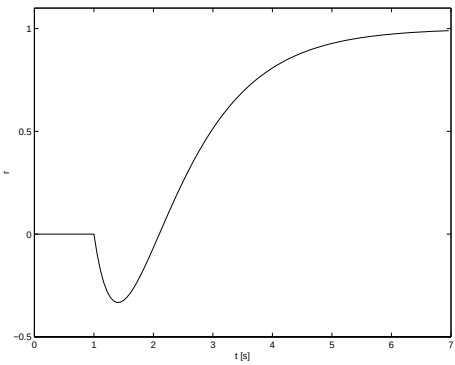


Figure 3.2: Non minimum-phase fault response in the residual

For example, designing a filter that decouples both f_1 and f_4, i.e. faults in the first sensor and the first actuator results in a null-space of dimension 1 and a minimal basis of order 3. The polynomial $c(s)$ must be at least a third order polynomial to make the filter realizable since the row-degree of the basis vector is 3. Selecting $c(s)$ to be $c(s) = (1+s)^3$, the residual generator becomes

$$Q_2(s) = \frac{1}{(1+s)^3} \begin{bmatrix} 0 \\ 36.825s^2 + 13.8953s + 1.1157 \\ s^3 + 0.61619s^2 + 4.4322s + 2.048 \\ 0 \\ -36.825s - 11.9626 \\ 1.665s - 0.28273 \end{bmatrix}^T$$

where the decoupling of faults in sensor 1 is evident since the transfer function from y_1 (the first column in $Q_2(s)$) is 0.

Designing a complete diagnosis system

As was noted above, there exists design-freedom to "insert" one or two zeros in the influence structure which means that there exists some freedom when choosing the influence structure. One example is to decouple one fault in each residual as illustrated in the influence structure

	f_1	f_2	f_3	f_4	f_5	f_6
r_1	0	X	X	X	X	X
r_2	X	0	X	X	X	X
r_3	X	X	0	X	X	X
r_4	X	X	X	0	X	X
r_5	X	X	X	X	0	X
r_6	X	X	X	X	X	0

With this structure is it possible to detect and isolate all 6 different faults, assuming only single faults. Calculations in MATLAB shows that to achieve this, the minimum order filter that generates all residuals is an 11:th order filter (2:nd order for r_1 to r_5 and a 1:st order filter for r_6).

Another example of an influence structure that also isolates the 6 different faults is

	f_1	f_2	f_3	f_4	f_5	f_6
r_1	0	0	X	X	X	X
r_2	X	0	0	X	X	X
r_3	X	X	0	0	X	X
r_4	X	X	X	0	0	X

where the possibility to introduce more than one 0 in each row is utilized. Because of this additional freedom, only 4 residuals are required here to form an isolating influence structure. Continuing in the same fashion, decoupling two faults in each residual it is possible to design residuals with an influence structure that is able to isolate two simultaneous faults. The price for this increased isolation performance is that the minimal filter generating residuals according to the second influence structure above is a 15:th order filter where calculations in MATLAB gives the (minimal) orders of residuals r_1 to r_4 to be 4, 5, 4, and 2 respectively.

To illustrate how the final diagnosis system could work, residual generators r_1 to r_6 have been designed according to the first influence structure (the one with 0 in the diagonal). Simulations are shown in Figure 3.3 and 3.4. These simulations only show principle operation of the system, e.g. no noise attenuation aspects has been considered. In the simulations, step faults of amplitude 1 are introduced at $t = 1$. It is evident that, nominally, the f_3 response in r_3 and the f_4 response in r_4 is 0. The isolation procedure is then simply to match observed residual pattern to columns in the influence structure.

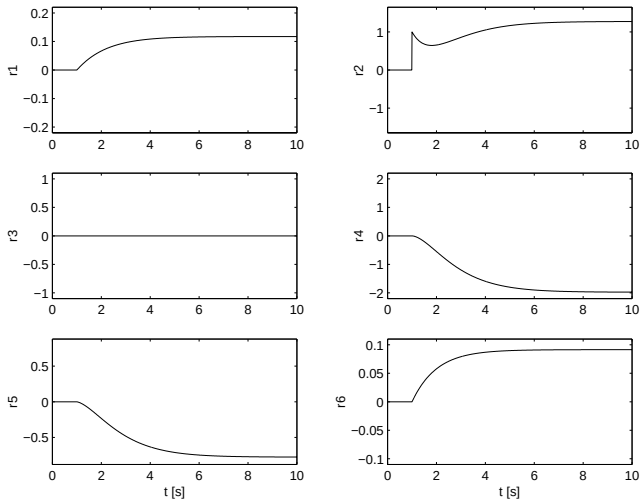


Figure 3.3: Fault simulation with step fault in f_3 at $t = 1$. Especially, note that fault influence on residual 3 is zero in correspondence with column 3 in the influence structure.

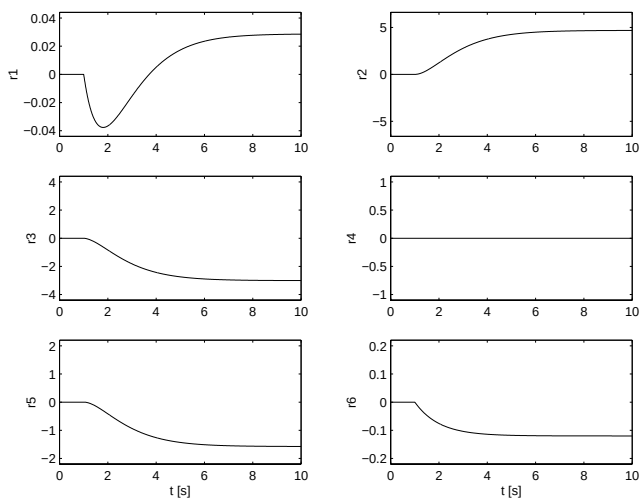


Figure 3.4: Fault simulation with step fault in f_4 at $t = 1$. Note that fault influence on residual 4 is zero in correspondence with column 4 in the influence structure.

3.6.2 Design example 2: Turbo-jet engine

The turbo-jet engine used in this example is developed by Volvo Aero Corporation (VAC) and used in fighter aircrafts of the Swedish air force. Figure 3.5 shows a schematic picture of the engine. VAC uses a high-order non-linear model

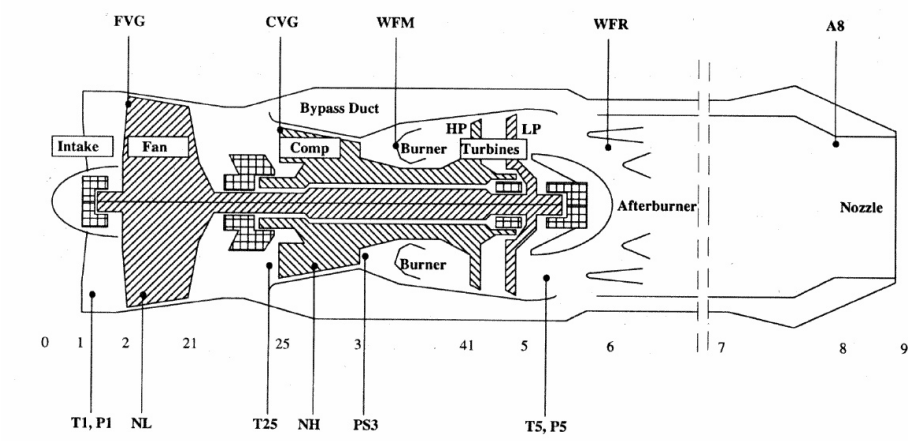


Figure 3.5: The turbo-jet engine

of the engine for analysis and control design. This model can also be used for diagnosis purposes. In this example, the model has been linearized in an operating point and the resulting model, after non-controllable and non-observable modes are eliminated, is a 26:th order model. The model used includes 8 sensors, indicated by the acronyms at the bottom half of the figure, and 5 actuators indicated by the acronyms on the upper half of the figure. In the operating point where the model is linearized, only 4 actuators are used. The meanings of the acronyms are given in the following table.

	Inputs		Outputs
FVG:	Variable fan geometry	T1:	Inlet temperature
CVG:	Variable compressor geometry	T25:	Compressor temperature
WFM:	Main fuel injector	T5:	Low-pressure turbine temperature
A8:	Variable Nozzle Area	P1:	Inlet pressure
WFR:	Afterburner fuel injector	PS3:	Compressor pressure
		P5:	Low-pressure turbine pressure
		NL:	Low-pressure turbine speed
		NH:	High-pressure turbine speed

The model is numerically stiff since it models fast dynamics, such as thermodynamics in small control volumes, and slow dynamics such as heating phenomena of metal. The largest time-constant in the model is about 10^5 times larger than the smallest time constant. This, in connection with the high-order model makes the model numerically sensitive which demands good numerical performance of the design algorithm.

Suppose actuator and sensor faults are studied, modeled in the same manner as in the aircraft dynamics example with additive fault models. Due to the large amount of sensors in the process, there exists a large amount of freedom shaping the influence structure. If no faults are decoupled, the dimension of $N_M(s)$ is 8, according to Theorem 3.2. This means that up to 7 independent faults can be decoupled in the residual, i.e. it is possible to introduce up to 7 zeros in a row of the influence structure. This issue will however not be explored further here.

A design with the minimal polynomial basis approach is performed where all 4 actuator faults are decoupled, i.e. only relations between measurements are desired. Due to the number of states, and sensors, solving for $N_M(s)$ in this problem is computationally (and numerically) quite demanding. The nullspace $\mathcal{N}_L(M(s))$ is of dimension 4 and the basis has row-degrees 4, 5, 5, and 6. Suppose faults with frequency up to 1 rad/s is of interest, then choosing the first row in $N_M(s)$ as a numerator and selecting $c(s) = (s+1)^4$ results in a 4:th

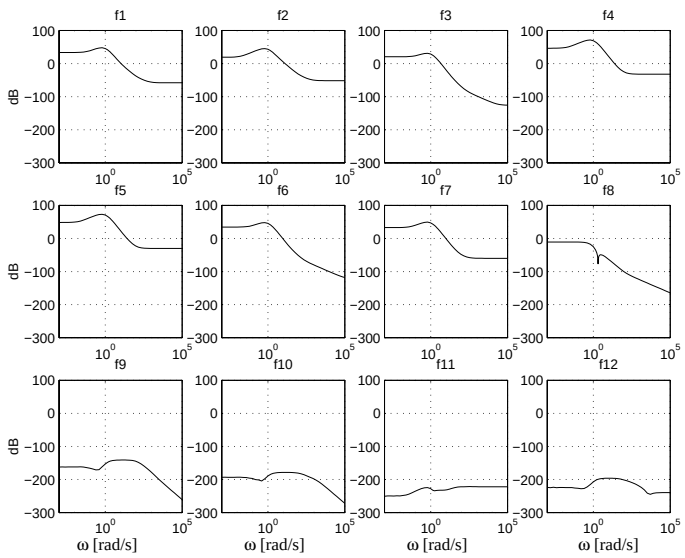


Figure 3.6: Fault response in residual. Residual generator designed with methods from Chapter 3. Note how the gains from the decoupled faults $f_9, \ldots, f_{12}$ is significantly lower than the gains from the other faults.

order filter.

Figure 3.6 shows the absolute values of the transfer functions from the different faults to the residual. Faults 1 to 8 is the sensor faults and 9 to 12 the actuator faults. It is clear that f_9 to f_{12} (the actuator faults) are attenuated according to design specifications.

To illustrate the numerical difficulties in this example, a design is also performed with a well-known method, the Chow-Willsky scheme. Performing the same design with the basic Chow-Willsky design method on a balanced realization of the model gives Figure 3.7 which should be compared to Figure 3.6. It

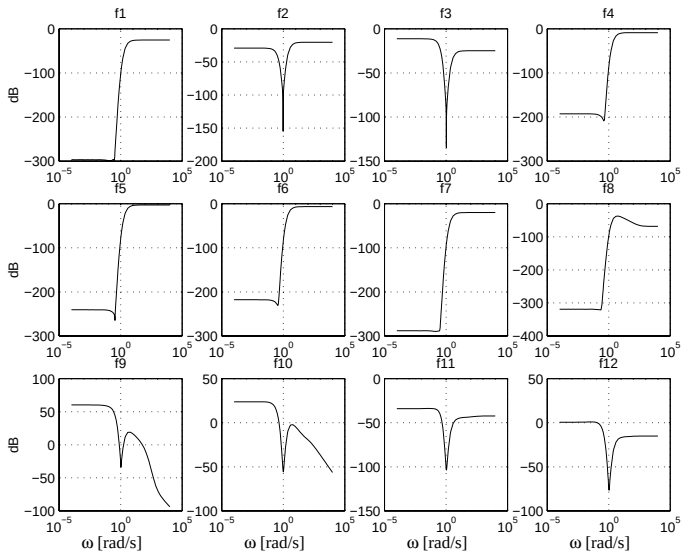


Figure 3.7: Fault response in residual. Residual generator designed with the Chow-Willsky scheme. The decoupling has failed since the gains from $f_9, \ldots, f_{12}$ is not lower than the gains from $f_1, \ldots, f_8$.

is clear that the decoupling of f_9 to f_{12} here has failed and this is because of severe numerical problems during design.

In the minimal polynomial basis approach there were some freedom in selecting the numerator, since there were 4 to choose from. This kind of freedom also exists in the Chow-Willsky approach, however the solution is in the basic form over-parameterized so there are dependent numerators. In this rather large example, where in the case of the minimal polynomial basis approach there were 4 numerators, the standard Chow-Willsky solution gives 211 numerators. Because of the large over-parameterization not all numerators were evaluated but when generating Figure 3.7 a representative numerator was chosen. Note that the Chow-Willsky solution gives all numerators, thus also those of minimal order. The minimal solution can be obtained by selecting a clever linear

combination of all 211 numerators. However, this operation is non-trivial and is of the same complexity order as solving the original design problem. Readers interested in details regarding the relationship between the minimal polynomial basis approach and Chow-Willsky schemes are referred to (Nyberg, 1999b) where this is investigated in great detail.

3.7 Descriptor systems

This section shows how the design and analysis algorithms developed in this chapter easily can be extended to cover also *descriptor* systems, or generalized state-space systems. Pros and cons of descriptor formulations is discussed in (Müller, 2000). Linear descriptor systems are described on state-space form as

$$E\dot{x} = Ax + B_u u + B_d d \tag{3.23a}$$

$$y = Cx + D_u u + D_d d \tag{3.23b}$$

where the only difference from a standard state-space description is the matrix E where E can be square, non-square, singular, or non-singular. A model with a square, non-singular E can of course be transferred to a state-space description. However, due to numerical reasons this may not be a suitable operation. In (Sincovec et al., 1981) it is argued that the descriptor formulation in general has numerical advantages compared to an ordinary state-space descriptions and in addition, it is more closely related to physical variables and the physical system structure.

A descriptor system is solvable, i.e. a unique solution exists given input signals and initial conditions, if and only if the matrix pencil $sE - A$ is regular (or equivalently matrix $sE - A$ is square and full rank) (Yip and Sincovec, 1981). A solvability assumption of (3.23), sometimes referred to as a regularity assumption, is often made when analyzing descriptor systems for fault diagnosis (Kratz et al., 1995; Shields, 1994). However, in general, for example for under-modeled or differential-algebraic systems, matrix E need not even be square and it is shown in e.g. (Müller and Hou, 1993; Hou, 2000) that a solvability assumption is not necessary in observer design or residual generator design. Therefore, no solvability assumption is made here either, i.e. the matrix pencil $sE - A$ can be square or non-square, singular or non-singular.

The design methodology for descriptor systems is exactly as described for non-descriptor systems in Section 3.1 and all residual generators can be parameterized as in (3.6) and (3.7). A slight modification of the design is needed though, since for non-solvable descriptor systems, it is not even possible to write the system on a transfer function form like in (3.1). This minor difference is dealt with in the next section where a set $\mathcal{F}$ of all polynomial consistency relations is defined instead of the left null-space of matrix $M(s)$. With this notational difference, the only difference from the non-descriptor case is a slight modification on how to compute a basis for $\mathcal{F}$/the left null-space of $M(s)$ which

will be discussed next. In Chapter 5, the design algorithm is extended to consider stochastic linear models where it will also be shown how the algorithm handles stochastic descriptor systems.

3.7.1 Computing $\mathcal{N}_L(M(s))$ for descriptor systems

If the descriptor system (3.23) is solvable, it can be written on a, possibly non-proper, transfer-function form. Thus, just as for non-descriptor systems, the system description can be on transfer function form (3.1) or on a generalized state-space form (3.23). In case of a transfer function model, the frequency domain solution from Section 3.2 is directly applicable. The only difference from the non-descriptor case is that $M(s)$ might be non-proper, which does not influence the computation of $N_M(s)$.

The state-space solution from Section 3.2 of course need a slight modification since a new state-space matrix E has been introduced. Here however, a fundamental difference from the non-descriptor case appears. The singularity of matrix E has implications on minimality properties and this will be investigated further in this section. The system matrix $M_s(s)$ from (3.10) is in the descriptor case replaced by

$$M_s(s) = \begin{bmatrix} C & D_d \\ -(sE - A) & B_d \end{bmatrix} \tag{3.24}$$

When the system was written on transfer function form we saw that for all polynomial row-vectors $F(s) \in \mathcal{N}_L(M(s))$, a consistency relation could be formed as

$$F(s) \begin{pmatrix} y \\ u \end{pmatrix} = 0$$

However, as indicated previously, in the general case it may not even be possible to write the system on transfer function form, a replacement for the space $\mathcal{N}_L(M(s))$ is needed that characterizes all consistency relations. From now on, let the set $\mathcal{F}$, over rational functions, be defined by

$$\mathcal{F} = \{F(s) \in \mathbb{R}^{1\times(m+k_u)}(s). < y, u > \text{satisfies (3.23)} \rightarrow F(s) \begin{pmatrix} y \\ u \end{pmatrix} = 0\}$$

A version of Theorem 3.1, the theorem used for design based on a state-space description of the process, for descriptor systems is then given by

Theorem 3.7. *Assume that the model equations (3.23) is linearly independent and let the rows of a polynomial matrix $V(s)$ form a polynomial basis for $\mathcal{N}_L(M_s(s))$. Then the rows of $W(s) = V(s)P$ form a polynomial basis for $\mathcal{F}$ where P is given by (3.11).*

Proof. The proof idea is to first prove that $W(s)$ spans $\mathcal{F}$, and then show that the rows are linearly independent.

First, note that for the descriptor system (3.23) it holds that

$$P\begin{pmatrix} y \\ u \end{pmatrix} = M_s(s)\begin{pmatrix} x \\ d \end{pmatrix} \tag{3.25}$$

and that matrix $F(s) \in \mathcal{F}$ if and only if

$$\exists x, d;\ M_s(s)\begin{pmatrix} x \\ d \end{pmatrix} = P\begin{pmatrix} y \\ u \end{pmatrix} \Rightarrow F(s)\begin{pmatrix} y \\ u \end{pmatrix} = 0$$

The spanning property of $W(s)$ is proven by observing that:

$$\begin{aligned} \text{Measurements satisfy model equations} \Leftrightarrow \exists x, d;\ M_s(s)\begin{pmatrix} x \\ d \end{pmatrix} = P\begin{pmatrix} y \\ u \end{pmatrix} \Leftrightarrow \\ \Leftrightarrow P\begin{pmatrix} y \\ u \end{pmatrix} \in \text{ Im } M_s(s) \Leftrightarrow N_{M_s}(s)P\begin{pmatrix} y \\ u \end{pmatrix} = 0 \end{aligned} \tag{3.26}$$

Thus, any $F(s) \in \mathcal{F}$ can be written as $F(s) = \varphi(s)W(s)$, i.e. $W(s)$ spans $\mathcal{F}$. The last equivalence in (3.26) is due to the fact that a vector x is in the column space of matrix A if and only if x is orthogonal to the complementary space of A, i.e. $x \in$ Im $A \Leftrightarrow N_A x = 0$ where N_A is a basis for the left null-space of A.

The linear independence of the rows in $W(s)$ is easily seen by observing the following matrix:

$$V(s)[P\ M_s(s)] = V(s)\begin{bmatrix} I & -D_u & C & D_d \\ 0 & -B_u & -(sE-A) & B_d \end{bmatrix} = [W(s)\ \ 0] \tag{3.27}$$

Since $V(s)$ is a polynomial basis for $\mathcal{N}_L(M_s(s))$, it has full row-rank. Also, the assumption on the linear independence of the descriptor model (3.23) gives that the matrix $[P\ M_s(s)]$ has full row rank and multiplication of two full row-rank matrices gives a full row-rank matrix. Thereby is linear independence of the rows of $W(s)$ proved, i.e. $W(s)$ has been proven full-rank and to span $\mathcal{F}$ thus ending the proof. ■

Remark: If the assumption on linear independence of the descriptor equations (3.23) is not fulfilled, it is evident from the proof of Theorem 3.7 that $W(s)$ may not be a basis even though it spans $\mathcal{F}$. So, linear independence among the model equations might lead to an over-parameterized solution.

Also, it is important to note that the $W(s)$ in the theorem is *not* a minimal polynomial basis, it is merely a basis for $\mathcal{F}$. The two properties needed for $W(s)$ to be a minimal polynomial basis is according to Theorem 3.B.1, row-reducedness and irreducibility. The matrix $W(s)$ can always be made row-reduced and irreducible by a full-rank transformation matrix $T(s)$

$$W_{\min}(s) = T^{-1}(s)W(s)$$

where $T(s)$ is e.g. the greatest left divisor of $W(s)$. However, if $T(s)$ has non-stable zeros, feasible residual generators can not (unless cancellation of the non-stable zeros occurs) be parameterized as

$$Q(s) = c^{-1}(s)\phi(s)W_{\min}(s) \tag{3.28}$$

This is because of unknown initial conditions, thus it is not always desirable to make $W(s)$ an irreducible basis. This is further discussed in Section 3.7.3 where influence from non-zero initial states are considered. If $W(s)$ is irreducible but not row-reduced, a minimal polynomial basis can always safely be obtained by a unimodular transformation $T(s)$ (which of course is stable). This transformation can be done in several ways, see (Kailath, 1980, p. 386) for principles and e.g. (Polynomial Toolbox 2.5 for Matlab 5, 2001) for numerically stable algorithms.

For state-space systems, the irreducibility property was ensured by assuming that the pair $\{A, [B_u \ B_d]\}$ was controllable. A similar condition for the descriptor case is given by the following corollary

Corollary 3.2. *Matrix $W(s)$ in Theorem 3.7 is irreducible if $\{E, A, [B_u \ B_d]\}$ is R- or C-controllable.*

Proof. If $\{E, A, [B_u \ B_d]\}$ is R-controllable, then the bottom block-row of matrix (3.27) is irreducible according to Theorem 3.B.8 which gives that $[P \ M_s(s)]$ is irreducible. The product of two full row-rank, irreducible matrices results in a irreducible matrix and since $V(s)$ is irreducible by definition, $W(s)$ is irreducible. Theorem 3.B.8 and Theorem 3.B.9 directly gives that C-controllability implies R-controllability and therefore is also C-controllabilty a sufficient condition on irreducibility of $W(s)$. ■

Another important topic is fault detectability. Nyberg (2000) provides a thorough treatment on criterions for fault detectability for non-descriptor systems based on the minimal polynomial basis approach. It is trivial to show that the criterions is identical for descriptor systems, e.g. a fault is detectable if and only if

$$\text{Im} \begin{bmatrix} D_f \\ B_f \end{bmatrix} \not\subseteq \text{Im} \begin{bmatrix} C & D_d \\ -(sE - A) & B_d \end{bmatrix}$$

which can be controlled with a simple rank test.

3.7.2 Design example

The design example is taken from (Hou, 2000), where a descriptor model of a three-link planar manipulator is used, see Figure 3.8. The process works by moving the end effector repeatedly from point A to point B, e.g. cleaning a facade. The manipulator is equipped with three actuators that can apply torques at all three joints. Four sensors is used measuring the height of the end effector, the contact force in the x direction, and tracking signals. The fault-free

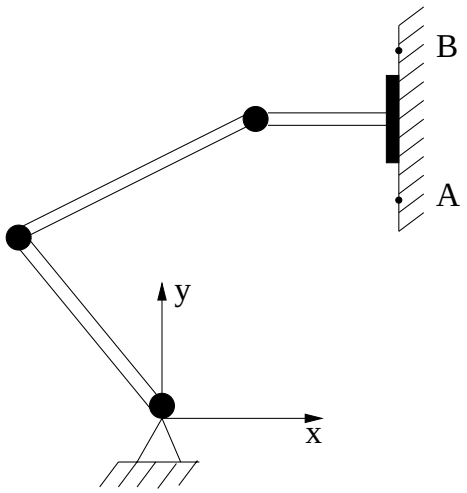


Figure 3.8: The three-link manipulator

model is stated on descriptor form in (Hou and Müller, 1996). The model has 11 states: Cartesian coordinates of the end effector (3 states), derivatives of the Cartesian coordinates (3 states), two Lagrangian multipliers (2 states), and controller states (3 states).

The model is given on the form:

$$\begin{aligned} E\dot{x} &= Ax + B_u u + B_{f_1} f_1 + B_{f_2} f_2 \\ y &= Cx \end{aligned}$$

The process is subjected to two faults, f_1 and f_2 corresponding to malfunctions in an actuator and in the tracking reference signal. Numerical values for the state-space matrices are taken from (Hou and Müller, 1996) and (Hou, 2000) and are also included in Appendix 3.C. The descriptor system is solvable, i.e. E is square and the pencil $sE - A$ has full rank.

The design goal is to design two residual generators $Q_1(s)$ and $Q_2(s)$, where fault f_2 is decoupled in $Q_1(s)$ and fault f_1 is decoupled in $Q_2(s)$. Performing the design using Theorem 3.7 followed by a transformation to a minimal polynomial basis gives

$$N_{M_s}(s) = \begin{bmatrix} -0.38s - 1.6 & 1.3 & -0.99s - 49 & \cdots \\ 0.12s + 0.52 & -0.42 & 0.027s + 1.1 & \cdots \\ -42s^2 - 2.4 \cdot 10^2 s - 5.3 \cdot 10^2 & 3.7 \cdot 10^2 & 0.27s - 64 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} \cdots & 0.098s + 4.5 & 0 & 0.99 & -0.098 \\ \cdots & -0.77s - 38 & 0 & -0.027 & 0.77 \\ \cdots & 0.65s + 64 & 0 & -0.27 & -0.65 \end{bmatrix}$$

Since $N_{M_s}(s)$ has three rows, $\mathcal{N}_L(M_s(s))$ has dimension 3 and that $N_{M_s}(s)$ has row-degrees $\{1, 1, 2\}$. This means that there exists exactly three linearly independent residual generators and a residual generator of minimal order has 1 state. The first residual generator in this example is formed in accordance with (3.7) as:

$$Q_1(s) = \frac{1}{1 + s/3}[1\ 1\ 0]N_{M_s}(s) \tag{3.29}$$

The parameterization $\phi(s)$ from (3.6) is here chosen, ad-hoc, to [1 1 0]. For the second design, the dimension of $\mathcal{N}_L(M_s(s))$ is also three and the row-degrees of $N_{M_s}(s)$ is also $\{1, 1, 2\}$ and a similar design as (3.29) is performed to form $Q_2(s)$. Thus, two first order residual generators was designed, adding low-pass dynamics $c(s) = 1 + s/3$.

In the simulations, control variables u_2 and u_3 is chosen to be zeros while u_1 is chosen according to Figure 3.9. The time response of the residual generators

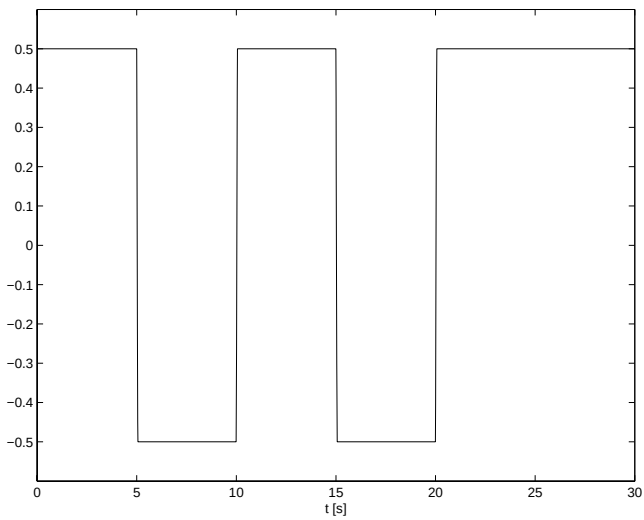
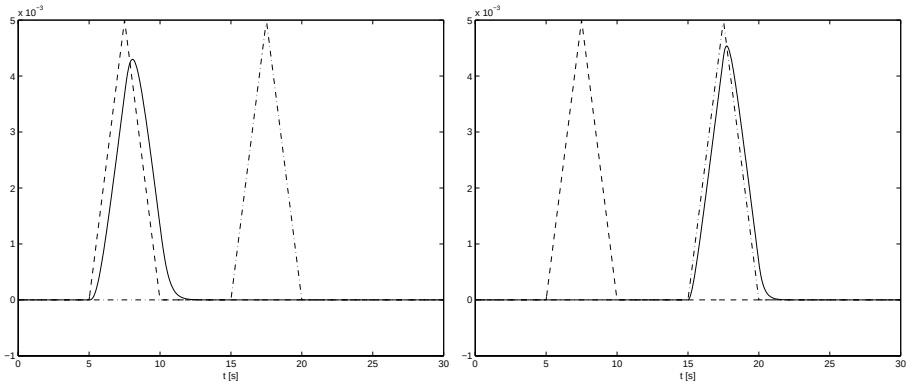


Figure 3.9: Control signal u_1 during simulations.

is shown in Figure 3.10 and it is clear that the residual generators meet the specifications. The work (Hou, 2000) is a chapter in the book (Patton et al., 2000) which is an edited book, collecting state of the art methods and algorithms as of year 2000. It is then interesting to make a few comparisons with the design made there. The design made by Hou resulted in two 4:th order observers while with the minimal polynomial basis approach, it was straightforward to find two first order residual generators meeting the design specifications. Also, since this is a quite large example (11 states), some numerical performance of the design algorithm can be evaluated. Figure 3.10 shows that the residual has no influence from control signals and the decoupled faults. Computing the norm



(a) The solid line is residual r_1, the dashed fault f_1 and the dash-dotted fault f_2. Residual r_1 is zero in fault-free operation and is sensitive to f_1 and insensitive to f_2 according to specifications.

(b) The solid line is residual r_2, the dashed fault f_1 and the dash-dotted fault f_2. Residual r_2 is zero in fault-free operation and is sensitive to f_2 and insensitive to f_1 according to specifications.

Figure 3.10: Time-responses of the residual generators.

of the transfer functions from u and d to r, which ideally should be 0, gives that $\|Q(s)M(s)\| \approx -200$ dB which is close to machine precision indicating a successful design. In the design made by Hou, severe numerical problems is evident since the residual fluctuates notably in fault free operation and influence from u_1 is clearly visible in the residuals. This indicates that the design problem studied is a numerically demanding problem for which the algorithm presented here produces a small and feasible design.

3.7.3 Non-zero initial states

Previously, considering state-space descriptions, initial conditions has been neglected in the design and this works under the, very reasonable, assumption that only stable residual generators are considered. In such cases, influence from the unknown initial conditions will disappear exponentially. However, for descriptor systems this is not always so which is shown by the following small example.

Example 3.1
Consider an equation stating that the derivative of an input equals the derivative of the output.

$$\dot{y} = \dot{u}$$

A descriptor formulation of this model equation is given by

$$\begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \dot{x} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u$$
$$y = x_1$$

Computing the basis $W(s)$ according to Theorem 3.7 gives the expected result

$$N_{M_s}(s)P = [s \ \ -s]$$

Here it is immediate that the basis $[s \ \ -s]$ is not irreducible since it loses rank for $s = 0$. It would be tempting to obtain a lower-degree basis by making it irreducible by factoring out the polynomial s and then form a *stable* residual generator

$$Q(s) = \frac{1}{s+1}[1 \ \ -1]$$

But with initial conditions $y(0) = y_0$, $u(0) = u_0$, and a Laplace limit theorem we would get

$$\lim_{t\to\infty} r(t) = \lim_{s\to 0} \frac{s}{s+1}(y-u) = \lim_{s\to 0} \frac{1}{s+1}(sy - su) =$$
$$= \lim_{s\to 0} \frac{1}{s+1}(y_0 - u_0) = y_0 - y_0$$

Thus, if $y(0) \neq u(0)$, the influence from initial conditions will not vanish.

For a general treatment considering arbitrary initial conditions $x(0) = x_0$, perform a single-sided Laplace transform of (3.23). Description (3.25) then transforms into

$$M_s(s) \begin{pmatrix} x \\ d \end{pmatrix} = P \begin{pmatrix} y \\ u \end{pmatrix} - \begin{bmatrix} 0 \\ E \end{bmatrix} x_0$$

Now, assume that $N_{M_s}(s)P$ is made irreducible with a full-rank invertible transformation $T(s)$, and then a residual generator is formed according to (3.28). The internal form of the residual generator is then in fault-free operation

$$r = c^{-1}(s)\phi(s)W_{\min}(s) \begin{pmatrix} y \\ u \end{pmatrix} = c^{-1}(s)\phi(s)T^{-1}(s)N_{M_s}(s)P \begin{pmatrix} y \\ u \end{pmatrix} =$$
$$= \varphi(s)T^{-1}(s)N_{M_s}(s) \begin{bmatrix} 0 \\ E \end{bmatrix} x_0$$

Now, it is clear that if $T(s)$ has zeros in the closed right half-plane, the influence from the initial state x_0 will not vanish (or even grow exponentially). This could not happen in the state-space case if, according to Theorem 3.1, the pair $\{A, [B_u B_d]\}$ was controllable since $N_{M_s}(s)$ then would be irreducible to begin with. For non R-controllable descriptor formulations, we can not guarantee

irreducibility of $N_{M_s}(s)P$. These issues are dealt with in detail in (Frisk and Nyberg, 2002) for a more general class of linear system descriptions; systems described by differential-algebraic linear equations. The general rule is briefly that if the greatest left divisor of $N_{M_s}(s)P$ is strictly stable, it is safe to make the residual generator irreducible, otherwise not. Also, a non row-reduced residual generator is always safe to make row-reduced since this can be made by a unimodular transformation $T(s)$ (which by definition is stable since it has no finite zeros).

3.7.4 Links to observer design

As was discussed in Chapter 2, in the non-descriptor case close relations to observers exists. To briefly point out a close link between observers and residual generators also in the descriptor case, consider Equation (3.25) which form a straightforward link for polynomial design of state observers for descriptor systems. Assume that no unknown inputs are present and that (3.23) is R-observable. Then, $M_s(s)$ is given by $[C^T \ -(sE^T - A^T)]^T$ which has full row-rank for all s. According to Theorem 3.B.11, matrices C and $-sE + A$ are right co-prime which together with Theorem 3.B.10 gives that there exists a matrix $H(s)$ such that

$$H(s)\begin{pmatrix} C \\ -(sE - A) \end{pmatrix} = I$$

Then, equation (3.25) directly gives that a state-observer is given by

$$\hat{x} = c^{-1}(s)H(s)P\begin{pmatrix} y \\ u \end{pmatrix}$$

where $c^{-1}(s)$ determines the dynamics, ensures stability, and properness of the observer. In case there are unknown input signals you may have to settle for estimating a set of (less than number of states) linear combinations of state-variables. This is e.g. achieved by left multiplication of (3.25) with a basis for the left null-space of $[D_d^T \ B_d^T]^T$ and treating the result as the no-disturbance case.

3.8 Conclusions

Residual generation for systems described by deterministic models on transfer function, state-space, or descriptor form has been considered. The design/analysis algorithms described in this chapter is formulated by using polynomial matrices as a fundamental mathematical object, rather than constant matrices commonly used. This, together with the existence of high performance computational tools for polynomial matrices, enables the use of polynomial matrix theory not only as a theoretical tool, but also as a design tool. Further, this makes it possible to state a simple and straightforward design algorithm. The

computational procedure is based on standard operations on polynomial matrices and no diagnosis specific code is necessary. The design procedure consists of three steps,

1. Compute a basis for the left null-space of a matrix which is directly given by the model equations.

2. Select the free design variables, i.e. dynamics of the residual generator and which consistency relations to utilize

3. Realize the residual generator on state-space form for implementation

which can be written in about two lines of Matlab-code.

An important property of the algorithm is that it, with a minimum number of parameters, parameterizes all possible consistency relations and residual generators. Also, the polynomial framework proves beneficial when performing e.g. order analysis of residual generators/consistency relations. Upper and lower bounds for orders of consistency relations is given based on number of states, sensors, disturbances, and observability indices of the model.

Also, the use of polynomial matrices and the simple design specification makes it possible to extend the algorithm to directly apply to not only standard state-space models but also transfer function and descriptor models. Later, in Chapter 5, the algorithm is extended to include also stochastic state-space and stochastic descriptor models.

The examples included showed important properties of the approach. Methodology of design was presented in a first example, where also interpretations of available design freedom were shown, e.g. in the residual structure choice, detectability analysis, and minimality issues. Implications of minimal order residual generators were particularly clear in the second turbo jet-engine example where a 4:th order filter was found with the polynomial approach, instead of a 26:th order filter that would have been found with a method neglecting the minimality issue. Finally, the third example, taken from (Hou, 2000), shows how the algorithm is directly applicable also to descriptor models. Numerical performance of the algorithm is clearly demonstrated in both the turbo jet-engine example and in the descriptor example, where the algorithm produced a feasible design where the algorithm in (Hou, 2000) clearly experiences numerical difficulties.

Appendix

This appendix is intended to serve as a compilation of definitions, theorems, and basic properties of linear systems, polynomial matrices, and polynomial bases for rational vector spaces used in this work. Sources describing these matters in detail are e.g. (Forney, 1975; Kailath, 1980; Chen, 1984) for control oriented views, and (Lancaster and Tismenetsky, 1985) for a mathematical presentation.

3.A Standard notions from linear systems theory

Definition 3.A.2 (normal rank). *The normal rank of a polynomial matrix* $A(s) \in \mathbb{R}^{m\times n}[s]$ *is defined as e.g.*[1]

1. *the number*
$$\textit{rank } A(s) = \max_{s\in\mathbb{C}} \textit{rank } A(s)$$
2. *the number of linearly independent columns (or rows) of* $A(s)$
3. *the number of invariant polynomials of the Smith form of* $A(s)$
4. *The dimension of the space spanned by the rows/columns*

The normal rank is sometimes also referred to as generic rank.

Definition 3.A.3 (rank of a rational matrix). *The rank of a matrix* $M(s) \in \mathbb{R}^{m\times n}(s)$ *is defined as e.g.*

1. *number of linearly independent rows/columns with rational coeffecients*
2. *the (normal) rank of* $N(s)$ *where* $N(s)$ *is the numerator in a left or right MFD of* $M(s)$
3. *The dimension of the space spanned by the rows/columns*

Definition 3.A.4 (matrix pencil). *A polynomial matrix* $M(s)$ *of degree* 1 *is called a matrix pencil. Often, a matrix pencil is written as*

$$M(s) = sE + F$$

where E *and* F *are constant matrices.*

Definition 3.A.5 (regular matrix pencil). *A matrix pencil* $M(s)$ *is said to be regular if* $M(s)$ *is square and full (normal) rank.*

[1] There exists several other, equivalent, definitions in the literature, but these are the ones most suitable for this presentation.

Definition 3.A.6 (irreducible polynomial matrix). *A polynomial matrix* $M(s) \in \mathbb{R}^{m\times n}[s]$ *is* irreducible *if it has full rank for all* $s \in \mathbb{C}$.

Definition 3.A.7 (unimodular polynomial matrix). *A square, irreducible, polynomial matrix is called unimodular. A defining property of unimodular matrices is that they have* polynomial *inverses.*

Definition 3.A.8 (row-degree of a polynomial vector). *A polynomial vector* $v(s) \in \mathbb{R}^{1\times n}[s]$ *has row-degree* p *if*

$$p = \max_j \deg v_i(s)$$

Definition 3.A.9 (polynomial basis). *A polynomial basis for a rational vectors pace* $\mathcal{F}$ *is represented by a matrix* $M(s)$ *where the rows spans* $\mathcal{F}$ *and the order of the basis is defined as the sum of row-degrees of the rows of matrix* $M(s)$. *A* minimal polynomial basis *is thus a polynomial basis for* $\mathcal{F}$ *that minimizes this order.*

Definition 3.A.10 (leading row coefficient-matrix). *A matrix* $M(s) \in \mathbb{R}^{m\times n}[s]$ *with row-degrees* $\mu_1, \ldots, \mu_m$ *can always be written*

$$M(s) = S(s)M_{hr} + L(s)$$

where

$$S(s) = diag\,\{s^{\mu_j}, j = 1, \ldots, m\}$$
$$M_{hr} = the\ leading\ row\ coefficient\ matrix$$

The matrix $L(s)$ *denotes the remaining terms and is of row-degrees strictly less than those of* $M(s)$.

Definition 3.A.11 (row-reduced matrix). *A matrix* $M(s)$ *is said to be row-reduced if its leading row coefficient matrix is full rank.*

Definition 3.A.12 (relatively prime/coprime matrices). *Two polynomial matrices with the same number of columns(rows) are relatively right(left) coprime if all their greatest common right (left) divisors are unimodular.*

Definition 3.A.13 (R and C controllable descriptor systems). *A descriptor system*

$$E\dot{x} = Ax + Bu$$

is completely controllable (C-controllable) if one can reach any state from any initial state. The system is controllable within the set of reachable states (R-controllable) if one can reach any state in the set of reachable states from any admissible initial state.

Definition 3.A.14 (Sylvester matrix). *Let $W(s)$ be a polynomial matrix of degree d, i.e.*

$$W(s) = \sum_{i=0}^{d} W_i s^i$$

with $W_d \neq 0$. Then, the q:th order Sylvester matrix for $W(s)$ is defined as

$$sylv(W(s), q) \triangleq \begin{bmatrix} W_0 & W_1 & \cdots & W_d & 0 & 0 & 0 \\ 0 & W_0 & W_1 & \cdots & W_d & 0 & 0 \\ & & \ddots & & & \ddots & \vdots \\ 0 & 0 & & & & & \\ 0 & \cdots & 0 & W_0 & W_1 & \cdots & W_d \end{bmatrix}$$

3.B Standard results from linear systems theory

Theorem 3.B.1 (Kailath,1980; Theorem 6.5-10). *The rows of a matrix $F(s)$ form a minimal polynomial basis for the rational vector space they generate, if and only if $F(s)$ is irreducible and row-reduced.*

Theorem 3.B.2. *If the rows of $F(s)$ form an irreducible polynomial basis for a rational vector space $\mathcal{F}$, then all polynomial row vectors $x(s) \in \mathcal{F}$ can be written $x(s) = \phi(s)F(s)$ where $\phi(s)$ is a* polynomial *row vector.*

Proof. Assume $x(s) \in \mathcal{F}$, then there exists co-prime polynomials $n(s)$ and $d(s)$ such that

$$d(s)x(s) = n(s)F(s)$$

Now, assume that $d(s)$ has a zeros at $s = s_0$. Then it holds that

$$0 = n(s_0)F(s_0)$$

But since $n(s)$ and $d(s)$ were co-prime, $n(s) \neq 0$ which in turn means that $n(s_0)$ lies in the left null-space of $F(s_0)$. However, this is a contradiction since $F(s)$ was assumed irreducible. This means that $d(s)$ can not have any zeros, i.e. $d(s)$ is proven to be a constant which ends the proof. ■

Remark: Even though the basis vectors are polynomial, the vector-space they generate is *not* a polynomial vector space. This is because there can not exist such a thing as a polynomial vector space. A standard, formal definition on a vector space from (Lancaster and Tismenetsky, 1985) read as follows: Let $\mathscr{S}$ be a set on which a closed binary operation $(+)$ (like vector addition) is defined. Let $\mathscr{F}$ be a field and let a binary operation (like scalar multiplication) be defined from $\mathscr{F} \times \mathscr{S}$ to $\mathscr{S}$. If for there exists an additive zero, additive inverse, and the operations obey distributive and associative laws, then $\mathcal{S}$ is a vector-space. However, since polynomials have no polynomial multiplicative inverse, the set of polynomials can not be a field, only a ring. However, with a suitable basis, we can still characterize all polynomial *elements* in the rational vector-space.

To illustrate the concept of rational vector-spaces and polynomial bases, the following example has been included.

Example 3.2
Let the rows of the matrix $F(s)$ be a basis for the rational vector-space $\mathcal{F}$.

$$F(s) = \begin{bmatrix} s & 0 & 1 \\ 1 & 1 & 0 \\ 0 & -s & 2 \end{bmatrix}$$

It is clear that $F(s)$ is a basis since $\det(F(s)) = s \not\equiv 0$, i.e. the matrix has full rank and therefore, the rows are linearly independent. Any polynomial vector of dimension 3 will of course belong to $\mathcal{F}$. Consider for example the vector

$$b_1(s) = \begin{bmatrix} s & 0 & 0 \end{bmatrix} \in \mathcal{F}$$

This vector can be written as a linear combination of the columns as follows:

$$b_1(s) = \begin{bmatrix} 2 & -s & -1 \end{bmatrix} \begin{bmatrix} s & 0 & 1 \\ 1 & 1 & 0 \\ 0 & -s & 2 \end{bmatrix} = x(s)F(s)$$

Here, $x(s)$ happens to be a polynomial vector. In general however, rational vectors are needed. Consider for example the vector

$$b_2(s) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{2}{s} & -1 & -\frac{1}{s} \end{bmatrix} \begin{bmatrix} s & 0 & 1 \\ 1 & 1 & 0 \\ 0 & -s & 2 \end{bmatrix} = x(s)F(s)$$

In this case, $x(s)$ is rational and there exists no *polynomial* $x(s)$ such that $b_2(s) = x(s)F(s)$.
If the polynomial basis is irreducible, then according to Theorem 3.B.2, only polynomial $x(s)$:s are needed. An irreducible, but not row-reduced, basis for the same vector-space $\mathcal{F}$ is for example

$$F'(s) = \begin{bmatrix} -2 & -1 & 0 \\ 1 & 1 & 0 \\ s & 0 & 1 \end{bmatrix} = T^{-1}(s)F(s)$$

which is achieved by extraction of the greatest left divisor $T(s)$ of $F(s)$. Now $b_2(s)$ can be written

$$b_2(s) = [-1 \ -1 \ 0]F'(s)$$

Theorem 3.B.3 (Kailath,1980;p. 366). *The PBH Rank Tests*

1. *A pair* $\{A, B\}$ *will be controllable if and only if the matrix*

$$[sI - A \quad B] \text{ has full-rank } n \text{ for all } s \in \mathbb{C}$$

2. *A pair* $\{C, A\}$ *will be observable if and only if the matrix*

$$\begin{bmatrix} C \\ sI - A \end{bmatrix} \text{ has full-rank } n \text{ for all } s \in \mathbb{C}$$

Theorem 3.B.4 (Kronecker Canonical Form of a matrix pencil). *For any linear matrix pencil* $A - sB$*, it is possible to find constant, square, and nonsingular matrices* U *and* V *such that*

$$U(A - sB)V = \text{block-diag}\{L_{\mu_1}, \ldots, L_{\mu_\alpha}, \widetilde{L}_{\nu_1}, \ldots, \widetilde{L}_{\nu_\beta}, sJ - I, sI - F\}$$

where

1. F *is in Jordan form*
2. J *is a nilpotent Jordan matrix*
3. $\widetilde{L}_\nu$ *is a* $(\nu + 1) \times \nu$ *matrix of the form*

$$\underbrace{\left.\begin{bmatrix} -s & & & \\ 1 & -s & & \\ & 1 & & \\ & & \ddots & \\ & & & -s \\ & & & 1 \end{bmatrix}\right\}}_{\nu} \nu + 1$$

4. $L_\mu = \widetilde{L}_\mu^T$

The $\{\nu_i\}$ *and* $\{\mu_i\}$ *are called left and right Kronecker indices.*

Theorem 3.B.5 (Kailath,1980;Theorem 6.3-13). *Let* $D(s)$ *be a polynomial matrix of full row-rank, and for any polynomial vector* $p(s)$*, let*

$$q(s) = p(s)D(s)$$

Then, $D(s)$ *is row-reduced if and only if*

$$deg\ q(s) = \max_{i:p_i(s)\not\equiv 0} [deg\ p_i(s) + \mu_i]$$

where $p_i(s)$ *is the i:th entry of* $p(s)$ *and* μ_i *is the degree of the i:th row of* $D(s)$*. This result is also called the predictable-degree property of row-reduced matrices.*

Theorem 3.B.6 (Kailath,1980; Theorem 6.3-12). *Let* $N(s) \in \mathbb{R}^{n\times m}$ *and* $D(s) \in \mathbb{R}^{n\times n}[s]$ *be full-rank and row-reduced. Then the solution* $H(s)$ *to the matrix equation*

$$H(s)D(s) = N(s)$$

is proper if and only if each row of $N(s)$ *has row-degrees less than or equal to corresponding row in* $D(s)$*.*

Theorem 3.B.7 (Division Theorem for Polynomial Matrices; Theorem 6.3-15 in Kailath,1980.). *Let $D(s)$ be an $m \times m$ nonsingular polynomial matrix. Then, for any $p \times m$ polynomial matrix $N(s)$, there exists unique polynomial matrices $\{W(s), R(s)\}$ such that $N(s) = D(s)W(s)+R(s)$ and $D^{-1}(s)R(s)$ strictly proper. Matrices $W(s)$ and $R(s)$ is called the polynomial matrix quotient and remainder of $D^{-1}(s)N(s)$.*

Theorem 3.B.8 (R-controllability;Yip and Sincovec,1981). *A descriptor model*

$$E\dot{x} = Ax + Bu$$

is R-controllable if and only if the matrix $[(sE - A)\ B]$ has full row-rank for all $s \in \mathbb{C}$.

Theorem 3.B.9 (C-controllability;Yip and Sincovec,1981). *A descriptor model*

$$E\dot{x} = Ax + Bu$$

is C-controllable if and only if the matrix $[(sE - A)\ B]$ has full row-rank for all $s \in \mathbb{C}$ and matrix $[E\ B]$ has full row-rank.

Theorem 3.B.10 (Bezout identity). *Polynomial matrices $D(s)$ and $N(s)$ will be right co-prime if and only if there exists polynomial matrices $X(s)$ and $Y(s)$ such that*

$$X(s)D(s) + Y(s)N(s) = I$$

Theorem 3.B.11 (Rank criterion for relative primeness). *Polynomial matrices $N(s)$ and $D(s)$ will be right coprime if and only if matrix*

$$\begin{bmatrix} D(s) \\ N(s) \end{bmatrix}$$

has full rank for all $s \in \mathbb{C}$. Similarly, $N(s)$ and $D(s)$ will be left coprime if and only if matrix

$$\begin{bmatrix} D(s) & N(s) \end{bmatrix}$$

has full rank for all $s \in \mathbb{C}$.

3.C State-space matrices for descriptor example

$$E = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 18.75 & -7.95 & 7.95 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -7.95 & 31.82 & -26.82 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 7.95 & -26.82 & 26.82 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

$$A = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 1 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 1 & \cdots \\
-68.7 & 98.96 & -77.74 & -601.16 & 43.92 & -107.77 & \cdots \\
45.23 & -402.43 & 337.54 & -906.97 & -177.27 & 179.24 & \cdots \\
4.48 & 339.82 & -219.17 & 697.11 & 149.56 & -360.37 & \cdots \\
1 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 1 & 0 & 0 & 0 & \cdots \\
0 & -1 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots
\end{bmatrix}$$

$$\begin{bmatrix}
\cdots & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 \\
\cdots & 1 & 0 & -68.83 & -34.83 & -6.41 \\
\cdots & 0 & 0 & 280.46 & -58.29 & 24.22 \\
\cdots & 0 & 1 & -236.89 & 48.76 & -69.93 \\
\cdots & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 \\
\cdots & -1 & 0 & 0 & 0 & 0 \\
\cdots & 0 & -1 & 0 & 0 & 0
\end{bmatrix}$$

$$B_u = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}^T$$

$$C = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 & 0 & 0 & -36.334 & 76.914 & -76.914 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$$

4

Residual Generation Based on Non-Linear Models

The perhaps most common approach to residual generation for non-linear models is to use state-observers based on a state-space description of the process dynamics. However, as was discussed in Section 2.2, consistency relations is an equally valid notion for non-linear systems and have been used in a number of works for supervising non-linear systems e.g. (Krishnaswami et al., 1994; Krishnaswami and Rizzoni, 1994; Guernez et al., 1997; Zhirabok, 1999; Zhang et al., 1998). In the linear case, consistency relations proved to be beneficial in the analysis and design of residual generators. Therefore, in this chapter, a systematic design procedure for non-linear system is outlined, exhibiting strong similarities with the linear design method described in Chapter 3 and with similar design freedom. The goal, to state a constructive design algorithm for non-linear systems with computer support is rather ambitious and not yet realistic. To still reach for this goal, the class of non-linearities is restricted and only a class of nonlinear systems is considered, systems described by polynomial differential-algebraic equations.

Section 4.1 describes the non-linear design problem and defines the class of models that is considered, polynomial differential-algebraic systems. Then, basic notation from standard elimination theory, needed for the design procedure, is introduced in Section 4.2 and the computational tool is exemplified. Section 4.3 describes a systematic design procedure for non-linear consistency relations and Section 4.4 addresses the problem on how to compute a residual based on the consistency relation. Section 4.5 proceeds to describe how isolability analysis can be performed on models in this class of systems. One major issue that, even for moderately sized models, becomes a problem is the computational

complexity of the design procedure. Section 4.6 shows how structural analysis can be used to manage this complexity. Finally in Section 4.7, the proposed design procedure is applied to a non-linear model in a noisy environment.

4.1 Problem formulation

The model is a set of *polynomial* differential-algebraic equations:

$$g_i(\bar{y}, \bar{u}, \bar{d}, \bar{f}, \bar{x}) = 0 \qquad i = 1, \ldots, p \tag{4.1}$$

where u is the control vector, y the measurement vector, d disturbance vector, f the fault vector, and x a vector of unknown internal states. The notation $\bar{y}$ denotes y, $\dot{y}$, $\ddot{y}$, ... and $\bar{u}$, $\bar{d}$, $\bar{f}$, $\bar{x}$ correspondingly for u, d, f, and x. The type of non-linearities considered is polynomials, i.e. the g_i in (4.1) are polynomials in $\bar{y}$, $\bar{u}$, $\bar{d}$, $\bar{f}$, and $\bar{x}$.

The design problem is, based on the model description (4.1), to find a consistency relation that satisfies the design specifications and then form a realizable residual generator based on that consistency relation. As before, all faults that are to be decoupled and any modeled disturbances are collected in a vector d, and the faults we wish to detect are collected in a vector f. Thus, given a set of model equations (4.1), find a computable quantity r that is a function of $\bar{y}$ and $\bar{u}$ only, such that when $\bar{f} = 0$ it holds that

$$r(\bar{y}, \bar{u}) = 0 \quad \forall \bar{d}, \bar{x}$$

4.1.1 Elementary functions as polynomials

The restriction to polynomial non-linearities is not as restrictive as it may seem, many non-polynomial non-linearities can be rewritten in polynomial form, e.g.

$$y = \sin u \quad \Leftrightarrow \quad \dot{y}^2 - \dot{u}^2(1 - y^2) = 0$$

These two forms are not really strictly equivalent, there exists a number of solutions that satisfies the right hand side but not the left hand side. For example, all solutions on the form $y(t) = \sin(u(t) + c)$ where c is any constant, is a solution to the differential equation on the right hand side. In (Lindskog, 1996) a translation table is included, where a polynomial description is provided for all elementary functions. This is done by describing the function as the solution of a polynomial differential equation. More formally stated; the elementary functions are differentially algebraic and since many of the commonly used functions e.g. trigonometric, inverse trigonometric, exponential, logarithmic functions are elementary functions, quite general systems can be handled within the framework of polynomial systems. Also, any smooth non-linearity can be described by polynomials to an arbitrary accuracy by Taylor expansion.

4.2 Basic elimination theory

This section will introduce some notation and a theorem from basic elimination theory that is needed for the design procedure in Section 4.3. It will not describe elimination theory other than with a small example. For more details, see e.g. (Cox et al., 1991).

Let $k[x_1, \ldots, x_n]$ denote the set of polynomials in variables $x_1, \ldots, x_n$ with coefficients in k, e.g.

$$x_1 x_2 + x_3^3 - 2x_1^2 x_2 x_3 \in \mathbb{R}[x_1, x_2, x_3]$$

An important concept, *ideal*, is now defined:

Definition 4.1 (Ideal). *Let $g_1, \ldots, g_s$ be polynomials in $k[x_1, \ldots, x_n]$. Then denote*

$$I =< g_1, \ldots, g_s >= \left\{ \sum_{i=1}^{s} h_i g_i : h_i \in k[x_1, \ldots, x_n] \right\}$$

I is called the ideal *generated by the polynomials $g_1, \ldots, g_s$.*

This means that I is the set of all linear combinations of the polynomials g_i with polynomial coefficients h_i. It is helpful to think of ideals using an analogy with subspaces from linear algebra. Both ideals and subspaces are closed under addition and multiplication, except that for subspaces we multiply with scalars but for ideals we multiply with polynomials. However, the analogy should not be taken too far. For example, in linear algebra a basis always consists of linearly independent vectors whereas for an ideal a basis is only concerned with spanning, independence is not mentioned. This is natural since it is in fact easy to prove that for any two polynomials g_i and g_j, zero can be written as a linear combination with polynomial coefficients. Thus, if f is an element in $< g_1, \ldots, g_s >$ with coefficients h_i, i.e.

$$f = \sum_{k=1}^{s} h_i g_i \tag{4.2}$$

the coefficients h_i is not unique.

Connections to the linear case

Expression (4.2) indicates relations with polynomial elements in rational vector spaces which appeared when deriving linear consistency relations in Chapter 3. If the rows of $F(s)$ span a rational vector-space $\mathcal{F}$, a vector $f(s) \in \mathcal{F}$ if and only if there exists co-prime polynomials $n(s)$, $d(s)$ such that $d(s)f(s) = n(s)F(s)$. However, if $F(s)$ is irreducible, Theorem 3.B.2 gives that $f(s) \in \mathcal{F}$ if and only if there exists a polynomial $n(s)$ such that

$$f(s) = n(s)F(s)$$

This irreducibility property was important in equation (3.6) where all linear consistency relations were parameterized by the polynomial row-vector $\phi(s)$. This unique polynomial parameterization was only possible due to the fact that $N_M(s)$ was irreducible. The polynomials g_i in Definition 4.1 therefore share similar traits with irreducible polynomial bases from Chapter 3.

Main elimination theorem

Now the main theorem used in the design, the well known elimination theorem, is stated.

Theorem 4.1 (Elimination Theorem).
Let $I \subset k[x_1, \ldots, x_n]$ be an ideal and let G be a Gröbner basis of I with respect to lex order $x_1 \succ x_2 \succ \cdots \succ x_n$. Then, for every $0 \leq m \leq n$, the set

$$G_m = G \cap k[x_{m+1}, \ldots, x_n]$$

is a Gröbner basis of the m:th elimination ideal I_m

$$I_m = I \cap k[x_{m+1}, \ldots, x_n]$$

This means that all polynomials, where variables $x_1, \ldots, x_m$ have been eliminated, can be written as in Definition 4.1 where $g_1, \ldots, g_s$ are the polynomials in G_m. Now follows a small example showing how the elimination theorem can be used in practice.

Example 4.1
Consider the following set of equations

$$\begin{aligned} x^2 + y + z &= 1 \\ x + y^2 + z &= 1 \\ x + y + z^2 &= 1 \end{aligned}$$

Suppose we wish to eliminate all variables but z, then according to the elimination theorem, first compute a Gröbner basis for the ideal

$$I = < x^2 + y + z - 1, x + y^2 + z - 1, x + y + z^2 - 1 >$$

with lex order $x \succ y \succ z$. In a Mathematica session, this is done by:

```
In[1]:= F={-1+x^2+y+z, -1+x+y^2+z, -1+x+y+z^2};
        GroebnerBasis[F,{x,y,z}]

Out[2]= {-1 + x + y + z^2, y^2 + z-z^2-y,
         2yz^2 + z^4-z^2, z^2 -4z^3 + 4z^4-z^6}
```

Thus, according to the elimination theorem $I \cap k[z] =< z^2 - 4z^3 + 4z^4 - z^6 >$. Note the triangular structure of the Gröbner basis, the last polynomial is a function of z only, the second and third depends on y and z while the first depends on all three x, y, z.
Another useful property of the Gröbner basis is given by the extension theorem (Cox et al., 1991), which gives a procedure to determine all solutions to the polynomial equation system. First, solve for z in the fourth equation, then substitute the roots into the second and third polynomial to obtain possible solutions y. Finally, solve for possible x using the first equation by substituting possible y and z.

4.3 Design using elimination theory

The model is a set of polynomial equations on the form:

$$g_i(\bar{y}, \bar{u}, \bar{d}, \bar{f}, \bar{x}) = 0 \qquad i = 1, \ldots, p \tag{4.3}$$

The basic step in the design algorithm is to manipulate the model equations (4.3) such that a consistency relation is obtained where all disturbances $\bar{d}$ (including faults that are to be decoupled) and internal states $\bar{x}$ have been eliminated. This relation can then be used to form a residual generator where the computational form is a function of $\bar{y}$ and $\bar{u}$ only and the internal form is a function of $\bar{f}$ only. Calculating a Gröbner basis and using Theorem 4.1 provides a systematic procedure to perform such manipulations. All derivations in this section are made in the time-continuous case. However, corresponding results for the time-discrete case is immediate by exchanging the time differentiation operator with the time shift operator. A difference between the time-discrete and the time-continuous case occurs when the consistency relation is used to realize a residual generator. More comments on this topic are included in Section 4.4.

Gröbner bases is a *non-differential* tool, i.e. the differentiation operator is not considered and when applying Gröbner basis methodology to systems like (4.1), x and $\dot{x}$ are seen as two completely unrelated variables. Of course, the differentiation operator is essential when analyzing dynamic systems. To be able to use non-differential tools like Gröbner bases, the differentiation has to be done "by-hand". Thus, the model equations (4.3) is differentiated a number of times and a new, larger set of equations is obtained. When differentiating the model equations, new model equations are obtained. It is however difficult to know when to stop differentiating. For an n:th order linear system, n differentiations is enough to extract all information in the model equations. For non-linear systems, no such limit exists. You may even have to differentiate infinitely many times to extract all information. In e.g. (Jirstrand, 1998), the input-

output ideal of the first order system

$$\begin{aligned}\dot{x}_1 &= -x_2 x_1 \\ \dot{x}_2 &= 0 \\ y &= x_1\end{aligned}$$

is proven to be non-finitely generated[1]. Therefore, when using the method outlined here, residual generators and consistency relations up to a certain order are considered, equal to the number of times the model equations were differentiated.

Another approach is to use differential tools like Ritt's characteristic sets (Ritt, 1950) to perform the elimination. However, the differential theory is currently not as developed as the non-differential, and differential bases is an active research topic.

4.3.1 Algorithm outline

The first step in the design procedure is to obtain a set of equations consisting of the original equations and differentiated model equations. The next step is to compute the (reduced) Gröbner basis for the elimination ideal where $\bar{d}, \bar{x}$ has been eliminated. Denote this basis with

$$GB =< b_1, \ldots, b_r >$$

where b_i are polynomials in all variables but $\bar{d}$ and $\bar{x}$. This Gröbner basis GB means that any polynomial, analytical relation, inferred from the model equations (4.3) *without* using the differentiation operation, where the unknown signals are eliminated can be written as

$$\sum_{i=1}^{r} h_i b_i \tag{4.4}$$

for a set of polynomials h_i. Each of the r polynomials in GB, or any combination as in (4.4), can be used to form a consistency relation where the fault-free relation is

$$\sum_{i=1}^{r} h_i b_i \big|_{f_i=0} = 0$$

The h_i polynomials are design variables available to the designer and this choice should be seen as a non-linear equivalent to the design matrix $\phi(s)$ in (3.6) for the linear design case. These can be used e.g. to shape the fault response in the residual or select the residual structure, i.e. to get sensitivity in the residual to a desired set of faults.

[1] In the example, no input signal was included. The input-output ideal then consists of relations including y, $\dot{y}$, ..., only.

As usual, the consistency relations includes higher order derivatives of y and u which are not known. Using the consistency relation as a residual generator is, in this non-linear case, non-trivial in the lack of a general realization theory. This issue is further discussed in Section 4.4. If however, these higher order derivatives were known, the computational and internal form of a residual generator could be formed as:

$$r^{comp}(\bar{y}, \bar{u}) = \sum_{i=1}^{r} h_i b_i \Big|_{f_i=0} \tag{4.5}$$

$$r^{int}(\bar{f}, \bar{y}, \bar{u}) = \sum_{i=1}^{r} h_i b_i - r^{comp} \tag{4.6}$$

For examples on computational and internal forms from the upcoming design example, see page 80.

A design variable, apart from choosing h_i polynomials in (4.5), is the variable ordering when calculating the Gröbner basis. Different variable orderings can produce very different bases with highly different complexity and computing time. The issue of variable ordering is not pursued further here, see e.g. (Boege et al., 1986) for examples of heuristic "optimal" variable orderings. Here, a natural way to choose variable ordering can be stated as:

1. Order the variables to be decoupled as the highest ranked variables, i.e. the faults to be decoupled, the disturbances, and the unknown internal states should be ordered first when using Theorem 4.1.

2. Next, order the known variables and their derivatives. To get a consistency relation of as low order as possible, order high order time-derivatives higher than low-order time-derivatives.

3. Finally, order the faults not to be decoupled as the lowest ranked variables to get as good fault sensitivity properties in the consistency relations as possible.

Related approaches

Related work has been published by (Zhang et al., 1998) where consistency relations are derived using Ritt's algorithm (although, they could just as well have used Gröbner bases as described above). However, in the Zhang et al. (1998) paper, faults are not decoupled in the residual, only the state variables. The reason for this is that additional decoupling of fault signals results in higher order consistency relations, rendering a more difficult realization problem. Is also implies that fault isolation is not performed using structured residuals, instead local statistical properties of the consistency relations are utilized and fault isolation is performed using the so called "local approach" (Basseville et al., 1987; Basseville and Nikiforov, 1993). The gain using the local approach rather

than structured residuals for fault isolation is, as was noted above, an easier realization problem due to lower order derivatives. The prize to pay is fault isolation performance. Zhang et al. concludes "for large parameter changes, FD still works in most practical situations (and even becomes easier), but FI is likely to become incorrect"[2].

4.4 Realizable residual generator

Note that $\dot{y}$, $\dot{u}$ etc. normally appear in $r^{comp}(\bar{y}, \bar{u})$. Usually, these derivatives are not known. This problem can be handled by realization theory or approximations. Realization is exemplified below, but first some simple approximation methods are recalled.

A simple method to compute the residual is to approximate differentiated variables with

$$\hat{\dot{y}} = \frac{s}{sT_d + 1} y$$

with a suitably chosen T_d. This simple method may not be a satisfactory solution in many cases. Another, less noise sensitive, solution may be to first fit suitable analytical functions, e.g. cubic splines, to possibly low-pass filtered data. The derivatives can then be estimated by extracting the derivatives analytically from fitted analytical expressions.

Other possibilities than estimating the derivatives, not explored further here other than the brief discussions below, are realization theory and time-discretization of the time-continuous model.

Realization theory

In the linear case, it was straightforward to add stable linear dynamics to form a realizable residual generator based on a consistency relation where the added dynamics is free to choose as long as it is stable. Thus, it would be desirable to approach the non-linear problem in the same way, e.g. add stable (possibly linear) low-pass dynamics

$$r + \alpha_1 \dot{r} + \alpha_2 \ddot{r} = r^{comp}$$

and find an explicit state-space realization of the residual generator. Unfortunately, realization theory with inputs is difficult, even when only polynomial systems are considered (Forsman, 1991). Below, a small example where the realization step is immediate demonstrates the idea on how to use realization theory.

[2] Zhang et al. uses the abbreviations FD for fault detection and FI for fault isolation.

Example 4.2
Consider a system described by the differential equation

$$\begin{aligned}\dot{x} &= -\sin^3(x)(u+f)^2\\ y &= x + (u+f)\end{aligned}$$

where f is an actuator fault that has to be supervised. A consistency relation for the system above can easily be derived by differentiating the measurement equation and eliminating the state-variable x.

$$\dot{y} + \sin^3(y-u)u^2 - \dot{u} = h(y,u,f) \tag{4.7}$$

where

$$h(y,u,f) = \dot{f} - \sin^3(y-u-f)(u+f)^2 + \sin^3(y-u)^2u^2$$

Thus, if $\dot{y}$ and $\dot{u}$ were known, the left hand side of (4.7) could be used to compute a residual that could be used to detect the actuator fault.

$$\dot{y} + \sin^3(y-u)u^2 - \dot{u} = \begin{cases} 0 & f \equiv 0 \\ c(t) \not\equiv 0 & f \not\equiv 0^3 \end{cases}$$

Here, the time derivatives are assumed to be unknown. In the light of the previous discussion, add stable first-order linear dynamics to the left hand side of (4.7), i.e.

$$r + \alpha\dot{r} = \dot{y} + \sin^3(y-u)u^2 - \dot{u} \tag{4.8}$$

with $\alpha > 0$ and try to find an explicit state-space representation of (4.8) with y and u as inputs and the residual r as output. The choice of α corresponds to $c(s) = 1 + \alpha s$ in equation (3.7). In this particular case this is easy and a realization is e.g.

$$\begin{aligned}\dot{z} &= -\frac{1}{\alpha}z - \frac{1}{\alpha}(y-u) + \sin^3(y-u)u^2\\ r &= \frac{1}{\alpha}z + \frac{1}{\alpha}(y-u)\end{aligned}$$

The internal form for this filter is

$$r + \alpha\dot{r} = h(y,u,f)$$

which will be 0 in the fault-free case and non-zero when a fault occurs.

As noted above, general non-linear realization of input-output descriptions is a difficult task. However, there exist an important difference between the basic realization problem (Sadegh, 2001; Isidori, 1995; Sontag and Wang, 1990; der

[3] This is not entirely true, of course there exists particular solutions $f(t)$ to the differential equation $0 = h(y,u,f)$ and for these solutions, the residual will be 0.

Schaft, 1987) and using realization theory to form a computable residual generator. The dynamics of the residual generator is free for the designer to choose (as long as it is stable). This means that quite a bit of freedom exists that is normally not present when studying the realization problem. In the example, linear dynamics $r + \alpha\dot{r}$ was added. For the general problem, this can be any dynamics $f(r, \dot{r}, \dots)$ such that $r = 0$ is a globally stable operating point of the differential equation

$$f(r, \dot{r}, \dots) = 0$$

Realization of residual generators based on non-linear consistency relations is a promising topic where further study is needed.

Time-discretization

A second approach is to transform the original, time-continuous model, to a time-discrete model. Then, an analogous design can be made using the time-discrete polynomial model, which results in time-discrete consistency relations. These time-discrete consistency relations can be directly used as residual generators (or arbitrary low-pass dynamics can be added) since no time-differentiated signals occur, only time-delayed known signals.

4.5 Isolability analysis

An interesting and important topic in diagnosis is diagnosability analysis, e.g. deciding if it is theoretically possible to separate the modeled faults. This section serves as an introduction the isolability analysis problem and also to indicate how theory from related areas of research can be applicable.

For example, if all faults are modeled as constant parameters, isolability is very closely connected to *identifiability*, and results from the identification community can be directly applied. An example for polynomial systems is (Ljung and Glad, 1994), where Ritt's algorithm and characteristic sets are used for identifiability analysis of non-linear dynamical systems on polynomial form. Briefly stated, Ritt's algorithm can be used to compute a characteristic set for the considered differential ideal. In (Ljung and Glad, 1994) it is then shown that a necessary and sufficient condition for global identifiability (and here also for isolability) is that all fault parameters appears linearly in the characteristic set. Thus, the polynomials in the characteristic sets should appear like

$$P(\bar{y}, \bar{u})f_i + Q(\bar{y}, \bar{u}) = 0, \;\; i = 1, \dots, k_f$$

for all polynomials including the fault signals/parameters f_i. When doing isolability analysis, it is important to utilize all restrictions on control signals and model parameters available to get as precise answers as possible. In particular it is desirable to only consider *real* solutions and also to incorporate any inequality constraints available. The above result from the identification community

does not consider only real solutions. However, quantifier elimination is such an interesting tool where it is possible to do real algebra and incorporating inequality constraints. See e.g. (Jirstrand, 1997, 1998) for theory and applications of quantifier elimination for controller design/analysis. Here, isolability analysis by quantifier elimination is illustrated only briefly, mainly to illustrate a fruitful view of the isolability analysis problem. But first, a small general example on quantifier elimination is included to illustrate the principles of quantifier elimination.

Example 4.3
Consider the problem of deciding for which real a that the equation $x^2 + ax + 1 = 0$ has real solutions. This problem can be posed and solved by quantifier elimination

$$\exists x \in \mathbb{R}.\ x^2 + ax + 1 = 0 \Leftrightarrow |a| \geq 2$$

The above simplification is called quantifier elimination since variables, bound by quantifiers, is eliminated resulting in an equivalent expression, free of quantified variables.

For the above operation, constructive computational tools exist and are finding their way into commercial computational algebra packages[4]. Unfortunately, the complexity of quantifier elimination is huge, and at the moment only toy-sized examples can be handled. However, this view of the isolability analysis problem is appealing which will be demonstrated next on a small, static, scalar model.

Example 4.4
Consider the static model:

$$y(t) = (1 + f_1(t))u(t) + f_2(t)$$

It is immediate by observing the model equation that for any $f_1(t)$ there exists a $f_2(t)$ that can reproduce the observed behavior. The converse is not true since when $u(t) = 0$, no $f_1(t)$ can reproduce the output of any $f_2(t) \neq 0$. Thus, fault f_2 is isolable from f_1 but f_1 is not isolable from f_2. However, if we know that $u(t) > 0$, then the faults are not isolable from each other. Now, lets see how quantifier elimination can answer these questions. First, verify that f_1 is not isolable from f_2 by verifying that for all control signals and all f_1, there exists y and f_2 that are consistent with the model equation in both fault modes.

$$\forall u \forall f_1 \exists f_2 \exists y.\ \{y - (1 + f_1)u = 0 \wedge y - u - f_2 = 0\}$$

Eliminating the quantifiers in Mathematica yields `True`, i.e. f_1 is not isolable from f_2. Exchanging f_1 and f_2 gives

$$\forall u \forall f_2 \exists f_1 \exists y.\ \{y - (1 + f_1)u = 0 \wedge y - u - f_2 = 0\}$$

[4]The command resolve, which is used in the examples here, is included as "experimental" with version 4 of Mathematica.

which resolves to `False` which means that f_2 is isolable from f_1. If we want to know the exact condition on u, when a fault f_1 can explain the behavior caused by f_2, simplify the formula

$$\forall f_2 \exists f_1 \exists y.\ \{y - (1 + f_1)u = 0 \wedge y - u - f_2 = 0\}$$

which is equivalent to $u \neq 0$ as expected. Finally, if we know that $u(t) > 0$, then it is true that for any f_2 there exists an f_1 that produces the same observables. This is verified by the formula

$$\forall u \forall f_2 \exists f_1 \exists y.\ \{(u > 0) \rightarrow y - (1 + f_1)u = 0 \wedge y - u - f_2 = 0\}$$

which resolves to `True` when the quantifiers are eliminated.

4.6 Complexity management

Although strong computational support exists, the computation of a Gröbner basis is a time and memory consuming operation that, even for moderately sized problems, quickly becomes computationally intractable. The quantifier elimination procedure illustrated in the previous section is even more computationally demanding. Thus, some means of handling complexity is desired.

A promising approach, for the elimination problem, is to perform structural analysis (Staroswiecki and Comtet-Varga, 2001) on the model equations, identifying a subset of the model equations from which it is possible to derive a consistency relation. Since only a subset of the equations is used, a computationally smaller problem is identified. An example, with equations borrowed from (Persis and Isidori, 2001), is used to illustrate such a procedure.

Example 4.5
The example is a 4 state model with 3 measurements given by the following 7 equations:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
x_1^2 \dot{x}_2 &= x_1^3 x_4^2 - \theta_1 + \theta_2 x_1^2 u_1 + d x_1^2 \\
\dot{x}_3 &= x_4 \\
x_1 \dot{x}_4 &= -2 x_2 x_4 + \theta_2 (u_2 + f) \\
y_1 &= x_1 \\
y_2 &= x_3 \\
y_3 &= x_4
\end{aligned}$$

To derive a consistency relation, the equations need to be differentiated at least once. Differentiating all the equations and feeding the resulting 14 equations into Mathematica gives an intractable problem (the Gröbner basis computation

did not finish within 30 minutes on a standard PC). With a structural analysis of the model equations it is, under certain technical conditions, possible to extract subsets of the model equations where it is possible to derive a consistency relation. Krysander and Nyberg (2002) call these subsets of equations *structurally singular*. A structurally singular set of equations is basically m equations with, at most, $m-1$ unknown variables. A structurally singular set is said to be minimal if no proper subset is also structurally singular. In the table below, a structural description of the model equations is given where an 'x' indicates that the corresponding variable appears in the equation.

	d	x_1	x_2	x_3	x_4	y_1	y_2	y_3	u_1	u_2	f
$\star(1)$		x	x								
(2)	x	x	x		x				x		
(3)				x	x						
$\star(4)$		x	x		x					x	x
$\star(5)$		x				x					
(6)				x			x				
$\star(7)$					x			x			

From this table, it is clear that equations 1,4,5,7 (marked with $\star$) form a structurally-singular set. This is because these four equations only include 3 unknown variables (x_1, x_2, x_4). It is also a minimally structurally-singular set since any proper subset of these equations is not structurally singular. Thus, these 4 equations can be used instead of the complete model. Now, using these 4 equations, Mathematica delivers (in 0.02 seconds!) the first order relation:

$$y_1\dot{y}_3 + 2\dot{y}_1 y_3 - \theta_2 u_2 = \theta_2 f$$

which clearly can be used to detect the fault assuming that the realization problem can be solved. For more details on structural analysis and algorithms to find structurally singular sets, the interested reader is referred to (Krysander and Nyberg, 2002) and the references therein.

4.7 Simulation example: Coupled water tanks

The model used to illustrate the approach is two coupled water tanks, shown in Figure 4.1. The process is equipped with four sensors, two sensors measuring the water level in each tank and two sensors measuring the outflow of water from each tank. The process is controlled by a pump.

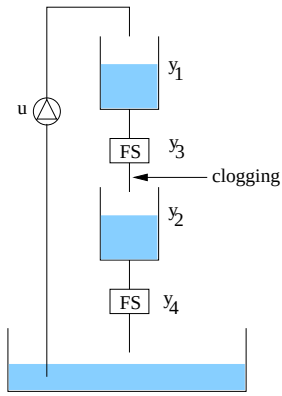


Figure 4.1: The simulation process: Coupled water tanks equipped with flow and water-level sensors.

4.7.1 Modeling

A first-principles model of the fault-free process, utilizing Bernoulli's law for the flows, is given by:

$$\begin{aligned} \dot{h}_1 &= a_1 u - a_2\sqrt{h_1} \\ \dot{h}_2 &= a_3\sqrt{h_1} - a_4\sqrt{h_2} \\ y_1 &= h_1 \\ y_2 &= h_2 \\ y_3 &= a_5\sqrt{h_1} \\ y_4 &= a_6\sqrt{h_2} \end{aligned} \tag{4.9}$$

where a_i are the model parameters, y_i the measurements, u the control signal to the pump, and h_i the height of water in each tank. The water level in the tanks can be between 0 and 10, i.e. $0 \leq h_1, h_2 \leq 10$.

The faults considered are faults in the actuator, sensors, and clogging in the pipe between the two water tanks at the point indicated by the arrow in Figure 4.1. Let f_1 denote an unknown additive fault on the actuator signal and f_2, f_3, f_4, f_5 additive faults on the four sensors. The clogging fault is modeled by f_6 where $f_6 = 1$ represents a completely clogged pipe and $0 < f_6 < 1$ represents partial clogging. Extending model (4.9) with fault models and introducing

auxiliary variables z_i to get a polynomial description gives:

$$\begin{aligned}
\dot{h}_1 &= a_1(u + f_1) - a_2(1 - f_6)z_1 \\
\dot{h}_2 &= a_3(1 - f_6)z_1 - a_4 z_2 \\
\dot{f}_i &= 0, \ i = 1, \ldots, 6 \\
z_1^2 &= h_1 \\
z_2^2 &= h_2 \\
y_1 &= h_1 + f_2 \\
y_2 &= h_2 + f_3 \\
y_3 &= a_5(1 - f_6)z_1 + f_4 \\
y_4 &= a_6 z_2 + f_5
\end{aligned} \tag{4.10}$$

The non-polynomial model (4.9) has been transferred into a polynomial (descriptor) model by adding the auxiliary variables z_i. Model (4.10) is not equivalent to (4.9) since both negative and positive z_i can satisfy (4.10), but $\sqrt{h_i} \geq 0$. Thus, the model specified by (4.10) covers (4.9) in fault-free mode. For the models to really be equivalent, conditions $z_i \geq 0$ has to be added, but since such inequalities can not be handled by the computational framework used here this issue is not pursued further. Note that this is not a problem in the design stage, if however we wish to answer e.g. detectability and isolability questions, these kind of issues need to be resolved.

Here, only constant faults are considered, i.e. $\dot{f}_i = 0$. Note that this assumption is not required by the approach, it is only made here to limit the size of the example.

Finally, the model equations can not be used on the form (4.10) since the mathematical tool used is non-differential. Thus, the static equations (last 6 equations in (4.10)) is differentiated. Therefore, differentiate the static equations k times, substitute for $\dot{f}_i$ and $\dot{h}_i$ using the dynamic equations, and collect the results. The higher order residual generators that is considered, the more of the dynamic model can be utilized. However, higher orders means a more difficult implementation problem according to the discussion in Section 4.4. In this design example, only first order residual generators are considered. Since there are 6 static equations in (4.10), the result is 12 equations on form (4.3).

4.7.2 Design

The object of the design is to find a set of residuals that form a fault isolating residual structure. To form a isolating influence structure, 6 elimination ideals (and the corresponding Gröbner bases), are calculated where one fault is eliminated in each ideal. Residual generators are then selected among the basis polynomials of the calculated Gröbner bases.

The design is performed as described in Section 4.3. The following variable

ordering is used when eliminating f_1:

$$f_1 \succ \{h_i\} \succ \{\dot{z}_i\} \succ \{z_i\} \succ \{\dot{y}_i\} \succ \{y_i\} \succ u \succ f_2 \succ f_3 \succ f_4 \succ f_5 \succ f_6$$

and corresponding ordering when eliminating the other variables. The variables $\dot{y}_i$ are given a high ordering since it is desirable to eliminate those variables to get simple computational forms of the residual generators.

Then, 6 consistency relations that creates a fault isolating structure is selected to form the residual generators. The computational form of the residual generators are:

$$\begin{aligned}
r_1^{comp} &= a_6^2 y_2 + y_3^2 - y_4^2 - a_5^2 y_1 \\
r_2^{comp} &= a_6^2 y_2 + a_2 a_5 y_3 + 2 y_3 \dot{y}_3 - y_4^2 - a_1 a_5^2 u \\
r_3^{comp} &= y_3^2 - a_5^2 y_1 \\
r_4^{comp} &= a_6^2 y_2 - y_4^2 \\
r_5^{comp} &= y_3^2 + 2 a_3 y_3 \dot{y}_2 + a_5 (a_4^2 y_2 - \dot{y}_2^2) - a_5^2 y_1 - a_3^2 a_5 y_1 \\
r_6^{comp} &= a_6^2 y_2 + a_1 a_5 u y_3 + a_6 (a_3 y_3 - a_5 \dot{y}_2) - 2 a_5 y_1 \dot{y}_3 - y_4^2 - a_4 a_5 y_4 - a_2 y_3^2
\end{aligned}$$

and the corresponding internal forms are:

$$\begin{aligned}
r_1^{int} =& f_4^2 + a_5^2 (f_2 - 1 + f_6^2 - (-2 + f_6) f_6 y_1) - 2 f_4 y_3 + \\
& + 2 f_5 y_4 - f_5^2 - a_6^2 f_3 \\
r_2^{int} =& 2 a_2 a_5 f_4 f_6 - 2 a_2 a_5 f_6 y_3 + a_2 a_5 f_6^2 y_3 + 2 f_5 y_4 - 2 f_4 \dot{y}_3 - \\
& - a_1 a_5^2 (f_1 - 1 + f_6^2 + (-2 + f_6) f_6 u) - \\
& - a_2 a_5 f_4 f_6^2 - f_5^2 - a_2 a_5 f_4 - a_6^2 f_3 \\
r_3^{int} =& a_5^2 (f_2 - 1 + f_6^2 - (-2 + f_6) f_6 y_1) + f_4 (f_4 - 2 y_3) \\
r_4^{int} =& 2 f_5 y_4 - f_5^2 - a_6^2 f_3 \\
r_5^{int} =& a_3^2 a_5 (f_2 - 1 + f_6^2 - (-2 + f_6) f_6 y_1) + \\
& + a_5^2 (f_2 - 1 + f_6^2 - (f_6 - 2) f_6 y_1) + f_4 (f_4 - 2 y_3) - 2 a_3 f_4 \dot{y}_2 - a_4^2 a_5 f_3 \\
r_6^{int} =& a_4 a_5 f_5 + 2 a_2 f_4 y_3 + 2 f_5 y_4 + 2 a_5 f_2 \dot{y}_3 - a_1 a_5 (f_4 u + f_1 (f_4 - y_3)) - \\
& - f_5^2 - a_2 f_4^2 - a_3 a_6 f_4 - a_6^2 f_3
\end{aligned}$$

By inspection of the internal forms, the influence structure can be concluded to be as in Table 4.1. All instances of fault variables in the internal forms has been marked by shaded boxes. A 1 in column i and row j of the table means that fault i ideally influences residual j. The first aim was to design residual generator r_i to be sensitive to all faults but f_i. This is possible, but at the cost of more complex residual generators. The design made here is made as simple as possible while keeping single-fault isolability, i.e. uniqueness of all columns

	f_1	f_2	f_3	f_4	f_5	f_6
r_1	0	1	1	1	1	1
r_2	1	0	1	1	1	1
r_3	0	1	0	1	0	1
r_4	0	0	1	0	1	0
r_5	0	1	1	1	0	1
r_6	1	1	1	1	1	0

Table 4.1: Influence structure

in Table 4.1. Detectability properties of the residual generators can be seen directly in the internal forms of the residual generators. For example, fault f_2 will be difficult to detect with residual r_6 since in case of a single fault f_2

$$r_6^{int} = 2a_5 f_2 \dot{y}_3$$

Thus, r_6 will only deviate from zero if y_3 is non stationary. In the simulation study that follows below, the process is regulated by a controller making water levels stationary making detection of f_2 using r_6 unreliable. This does not mean that f_2 can not be isolated, what it indicates is that the corresponding position in the decision structure need to be an X.

The computational forms of the residual generators can not be used directly since the time differentiated variables are not directly available and need to be estimated. In the simulations the process is subjected to noise and the time derivatives are estimated using the spline procedure outlined in Section 4.4. A three step procedure is used:

1. Low-pass filter the measurements and control signals.
2. Estimate cubic spline polynomials and extract $\hat{\dot{y}}_i$ from the estimated polynomials.
3. Compute the residuals according to r_i^{comp} and low-pass filter the residual again.

All residuals are also scaled such that a threshold 1 is used for all residuals.

4.7.3 Simulations

In the simulations, a simple proportional controller is used to control the water level in the upper tank to follow a square reference signal. All 4 sensors are subjected to rather high intensity measurement noise. Figure 4.2 shows the water levels in both tanks in a fault-free, but noisy, simulation. Noise-free simulations gives, as expected, ideal performance of the residual generators. Figure 4.3 shows the residuals in the fault-free case. All residuals are below the dotted thresholds. Here, only two fault scenarios are shown, a constant fault in

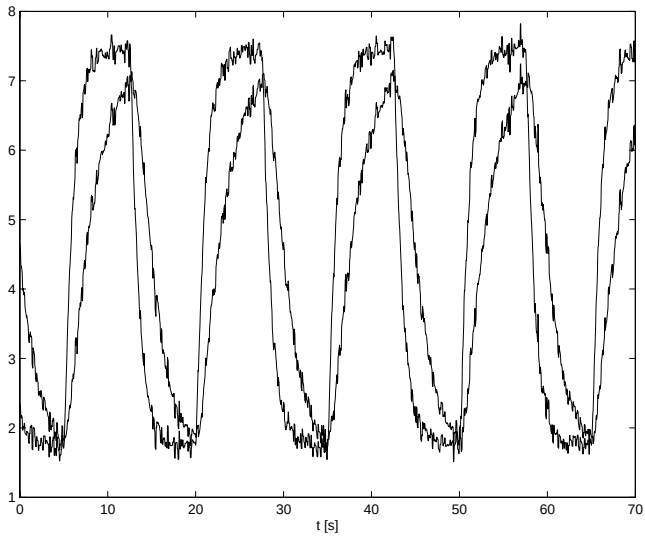


Figure 4.2: Water level in the upper tank, y_1, and the lower tank y_2, during fault-free simulations.

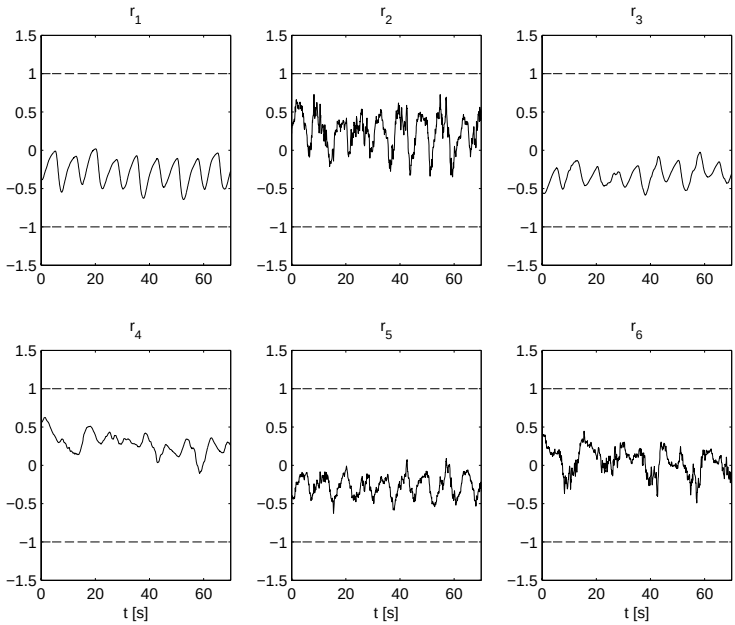


Figure 4.3: Residuals in the fault-free case. All residuals are below the dotted thresholds, i.e. no false alarms during the simulation.

the actuator and a fault in sensor y_2, measuring the water level in the second tank. Figure 4.4 shows the residuals when the actuator fault $f_1 = 0.1$ is induced at time $t = 40$ sec. It is clear that the residuals respond as expected by the influence structure in Table 4.1, i.e. residuals r_2 and r_6 respond to the fault while r_1, r_3, r_4, and r_5 does not. The fault is correctly isolated. Figure 4.5

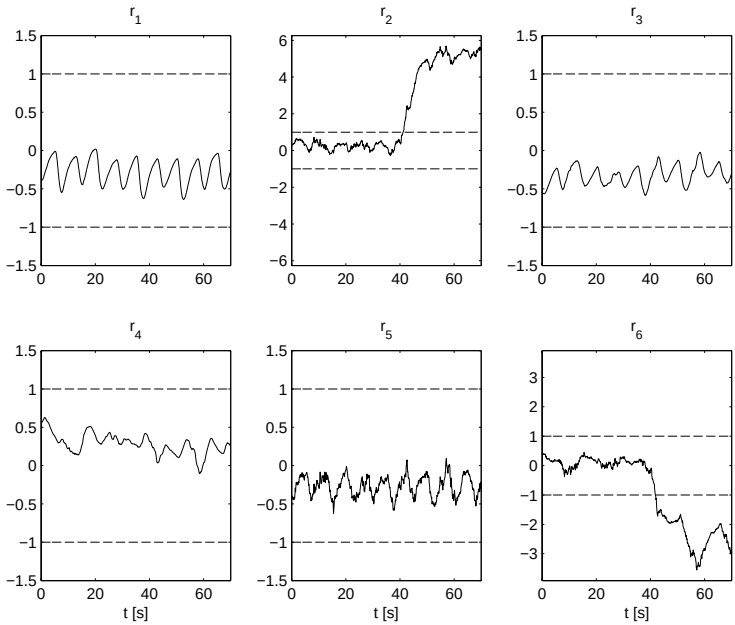


Figure 4.4: Residuals when fault $f_1 = 0.1$ is induced at time $t = 40$ sec. Residuals r_2 and r_6 respond to the fault while r_1, r_3, r_4, and r_5 does not, i.e. f_1 is correctly isolated according to Table 4.1.

shows the residuals when fault is sensor y_2 appears abruptly. The fault is also here induced at time $t = 40$ sec. Also here, the residuals respond according to the influence structure in Table 4.1, i.e. all residuals but r_3 respond to the fault. The fault is correctly isolated. Simulating the other faults gives similar results corresponding to the influence structure. Thus, for this simulation of a non-linear process, subjected to measurement noise the approach produced a feasible solution. The design was highly automated in Mathematica and the design choices were similar as in the linear case, i.e. mainly

- Choice of desired influence structure.
- Choice of consistency relations to realize the influence structure.
- Choice of low-pass dynamics to make residual generators realizable.

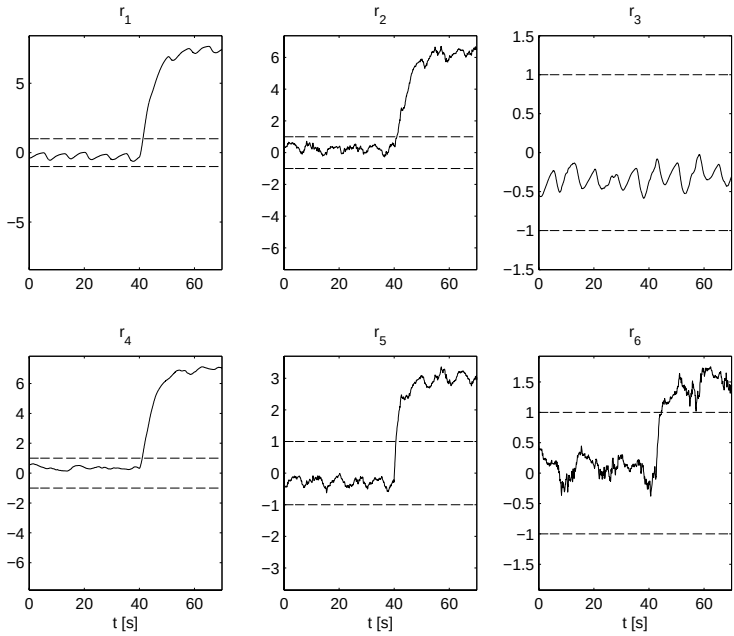


Figure 4.5: Residuals when fault $f_3 = 0.6$ is induced at time $t = 40$ sec. All residuals but r_3 respond to the fault which means that f_3 was correctly isolated.

4.8 Conclusions

A systematic and constructive design procedure for non-linear consistency relations has been developed with strong computational support in standard computer-algebra packages. Finding consistency relations is closely linked with variable elimination, and a suitable class of systems to consider is models described by polynomial, differential-algebraic equations. This is a suitable class since, for these types of non-linearities, elimination theory exists and practically any analytical expression can be restated in polynomial form.

A systematic approach for design and analysis of disturbance decoupling consistency relations is presented. The basic design step in the design procedure is to compute a Gröbner basis for an elimination ideal where all disturbances have been eliminated. A nice property of the approach is that the available design freedom is closely connected to the design freedom available in the linear case for which the design freedom is well understood. In the linear case, it was straightforward to use a consistency relation to form a realizable residual generator. The nonlinear case is more difficult and different aspects of this problem is discussed and approximate solutions are suggested in Section 4.4.

The strong theoretical support for polynomial systems also introduces some interesting possibilities for the future; isolability analysis is such a possibility. It is shown by example how advanced tools from real algebra, quantifier elimination, can be used for automatic isolability analysis. It is also noted how identifiability analysis by characteristic sets from differential algebra can be used for isolability analysis when only constant faults are considered.

Two major limitations of the approach exists, the computational complexity of computing Gröbner bases and secondly how to utilize the non-linear consistency relations to form a realizable residual generator. Computing Gröbner bases may become computationally intractable even for moderately sized problems. Section 4.6 exemplifies how structural analysis can be of great assistance in reducing the size of the problem. In an example, an intractable problem that could not be solved on a standard PC was reduced to a problem that could be solved in 0.02 seconds.

The approach is finally demonstrated on a small, but non-trivial, example model consisting of two coupled water tanks. The example shows how the design freedom can be used and how fault isolation properties of the model can be analyzed from the calculated Gröbner bases. The example also shows how both constant and time-varying faults are handled equally in the design process.

5

Residual Generation Based on Stochastic Linear Models

This chapter investigates residual generation in linear, stochastic systems. In Chapter 3, a design algorithm was developed for deterministic linear models based on polynomial methods. That algorithm will now be extended to cover also linear stochastic model descriptions.

A fundamental contribution to this problem is given by Nikoukhah (1994) where a class of residual generators, innovation filters, for models stated on state-space form were considered. The basic stochastic design requirement on an innovation filter is that the residuals should be zero mean and white in the fault-free case and that the whiteness property can be achieved without loosing *any* design freedom.

Here, the aim is to extend the polynomial methods that proved beneficial in the deterministic case to the stochastic case and address problems posed in (Nikoukhah, 1994) and also extend the problem formulation and solve a more general problem. In the more general problem formulation, the requirement on design-freedom is dropped which proves beneficial.

The use of polynomial theory facilitates, just as in the deterministic case, the development of an algorithm that covers not only state-space models, but also general descriptor models. The main algorithmic tool is *J-spectral cofactorization* which is shown to handle the stochastic problem. Algorithms for spectral factorization of polynomial matrices has recently received much attention since it plays a fundamental role in the solution of polynomial $\mathcal{H}_\infty$-(Green et al., 1990) and $\mathcal{H}_2$-(Kwakernaak, 2000b) standard problems. Therefore, feasible and numerically appealing algorithms and implementations has been proposed (Kwakernaak and Šebek, 1994; Kwakernaak, 2000a).

The design algorithm is mainly described for the continuous-time case. Additional considerations exists for the time-discrete case which are discussed in Section 5.7. Note that in the nominal design problem, continuous and discrete time systems could be handled analogously, but it will be shown that in the stochastic case, small but important differences exists.

5.1 Problem formulation

The system under consideration in this stochastic investigation is similar to what was considered in Chapter 3 with the difference that a stochastic term is added to (3.1), i.e. the following class of models is studied:

$$y = G_u(s)u + G_d(s)d + G_f(s)f + G_n(s)n \tag{5.1}$$

where $y \in \mathbb{R}^m$ is the measurement vector, $u \in \mathbb{R}^{k_u}$ control signals, $d \in \mathbb{R}^{k_d}$ unknown disturbances, $f \in \mathbb{R}^{k_f}$ faults, $n \in \mathbb{R}^{k_n}$ noise, and $G_u(s)$, $G_d(s)$, $G_f(s)$, and $G_n(s)$ are proper transfer matrices of suitable dimensions. The difference between the disturbances d and the noise n is that the disturbances are assumed to have no stochastic description and must be decoupled while the noise is modeled as a white stationary stochastic process with unit covariance. The noise is not decoupled but is handled otherwise.

For deterministic models, residual generators were defined in Definition 3.1. This definition is the basis also for stochastic residual generators and also here it is assumed that perfect decoupling of disturbances d is possible. In case perfect decoupling of d is not possible, new sensors may be needed or signals may have to be transferred from d to n and the model augmented with stochastic descriptions of these signals. Now, the residual generator definition needs an extension where stochastic properties of the residual generator are specified.

For linear models with no unknown inputs, the innovation process associated with the Kalman filter is often used as a residual because of its zero-mean and whiteness properties in the fault-free case. Once the innovations is generated, the fault detection problem reduces to a whiteness test of the residual. Also, other more elaborate decision algorithms can be used based on more deep utilization of stochastic properties of the residual (Basseville and Nikiforov, 1993). Trying to achieve the same properties but also including unknown disturbances in the system leads to the following extension to Definition 3.1:

Definition 5.1 (Whitening residual generator). *A stable and proper linear filter $Q(s)$ is a residual generator for (5.1) if and only if when $f \equiv 0$ it holds that*

$$r = Q(s)\begin{pmatrix} y \\ u \end{pmatrix}$$

is zero mean and white for all u and d.

Note that here, for the sake of convenience, the scalar assumption on r is dropped. This is mainly to be able to keep the presentation close to Nikoukhah

(1994). Of course, for the residual generator to be useful for fault detection, when $f \neq 0$ the zero mean whiteness property need to be violated.

Finally, a restricted class of residual generators defined by Nikoukhah (1994), where the whiteness property of the residual is achieved without restricting the number of linearly independent residuals. This has the consequence that the whiteness property of the residual is achieved without sacrificing any design freedom or fault detectability. This is a smaller class of residual generators than those defined in Definition 5.1.

Definition 5.2 (Innovation filter). *A finite-dimensional linear time-invariant system $Q(s)$ is called an innovation filter for system (5.1) if it is stable with the least number of outputs such that, in the absence of failure,*

1. *its output*
$$r = Q(s)\begin{pmatrix} y \\ u \end{pmatrix}$$
is zero-mean, white and decoupled from u and d,

2. *if $Q'(s)$ is any finite-dimensional linear time-invariant system such that*
$$r' = Q'(s)\begin{pmatrix} y \\ u \end{pmatrix}$$
is decoupled from u and d, then there exists a linear system $L(s)$ such that $Q'(s) = L(s)Q(s)$.

Note that for an innovation filter, the residual is no longer a scalar but equals the dimension of $\mathcal{N}_L(M(s))$.

Assumption: From now on it is assumed that perfect decoupling of both the noise n and disturbances d is not possible. A brief discussion on the case that arises when the noise is perfectly decoupled is presented in Section 5.8.

This chapter now describes an extension, under the assumption above, of the algorithms presented in Chapter 3 to synthesize whitening residual generators and innovation filters. Characterization of residual generators and innovation filters will be derived and presented in Theorem 5.1 and Theorem 5.2. Let the fault-free system (5.1) be described by a state-space realization on the form

$$\dot{x} = Ax + B_u u + B_d d + B_n n \tag{5.2a}$$
$$y = Cx + D_u u + D_d d + D_n n \tag{5.2b}$$

As shown in Chapter 3, any deterministic residual generator $Q(s)$ can be written as $Q(s) = \varphi(s)N_{M_s}(s)P_x$ where

$$M_s(s) = \begin{bmatrix} C & D_d \\ -(sI_{n_x} - A) & B_d \end{bmatrix} \quad \wedge \quad P_x = \begin{bmatrix} I_{k_m} & -D_u \\ 0_{n_x \times k_m} & -B_u \end{bmatrix} \tag{5.3}$$

where n_x is the number of states i.e. the size of x and $N_{M_s}(s)$ is a minimal polynomial basis for the left null-space of $M_s(s)$. In Chapter 3, Theorem 3.1, it

was also required that the pair $\{A, [B_u \ B_d]\}$ was controllable to get a minimal polynomial basis. This requirement is relaxed here because, due to clarity of presentation, minimality issues is neglected here. The consequence of this relaxation is that it possible that matrix $N_{M_s}(s)P_x$ is not irreducible according to Corollary 3.1. However, according to the proof of Theorem 3.1, it is still guaranteed to be row-reduced regardless if (5.2) is controllable or not. In this chapter, row-reducedness will show to be important while irreducibility is not.

For nominal designs in Chapter 3, all solutions were parameterized by a single, rational row-vector $\varphi(s)$ as in (3.6) and (3.7)

$$\varphi(s) = c^{-1}(s)\phi(s)$$

There, no additional modeling was available to guide the selection of the parameterization matrix. Here, additional constraints on the residual is imposed, i.e. the whiteness requirement. So here, parts of the available design freedom will be used to fulfill the whiteness requirements and other constraints according to Definitions 5.1 and 5.2. Now follows a characterization of the parameterization matrix $\varphi(s)$ to fulfill the added requirements:

Theorem 5.1. *A transfer matrix $Q(s)$ is a whitening residual generator for (5.2) if and only if there exists a $\varphi(s)$ such that*

$$Q(s) = \varphi(s)N_{M_s}(s)P_x$$

is proper, stable and it holds that

$$\forall s. H(s)H^T(-s) = \Psi$$

where $H(s) = \varphi(s)N_{M_s}(s)\begin{pmatrix} D_n \\ B_n \end{pmatrix}$ and Ψ is a constant matrix.

Proof. All disturbance decoupling residual generators can be written as

$$Q(s) = \varphi(s)N_{M_s}(s)P_x$$

Insertion of (5.2) into $r = Q(s)\left(\begin{smallmatrix} y \\ u \end{smallmatrix}\right)$ gives, after some straightforward calculations,

$$r = \varphi(s)N_{M_s}(s)P_x\begin{pmatrix} y \\ u \end{pmatrix} = \varphi(s)N_{M_s}(s)\begin{bmatrix} C & D_d & D_n \\ -(sI - A) & B_d & B_n \end{bmatrix}\begin{pmatrix} x \\ u \\ n \end{pmatrix} =$$

$$= \varphi(s)N_{M_s}(s)\begin{bmatrix} D_n \\ B_n \end{bmatrix} n \quad (5.4)$$

Whiteness of r is equivalent to $\Phi_r(j\omega)$ constant for all ω which, since $\Phi_r(s)$ is rational, is equivalent to $\Phi_r(s)$ is constant for all s. The spectrum $\Phi_r(s)$ can be

written as

$$\Phi_r(s) = \varphi(s)N_{M_s}(s)\begin{bmatrix} D_n \\ B_n \end{bmatrix}\Phi_n(s)\begin{bmatrix} D_n \\ B_n \end{bmatrix}^T N_{M_s}^T(-s)\varphi^T(-s) = \\ = \varphi(s)H(s)H^T(-s)\varphi^T(-s)$$

and the theorem follows immediately. ■

Theorem 5.2. *A transfer matrix $Q(s)$ is an innovation filter for system (5.2) if and only if there exists a matrix $\varphi(s)$ such that*

$$Q(s) = \varphi(s)N_{M_s}(s)P_x$$

is proper, stable and it holds that

$$\forall s.H(s)H^T(-s) = \Psi$$

where $H(s) = \varphi(s)N_{M_s}(s)\begin{pmatrix} D_n \\ B_n \end{pmatrix}$, $\Psi \in \mathbb{R}^{r\times r}$ is a constant full-rank matrix, and $r = \dim\ \mathcal{N}_L(M(s))$.

Proof. Following the same lines of proof as in Theorem 5.1, it is seen that $Q(s)$ satisfies the first requirement in Definition 5.2 if and only if there exists a $\varphi(s)$ such that the conditions given in the theorem are satisfied.

For the second requirement from the definition, consider a disturbance decoupling filter $Q'(s)$. From Chapter 3 we know that $Q'(s)$ can be parameterized by

$$Q'(s) = \varphi'(s)N_{M_s}(s)P_x$$

The second requirement is fulfilled if and only if $Q'(s) = L(s)Q(s)$ have a solution for any $\varphi'(s)$. For this to be true, the rank of $\varphi(s)$ must equal $\dim \mathcal{N}_L(M(s))$. This gives that $Q(s)$ fulfills both requirements if and only if there exists a $\varphi(s)$ with full column-rank that satisfies the conditions in the theorem.

Since the least number of rows of an innovation filter is given by $r = \dim \mathcal{N}_L(M(s))$, $Q(s)$ has the least number of outputs if and only if $\varphi(s)$ is square and therefore Ψ has dimension $r \times r$. Also, Ψ is full-rank under the main assumption that it was not possible to perfectly decouple the noise n. ■

Now, the parameterization matrix $\varphi(s)$ has been characterized for both whitening residual generators and innovation filters. However, the results so far does not indicate a design algorithm how to find $\varphi(s)$. Before such an algorithm can be described, some theory on polynomial J-spectral factorization is needed.

5.2 Spectral factorization theory

The material for this section comes mainly from (Kwakernaak and Šebek, 1994; Šebek, 1990), but also from (Ježek and Kučera, 1985; Callier, 1985). The numerical implementation used in this work is described in (Kwakernaak, 2000a). A corresponding discrete time version of this theory is also available and the time-discrete case is discussed in Section 5.7.

A polynomial matrix $Z(s)$ is said to be *para-hermitian* if $Z^T(-s) = Z(s)$. Para-hermitian is sometimes abbreviated as p.h. From now on, only para-hermitian matrices $Z(s)$ with real coefficients are considered in this work. A factorization

$$Z(s) = P^T(-s)JP(s)$$

is called a J-spectral factorization if J is a signature matrix and $P(s)$ a square matrix with real coefficients such that $\det P(s)$ is Hurwitz. This is equivalent to all zeros of the invariant polynomials lying in the closed left half plane. For short, $P(s)$ is said to be strictly stable. Sometimes the J is omitted and the factorization is called a spectral factorization. The signature matrix J has the following form

$$J = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & -I_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 5.1 show how zeros of a para-hermitian matrix and the zeros of its spectral factor is related. A factorization on the form $Z(s) = P(s)JP^T(-s)$ is

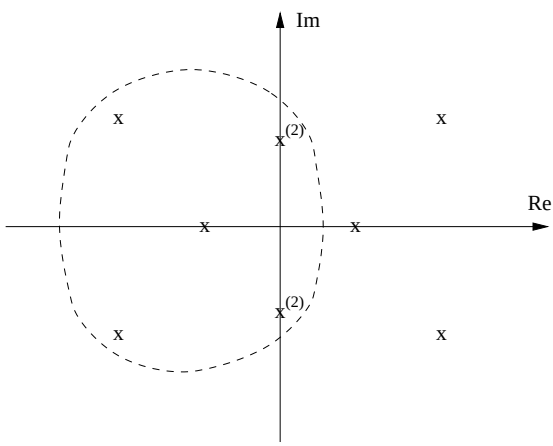


Figure 5.1: Distribution of zeros of a para-hermitian matrix $Z(s)$ and its spectral factor $P(s)$. The X marks the zeros of the invariant polynomials of $Z(s)$ and the dotted line marks the zeros of a spectral factor $P(s)$. Only half of the zeros on the imaginary axis is a zero of $P(s)$.

called a *J-spectral co-factorization*. A specific class of spectral factorizations is of particular importance in this work.

Definition 5.3 (Canonical spectral factorization). *Let $P(s)$ be a spectral factor of the para-hermitian matrix $Z(s)$. The spectral factorization is said to be* canonical *if $P(s)$ is column-reduced and the column-degrees equals the half diagonal degrees of $Z(s)$.*

No necessary and sufficient existence conditions are known for J-spectral factorization (Kwakernaak and Šebek, 1994). However, the following necessary condition due to (Jakubovič, 1970) (referred in (Kwakernaak and Šebek, 1994)) gives a necessary condition.

Theorem 5.3 (Existence of J-Spectral Factorization). *Suppose that the multiplicity of the zeros on the imaginary axis of each of the invariant polynomials of the para-hermitian polynomial matrix $Z(s)$ is even, then $Z(s)$ has a spectral factorization $Z(s) = P^T(-s)JP(s)$.*

A related issue is uniqueness of the factorization

Theorem 5.4 (Non-uniqueness of J-Spectral Factorization). *Let the polynomial Matrix $P(s)$ be a spectral factor of the full-rank para-hermitian matrix $Z(s)$ with corresponding signature matrix J.*

1. *All other spectral factors of $Z(s)$ are of the form $U(s)P(s)$ with $U(s)$ unimodular such that*
$$U^T(-s)JU(s) = J$$
Matrix $U(s)$ is said to be a J-unitary *unimodular matrix.*

2. *If the factorization is canonical, i.e. $P(s)$ is column reduced, any other spectral factor is on the form $UP(s)$ with U constant J-unitary.*

Proof. See (Kwakernaak and Šebek, 1994) ■

Theorem 5.5. *Let $Z(s)$ be positive definite on the imaginary axis, then the J-spectral factorization of $Z(s)$ is canonical.*

Proof. See (Kwakernaak and Šebek, 1994) ■

5.2.1 Note on the singular case

A brief description of spectral (co-)factorization of singular para-hermitian matrices is now given. This presentation follows (Šebek, 1990), and the characterization of spectral factors of singular para-hermitian matrices will be used in Section 5.8 where the design of residual generators where the assumption from Section 5.1 is violated, i.e. it is possible to perfectly decouple the stochastic noise n.

Let $Z(s)$ be a para-hermitian, $n \times n$ matrix of rank $m < n$. First, find the greatest right divisor of $Z(s)$, i.e. find $Z_R(s)$ and a unimodular $U(s)$ such that

$$Z(s) = U(s) \begin{bmatrix} Z_R(s) \\ 0 \end{bmatrix}$$

By symmetrical extraction of $U(s)$ we get

$$Z(s) = U(s) \begin{bmatrix} \bar{Z}(s) & 0 \\ 0 & 0 \end{bmatrix} U^T(-s)$$

where $\bar{Z}(s)$ is a square $m \times m$ non-singular para-hermitian matrix. Let a $\bar{P}(s)$ and $\bar{J}$ be a spectral co-factor and the signature of $\bar{Z}(s)$. Then a spectral co-factorization of $Z(s)$ is given by

$$Z(s) = U(s) \begin{bmatrix} \bar{P}(s) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{J} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{P}^T(-s) & 0 \\ 0 & I \end{bmatrix} U^T(-s)$$

That is, a spectral co-factor and the signature of $Z(s)$ can always be written as

$$P(s) = U(s) \begin{bmatrix} \bar{P}(s) & 0 \\ 0 & I \end{bmatrix} \quad \wedge \quad J = \begin{bmatrix} \bar{J} & 0 \\ 0 & 0 \end{bmatrix}$$

where $\bar{P}(s)$ and $\bar{J}$ is a spectral co-factor and signature of a non-singular para-hermitian matrix.

Also, a spectral factorization of a singular para-hermitian matrix is not necessarily canonical and therefore the row-reducedness and degree property of Theorem 5.5 does not hold.

5.3 Introductory examples

Before going into details, describing a design algorithm and existence conditions, three small illustrative examples are presented that illustrates various issues a design algorithm must consider. The first describes a successful design and the last two illustrates the two cases when whitening residual generators do not exist. All these introductory examples are scalar in the sense that only one, linearly independent, disturbance decoupling residual generator exists and thereby is, in these cases, a whitening residual generator equivalent to an innovation filter.

5.3.1 Example 1: Successful design

Consider a system described by

$$y = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{s(s+1)} \end{bmatrix} u + \begin{pmatrix} 0 \\ 1 \end{pmatrix} d + \begin{pmatrix} 1 \\ 0 \end{pmatrix} f + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \tag{5.5}$$

The only residual generator that decouples d is parameterized by the free variable $\varphi(s)$ as

$$r = \varphi(s)[s+1 \quad 0 \quad -1] \begin{pmatrix} y \\ u \end{pmatrix} \tag{5.6}$$

Inserting (5.5) into (5.6), the internal form, in the fault-free case becomes

$$r = \varphi(s)(s+1)n_1$$

It is clear that by letting $\varphi(s) = \frac{1}{s+1}$ we get a white residual in the fault-free case by the stable and proper residual generator

$$Q(s) = [1 \quad 0 \quad -\frac{1}{s+1}]$$

5.3.2 Example 2: Zeros on the imaginary axis

Consider the same example as above, but switch the positions of f and d, i.e.

$$y = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{s(s+1)} \end{bmatrix} u + \begin{pmatrix} 1 \\ 0 \end{pmatrix} d + \begin{pmatrix} 0 \\ 1 \end{pmatrix} f + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

In the same way as before it is clear that all disturbance decoupling residual generators can be parameterized as

$$r = \varphi(s)[0 \quad s(s+1) \quad -1] \begin{pmatrix} y \\ u \end{pmatrix}$$

for which the fault-free internal form is given by

$$r = \varphi(s)s(s+1)n_2$$

Here it is clear that no strictly stable $\varphi(s)$ exists making r white, all because of the finite zero on the imaginary axis in the transfer function from n to r.

This also shows a link to non strongly-detectable faults (Chen and Patton, 1994; Nyberg, 2000). A zero at $s = 0$ will appear in the transfer function from n to r if n enters the system in the same way as a non strongly detectable fault f which was the case in the example above.

5.3.3 Example 3: Infinite zeros

Consider the scalar system

$$y = \frac{1}{s+1}u + f + \frac{1}{(s+2)^2}n$$

All residual generators can be written

$$r = \varphi(s)[s+1 \quad -1] \begin{pmatrix} y \\ u \end{pmatrix}$$

for which the internal form is

$$r = \varphi(s)\frac{s+1}{(s+2)^2}n \tag{5.7}$$

It is clear that for r to be white $\varphi(s) = \frac{(s+2)^2}{s+1}$ which gives an improper, and thus non-realizable, residual generator

$$r = [(s+2)^2 \quad -\frac{(s+2)^2}{s+1}]\begin{pmatrix} y \\ u \end{pmatrix}$$

And this was caused by the infinite zero of the transfer function $\frac{s+1}{(s+2)^2}$ in (5.7).

Now, with these three examples in mind, a design algorithm is described in the next section.

5.4 Design algorithm

The main step in designing both whitening residual generators and innovation filters is to first compute $N_{M_s}(s)$ in (5.3) and then find $\varphi(s)$ such that $Q(s) = \varphi(s)N_{M_s}(s)P_x$ is stable, proper, and the spectrum of r is constant for all s. For the innovation filter, additional requirements on $\varphi(s)$ is needed.

Now, existence conditions and design procedures for these filters will be derived. First, results for innovation filters is derived in Section 5.4.1 and then, the more involved case of whitening residual generators is addressed in Section 5.4.2. The proofs are constructive, outlining design algorithms that finds all possible whitening residual generators and innovation filters.

First, for sake of notational convenience, let $Z(s) \in \mathbb{R}^{m\times m}[s]$ denote

$$Z(s) = N_{M_s}(s)\begin{bmatrix} D_n \\ B_n \end{bmatrix}\begin{bmatrix} D_n \\ B_n \end{bmatrix}^T N_{M_s}^T(-s) \tag{5.8}$$

for the remaining part of this chapter. Then, from the proof of Theorem 5.1, the spectrum of r can be written

$$\Phi_r(s) = \varphi(s)Z(s)\varphi^T(-s)$$

This also implies that the assumption made in Section 5.1, that it is not possible to perfectly decouple the stochastic noise n, is equivalent to $Z(s)$ being full-rank. If $Z(s)$ would be rank deficient, there would exist a $\varphi(s)$ such that the spectrum of r would be 0, i.e. the noise would be perfectly decoupled. Therefore, in this section it is assumed, unless otherwise noted, that $Z(s)$ is full-rank. Further discussions on the case when $Z(s)$ is not full-rank is found in Section 5.8.

5.4.1 Design of innovation filters

Before the main results can be stated, a lemma characterizing the parameterization matrix $\varphi(s)$ in Theorem 5.1 and 5.2 is needed:

Lemma 5.1. *Assume $Z(s)$ full-rank. Then there exists a $\varphi(s)$ such that the linear time-invariant filter $Q(s) = \varphi(s)N_{M_s}(s)P_x$ produces white residuals if and only if $\varphi(s)$ can be written*

$$\varphi(s) = \eta(s)P^{-1}(s)$$

where $P(s)$ is a spectral co-factor of $Z(s)$ and $\eta(s)\eta^T(-s) = \Psi$ for some constant matrix Ψ.

Proof. According to the proof of Theorem 5.1, the spectrum of r can be written

$$\Phi_r(s) = \varphi(s)Z(s)\varphi^T(-s) \tag{5.9}$$

Note that $Z(s)$ is a para-hermitian polynomial matrix. Now, let $P(s)$ be a spectral co-factor and J a signature of $Z(s)$, i.e.

$$Z(s) = P(s)JP^T(-s) \tag{5.10}$$

Since $Z(s)$ is assumed positive definite it has signature $J = I_m$. Insertion of (5.10) into (5.9) and denoting $\eta(s) = \varphi(s)P(s)$ gives

$$\Phi_r(s) = \varphi(s)P(s)JP^T(-s)\varphi^T(-s) = \eta(s)\eta^T(-s)$$

Thus, $\Phi_r(s)$ is constant for all s if and only if $\eta(s)\eta^T(-s) = \Psi$ for some constant Ψ. The parameterization matrix $\varphi(s)$ is found by solving for $\varphi(s)$ in the equation

$$\eta(s) = \varphi(s)P(s) \tag{5.11}$$

which has only one unique solution $\varphi(s) = \eta(s)P^{-1}(s)$. ■

Remark: Any constant $\eta(s)$ yields a constant $\eta(s)\eta^T(-s)$ for all s. Note however that also non-constant $\eta(s)$ exists. One such example is $\eta(s) = [\frac{1}{s+1} \ \frac{s}{s+1}]$.

Now, we are ready to present the main theorem on design of innovation filters.

Theorem 5.6. *If $Z(s)$ is full rank, an innovation filter exists if and only if*

$$\forall i. \textit{row-deg}_i N_{M_s}(s) \begin{pmatrix} D_n \\ B_n \end{pmatrix} = \textit{row-deg}_i N_{M_s}(s)$$

and $Z(s)$ has no roots on the imaginary axis. Furthermore, if an innovation filter exist, all innovation filters can be parameterized as

$$Q(s) = \eta(s)P^{-1}(s)N_{M_s}(s)P_x$$

where $P(s)$ is a spectral co-factor of $Z(s)$ and $\eta(s)$ is any strictly stable, full-rank matrix, such that $\eta(s)\eta^T(-s)$ is constant.

Proof. According to Theorem 5.2 and Lemma 5.1, an innovation filter exists if and only if there exists an $\eta(s)$ such that

$$Q(s) = \eta(s)P^{-1}(s)N_{M_s}(s)P_x$$

is stable, proper, $\eta(s)\eta^T(-s)$ is constant and full-rank of dimension $r \times r$ with $r = \dim\ \mathcal{N}_L(M_s(s))$.

First it will be shown, by contradiction, that $Z(s)$ have no roots on the imaginary axis is a necessary condition for the existence of an innovation filter. For this, assume $Q(s)$ is an innovation filter and that $Z(s)$ has a zero at $s_0 = j\omega_0$. Since $Q(s)$ is strictly stable, $\lim_{s\to j\omega_0} Q(s)$ exists. But, $Z(s)$ has a zero at s_0 implies that $P(s_0)$ is rank deficient. Since, according to assumption, $Q(s_0)$ exists, it must hold that $\eta(s)$ looses rank at s_0 since $N_{M_s}(s)P_x$ is irreducible. However, this contradicts $\Psi = \eta(s)\eta^T(-s)$ being full-rank which gives that full-rank of $Z(s)$ on the imaginary axis is a necessary condition for $Q(s)$ to be stable.

Next it will be shown, also by contradiction, that the row-degree condition in the theorem is also a necessary condition for the existence of an innovation filter. Assume $Q(s)$ is an innovation filter and that there exists an i such that

$$\text{row-deg}_i\ N_{M_s}(s)\begin{pmatrix} D_n \\ B_n \end{pmatrix} < \text{row-deg}_i\ N_{M_s}(s) \tag{5.12}$$

Partition $N_{M_s}(s) = [V_1(s)\ V_2(s)]$ according to the block-structure of (5.3). Then, from the proof of Theorem 3.1, we know that $V_1(s)$ is row-reduced, $\text{row-deg}_i N_{M_s}(s) = \text{row-deg}_i V_1(s)$, and that $V_2(s) = V_1(s)C(sI - A)^{-1}$. Since $V_1(s)$ is row-reduced, we can rewrite (5.12) as

$$\text{row-deg}_i\ S_{V_1}(s)V_{1,hr}D_n + \tilde{V}_1(s)D_n + V_2(s)B_n < \text{row-deg}_i\ V_1(s)$$

where the decomposition of $V_1(s)$ is done according to Definition 3.A.10. Since the row-degrees of $\tilde{V}_1(s)$ and $V_2(s)$ is strictly less than the row-degrees of $S_{V_1}(s)$, the inequality can only be fulfilled if $V_{1,hr}D_n$ does not have full row-rank. This also gives that

$$\lim_{s\to\infty} V_1(s)D_n = \lim_{s\to\infty} S_{V_1}(s)V_{1,hr}D_n \tag{5.13}$$

does not have full row-rank. Now, since $Q(s)$ is an innovation filter, there exist an $\eta(s)$ such that

$$Q(s) = \eta(s)P^{-1}(s)N_{M_s}(s)P_x$$

and $H(s)H^T(-s)$ is square, full-rank, and constant where

$$H(s) = \eta(s)P^{-1}(s)N_{M_s}(s)\begin{bmatrix} D_n \\ B_n \end{bmatrix}$$

But, when s goes to infinity, it holds that

$$\lim_{s\to\infty} H(s) = \lim_{s\to\infty} \eta(s)P^{-1}(s)V_1(s)\left(C(sI - A)^{-1}B_n + D_n\right) = \\ = \lim_{s\to\infty} \eta(s)P^{-1}(s)V_1(s)D_n$$

which does not have full row-rank due to (5.13) and the fact that $\eta(s)$ and $P(s)$ is square and full-rank. Thus, $\lim_{s\to\infty} H(s)$ does not have full rank which contradicts that $H(s)H^T(-s)$ is constant and full-rank.

Now, sufficiency. Since $Z(s)$ does not have zeros on the imaginary axis, a spectral co-factor $P(s)$ will be strictly stable and, according to Theorem 5.5 and Definition 5.3, row-reduced with row-degrees satisfying

$$\text{row-deg}_i\ P(s) = \text{row-deg}_i\ N_{M_s}(s)\begin{bmatrix} D_n \\ B_n \end{bmatrix}$$

Thus, Theorem 3.B.6 gives that

$$Q(s) = P^{-1}(s)N_{M_s}(s)P_x$$

will be proper, strictly stable and fulfill all requirements in Definition 5.2, i.e. $Q(s)$ is an innovation filter.

Finally, if $Q(s)$ is an innovation filter, it is immediate that $Q'(s)$ is an innovation filter if and only if

$$Q'(s) = \eta(s)Q(s)$$

where $\eta(s)$ is a square, full-rank, all-pass link i.e. $\eta(s)\eta^T(-s)$ is constant and full rank. ■

Summary of design procedure

1. Form $M_s(s)$ according to (5.3) and compute $N_{M_s}(s)$.
2. Form $Z(s)$ as in (5.8). If $Z(s)$ is full-rank, an innovation filter exists if and only if $P(s)$ is strictly stable and

$$\forall i.\text{row-deg}_i N_{M_s}(s)\begin{pmatrix} D_n \\ B_n \end{pmatrix} = \text{row-deg}_i N_{M_s}(s)$$

3. All innovation filters, if any exists, are then given by:

$$Q(s) = \eta(s)P^{-1}(s)N_{M_s}P_x$$

where $P(s)$ is a spectral co-factor of $Z(s)$ and $\eta(s)$ is any invertible matrix such that $\eta(s)\eta^T(-s)$ is constant for all s.

5.4.2 Design of whitening residual generators

The design procedure for whitening residual generators is a bit more complex due to the increased design freedom, resulting in a possibly more involved design procedure. It will be shown that the design procedure is not necessarily more complex, only in special cases when $Z(s)$ has zeros on the imaginary axis and/or if no row of $P^{-1}(s)N_{M_s}(s)P_x$ is proper.

A whitening residual generator is, according to Definition 5.1, a stable and proper filter that produces white residuals in the fault-free case. The whiteness property has already been characterized in Lemma 5.1. Now, more details on the properness and stability properties of the residual generator is needed before the main result can be stated. We begin with considering the properness condition.

First, a result is needed characterizing all rational $\eta(s)$ such that $\eta(s)\eta^T(-s)$ is constant which is fundamental for the whiteness property but also influences properness/stability results later on.

Lemma 5.2. *Let $\eta(s)$ be a $1 \times m$ row-vector of transfer functions such that $\eta(s)\eta^T(-s) = \Psi$ with $\Psi \neq 0$. Then it holds that $\eta(s)$ is proper and at least one element of $\eta(s)$ is not strictly proper.*

Proof. Since $\eta(s)\eta^T(-s) = \Psi$, it holds that

$$\Psi = \eta(j\omega)\eta^T(-j\omega) = \sum_{k=1}^{m} |\eta_k(j\omega)|^2$$

Assume that element k of $\eta(s)$ is improper, then the limit $\lim_{\omega\to\infty} |\eta_k(j\omega)|^2$ would not exist. Since for all l, $|\eta_l(j\omega)|^2 \geq 0$, the improper assumption has lead to a contradiction, i.e. $\eta(s)$ is proper.

Left to prove is that at least one element of $\eta(s)$ is not strictly proper. Denote the constant coefficient matrix of $\eta(s)$ with H_0, i.e. $H_0 = \lim_{s\to\infty} \eta(s)$. The limit exists since $\eta(s)$ is proper. Then,

$$\eta(s) = H_0 + \tilde{\eta}(s)$$

where it holds that $\tilde{\eta}(s)$ is strictly proper. Now, assume $H_0 = 0$, i.e.

$$\Psi = \eta(j\omega)\eta^T(-j\omega) = \tilde{\eta}(j\omega)\tilde{\eta}^T(-j\omega)$$

Since all elements of $\tilde{\eta}(s)$ is strictly proper, $\lim_{\omega\to\infty} |\eta_k(j\omega)|^2 = 0$ which is a contradiction, i.e. $H_0 \neq 0$ which means that at least one element of $\eta(s)$ is not strictly proper. ■

Now we can almost characterize proper, whitening residual generators. But first some additional notation is needed. From now on, let matrix $W(s)$ denote

$$W(s) = \text{quotient } [P^{-1}(s)N_{M_s}(s)] \tag{5.14}$$

See Theorem 3.B.7 for definitions on matrix quotient and remainder. Let $W(s) \in \mathbb{R}^{m\times n}[s]$ have degree d, then a matrix that will prove useful is the Sylvester matrix of $W(s)$ which is defined as

$$\text{sylv}(W(s), q) \triangleq \begin{bmatrix} W_0 & W_1 & \cdots & W_d & 0 & 0 & 0 \\ 0 & W_0 & W_1 & \cdots & W_d & 0 & 0 \\ \vdots & \ddots & \ddots & & & \ddots & \vdots \\ 0 & \cdots & 0 & W_0 & W_1 & \cdots & W_d \end{bmatrix}$$

with $q+1$ block-rows. Finally, for $0 \leq q \leq d$, introduce the notation

$$\mathcal{W}_q \triangleq \begin{bmatrix} W_q & W_{q+1} & \dots & W_d & 0 & 0 \\ \vdots & & \ddots & & W_d & 0 \\ W_1 & W_2 & \dots\dots\dots & & W_{d-1} & W_d \end{bmatrix} \tag{5.15}$$

Now, we are ready to state the properness result in the following lemma.

Lemma 5.3. *Let $W(s)\mathbb{R}^{m\times n}[s]$, defined in (5.14), have degree d. Then there exists a rational $\eta(s)$ such that*

$$Q(s) = \eta(s)P^{-1}(s)N_{M_s}(s)P_x \text{ is proper} \quad \wedge \quad \eta(s)\eta^T(-s) \text{ is constant}$$

if and only if $\mathcal{N}_L(\mathcal{W}_d) \neq \emptyset$. Then $\eta(s) = d^{-1}(s)n(s)$ where $n(s)$ is parameterized by the scalar polynomial $\gamma_1(s)$ and the polynomial row-vector $\gamma_0(s)$ such that

$$n(s) = \gamma_1(s)[l_0 \ \cdots \ l_q] \begin{bmatrix} I \\ \vdots \\ s^q I \end{bmatrix} + \gamma_0(s)$$

where row-deg $\gamma_0(s) <$ deg $\gamma_1(s) = r$ and $[l_0 \ \cdots \ l_q] \in \mathcal{N}_L(\mathcal{W}_q)$ with $q \leq d$. A denominator $d(s)$ always exists that ensures $\eta(s)\eta^T(-s)$ constant.

Proof. Let $H(s) = P^{-1}(s)N_{M_s}(s)P_x$ and let $W(s)$ and $E(s)$ be the polynomial matrix quotient and remainder of $H(s)$ respectively, i.e.

$$Q(s) = \eta(s)H(s) = \eta(s)W(s) + \eta(s)P^{-1}(s)E(s)$$

First note that $\eta(s)$ is proper according to Lemma 5.2. Also, since $P^{-1}(s)E(s)$ is strictly proper it holds that $Q(s)$ is proper if and only if $\eta(s)W(s)$ is proper. Now, write $\eta(s)$ as

$$\eta(s) = d^{-1}(s)n(s)$$

where $d(s)$ is a scalar polynomial and $n(s)$ is a polynomial row-vector of the same dimensions as $\eta(s)$. Since $\eta(s)$ is proper, but not strictly proper it must hold that

$$\text{deg } d(s) = \text{row-deg } n(s) \tag{5.16}$$

Now, for $\eta(s)W(s)$ to be proper it must hold that deg $d(s) \geq$ row-deg $n(s)W(s)$. This together with (5.16) gives that $Q(s)$ is proper if and only if there exists an $n(s)$ such that

$$\text{row-deg } n(s) \geq \text{row-deg } n(s)W(s) \tag{5.17}$$

Let $W(s) \in \mathbb{R}^{m\times n}[s]$ be a matrix of degree d and let $n(s) = \sum_{i=0}^{q} n_i s^i$ with $q \leq d$ and $n_q \neq 0$. Assume that no $n(s)$ of degree less than q exists such that

(5.17) is satisfied. Then,

$$n(s)W(s) = [n_0 \ \cdots \ n_q]\text{sylv}(W(s), q) \begin{bmatrix} I \\ sI \\ \vdots \\ s^{q+d}I \end{bmatrix} = [n_0 \ \cdots \ n_q][\ \star \mid \mathcal{W}_q] \begin{bmatrix} I \\ \vdots \\ s^qI \\ \cdots\cdots \\ s^{q+1}I \\ \vdots \\ s^{q+d}I \end{bmatrix}$$

where $\star$ denotes a matrix that is of no importance here. From this, (5.17) is satisfied if and only if

$$[n_0 \ \cdots \ n_q] \in \mathcal{N}_L(\mathcal{W}_q)$$

Also, for $q > d$, it is straightforward to see that (5.17) is satisfied if and only if

$$[n_0 \ \cdots \ n_q] \in \mathcal{N}_L(\begin{bmatrix} 0 \\ \mathcal{W}_d \end{bmatrix})$$

Since, it was assumed that no $n(s)$ of degree less than q satisfied the lemma, $\mathcal{W}_d$ has full row-rank and therefore $n_i = 0, i = q-d, \ldots, q$, i.e. $n(s)$ has row-degree less than q. Thus, there exists a $n(s)$ such that $Q(s)$ is proper if and only if for any $q = 1, \ldots, d$

$$\mathcal{N}_L(\mathcal{W}_q) \neq \emptyset$$

And due to the triangular structure of $\mathcal{W}_q$ this is equivalent to $\mathcal{N}_L(\mathcal{W}_d) \neq \emptyset$. Also, if $n(s)$ satisfies (5.17), then so does $\gamma_1(s)n(s) + \gamma_0(s)$ when $\gamma_i(s)$ satisfies the conditions in the theorem.

To form (a stable) $\eta(s)$ that satisfies the lemma, a $d(s)$ always exists such that $\eta(s)\eta^T(-s)$ is constant. One such $d(s)$ is e.g. found by a spectral factorization of $n(s)n^T(-s) = d(s)d^T(-s)$. ■

In the case where $W(s)$ is row-reduced, which is the normal case since $W(s)$ is the quotient between two row-reduced matrices, the above result can be stated little simpler

Lemma 5.4. *Let $W(s) \in \mathbb{R}^{m\times n}[s]$ be row-reduced, then there exists an $n(s) \in \mathbb{R}^{1\times m}[s]$ such that*

$$\textit{row-deg } n(s) \geq \textit{row-deg } n(s)W(s)$$

if and only if $\mathcal{N}_L(\mathcal{W}_0)$ is non-empty.

Proof. Theorem 3.B.5 gives that

$$\text{row-deg } n(s)W(s) = \max_{i.n_i(s)\not\equiv 0} [\mu_i + \deg n_i(s)]$$

where μ_i is the row-degrees of $W(s)$. The condition in the theorem can thus be rewritten

$$\max_{i.n_i(s)\not\equiv 0}[\deg n_i(s)] \geq \max_{i.n_i(s)\not\equiv 0}[\mu_i + \deg n_i(s)] \tag{5.18}$$

which can be satisfied if and only if at least one $\mu_i = 0$.

Let W_i be the coefficient matrices of matrix $W(s)$, i.e.

$$W(s) = \sum_{i=0}^{d} W_i s^i$$

Then, since $W(s)$ is row-reduced matrix $[W_0 \cdots W_d]$ has full-row rank and $[W_1 \cdots W_d]$ looses rank if and only if any row-degree of $W(s)$ is zero. Thus

$$\exists i.\mu_i = 0 \Leftrightarrow \mathcal{N}_L(\mathcal{W}_0) \neq \emptyset$$

which together with (5.18) end the proof. ■

Now that properness of $Q(s)$ is analyzed, we proceed to the stability property, i.e. when does a rational $\eta(s)$ exist such that

$$Q(s) = \eta(s)P^{-1}(s)N_{M_s}(s)P_x \text{ is stable} \quad \wedge \quad \eta(s)\eta^T(-s) \text{ is constant}$$

The word stability is used slightly sloppy here, meaning no poles in the closed right half plane. Therefore, a non-proper $Q(s)$ can here be said to be stable although it is clearly not BIBO-stable. However, since properness is also required later on, having noted this (mis-)use of the word here, stability is used in this sense from now on.

The cause of non strictly-stable filters was, as was demonstrated in the small scalar example in Section 5.3.2, the existence of zeros on the imaginary axis of the transfer function from noise to residual. To be able to proceed with the stability analysis, we need some additional notation. Assume $Z(s)$ has purely imaginary zeros $j\omega_i$, $i = 1, \ldots, r$ with multiplicities $2m_i$, $i = 1, \ldots, r$. Then, denote a polynomial

$$p_{\text{imag}}(s) = \prod_{i=1}^{r}(s - j\omega_i)^{m_i} \tag{5.19}$$

These are the poles we need to cancel/nullify to be able to form a strictly stable residual generator. Also denote

$$p_{\text{stab}}(s) = \prod_{i=1}^{q}(s - \zeta_i)^{n_i} \tag{5.20}$$

where ζ_i, $i = 1, \ldots, q$ are the strictly stable zeros of $Z(s)$ with multiplicities n_i. This implies that for a spectral co-factor $P(s)$ of $Z(s)$ it holds that $\det(P(s)) = kp_{\text{stab}}(s)p_{\text{imag}}(s)$ where k is a real constant. Also, the rational $\eta(s)$ can always be rewritten as

$$\eta(s) = d^{-1}(s)n(s)$$

where $d(s)$ and $n(s)$ are relatively prime. Now, with this notation, $Q(s)$ can be rewritten as

$$Q(s) = \frac{1}{kp_{\text{stab}}(s)p_{\text{imag}}(s)} d^{-1}(s)n(s)\ \text{adj}\, P(s)N_{M_s}(s)P_x$$

and it is immediate that $Q(s)$ is stable if and only if $p_{\text{imag}}(s)$ divides $n(s)\text{adj}\, P(s)N_{M_s}(s)P_x$ and $d(s)$ is strictly stable. A way to compute $n(s)$, if any exists, is to compute the row-Hermite form $T(s)$ of $\text{adj}\, P(s)N_{M_s}(s)P_x$, i.e. find a unimodular matrix $U(s)$ such that

$$T(s) = U(s)\text{adj}\, P(s)N_{M_s}(s)P_x$$

Then, $Q(s)$ can be written

$$Q(s) = \frac{1}{kp_{\text{stab}}(s)p_{\text{imag}}(s)} d^{-1}(s)n(s)\ U^{-1}(s)T(s)$$

Let

$$n(s) = [\zeta_1(s)\ \cdots\ \zeta_m(s)]U(s) \tag{5.21}$$

Due to the upper triangular structure and properties of the Hermite form, $p_{\text{imag}}(s)$ divides $n(s)\ \text{adj}\, P(s)N_{M_s}(s)P_x$, if and only if $p_{\text{imag}}(s)$ divides $\zeta_i(s)T_i(s)$, $i = 1, \ldots, m$ where $T_i(s)$ is the i:th row of $T(s)$. This gives a direct method of selecting the $\zeta_i(s)$ polynomials, and thereby $n(s)$:

$$\zeta_i(s) = c_i(s)p_{\text{imag}}(s)/\text{gld}\,[p_{\text{imag}}(s)\ \ T_i(s)] \tag{5.22}$$

where $c_i(s)$ is an arbitrary polynomial (possibly 0) and gld is the greatest left divisor. Finally, a strictly stable denominator $d(s)$ of $\eta(s)$ exists if and only if the polynomials $c_i(s)$ is chosen such that $n(s)$ has no zeros on the imaginary axis. All steps in the above discussion goes both ways, i.e. a stable solution exists if and only if a rational $\eta(s)$ can be found by rule (5.22), and for referential convenience this is also summarized in a lemma.

Lemma 5.5. *There exist a rational $\eta(s)$ such that*

$$Q(s) = \eta(s)P^{-1}(s)N_{M_s}(s)P_x \text{ is stable} \quad \wedge \quad \eta(s)\eta^T(-s) \text{ is constant}$$

if and only if $\eta(s) = d^{-1}(s)n(s)$ where $n(s)$ is given by (5.21) and (5.22) and $d(s)$ is strictly stable.

Now, a simple and sufficient existence condition that is easy to evaluate is immediate which does not require the computation of Hermite forms.

Corollary 5.1. *If $Z(s)$ is full rank with no zeros on the imaginary axis, a whitening residual generator exists if*

$$\exists i.\text{row-deg}_i N_{M_s}(s) \begin{pmatrix} D_n \\ B_n \end{pmatrix} = \text{row-deg}_i N_{M_s}(s)$$

Proof. Since, according to assumption in the theorem, $Z(s)$ has no zeros on the imaginary axis, strict stability of the residual generator is assured. By Theorem 5.5 and Definition 5.3, $P(s)$ is row-reduced and the row-degrees of $P(s)$ equals the row-degrees of $N_{M_s}(s)\begin{pmatrix} D_n \\ B_n \end{pmatrix}$. Thus, at least one row of

$$P^{-1}(s)N_{M_s}(s)P_x$$

is also proper according to Theorem 3.B.6, and therefore fulfills the requirements of a whitening residual generator. ■

Summary of design procedure

Even though the proofs are constructive, the design steps might be a little concealed by the details in the proofs. Therefore is the design procedure summarized here.

1. Form $M_s(s)$ according to (5.3) and compute $N_{M_s}(s)$.
2. Form $Z(s)$ as in (5.8) and compute a spectral co-factor $P(s)$. If $Z(s)$ is not full-rank, it is possible to perfectly decouple n in the residual and the initial assumption made is violated. If $Z(s)$ has purely imaginary zeros, form the polynomials $p_{\text{imag}}(s)$ and $p_{\text{stab}}(s)$ according to (5.19) and (5.20).
3. Now, four cases can occur, regarding zeros on the imaginary axis of $Z(s)$ and properness of $H(s) = P^{-1}(s)N_{M_s}(s)P_x$.

   i) *No zeros on the imaginary axis and $H(s)$ proper.*
   This is the easiest case where any $Q(s) = \eta(s)H(s)$ with $\eta(s)\eta^T(-s)$ constant is a whitening residual generator.

   ii) *No zeros on the imaginary axis but $H(s)$ not proper.*
   Here, caution has to be taken to achieve a proper residual generator. First, compute the quotient $W(s)$ and matrix $\mathcal{W}_d$ according to (5.14) and (5.15). If $\mathcal{N}_L(\mathcal{W}_d) \neq \emptyset$, any $\eta(s)$ formed as in Lemma 5.3 gives a whitening residual generator $Q(s) = \eta(s)H(s)$. If $\mathcal{N}_L(\mathcal{W}_d) = \emptyset$, then no whitening residual generator exists.

   iii) *Zeros on the imaginary axis and $H(s)$ proper.*
   Here on the other hand, properness is guaranteed but stability has to be ensured. Compute the row-Hermite form of $\operatorname{adj} P(s)N_{M_s}(s)P_x$, i.e. find a unimodular matrix $U(s)$ such that the Hermite form $T(s)$ can be written

   $$T(s) = U(s)\operatorname{adj} P(s)N_{M_s}(s)P_x$$

   Then a rational $\eta(s)$ giving a stable residual generator exists if and only if it can be written as (5.21) using rule (5.22). The denominator $d(s)$ of $\eta(s)$ is most easily found by spectral factorization of

$n(s)n^T(-s)$. A suitable procedure to find a state-space realization is to first compute $n(s)$, then perform the polynomial matrix division

$$\frac{1}{p_{\text{imag}}(s)}\text{adj}\, P(s)N_{M_s}(s)P_x = V(s) + \frac{1}{p_{\text{imag}}(s)}R(s)$$

and then realize

$$Q(s) = \frac{1}{kd(s)p_{\text{stab}}(s)}V(s)$$

which will be strictly stable. The remainder $R(s)$ should be 0 (or close to zero due to finite precision arithmetics).

iv) *Zeros on the imaginary axis but $H(s)$ not proper.*
This case is not well covered. Even though all stable and all proper solutions is characterized, it is not obvious how to combine the results. A feasible heuristic is to solve e.g. the stability constraint in a first step, and then try to find a proper solution among the stable solutions.

5.5 Design examples

This section includes 4 design examples that illustrates different aspects of the design problem and the proposed design algorithm. The first three examples are based around the same linearized airplane model that was used in Section 3.6 to demonstrate the deterministic design problem.

In the first example, a complete design of an innovation filter and a whitening residual generator is shown. In the second example, only the noise environment is changed, leaving the rest of the model the same. For this second model setup it is shown that no innovation filter or whitening residual generator exists. In yet a third example, using a third noise setup, it is shown that an innovation filter does not exists but a whitening residual generator exists that has acceptable fault sensitivity. The case with purely imaginary zeros is demonstrated in a final fourth example.

All calculations are done in Matlab using Polynomial Toolbox 2.5 for Matlab 5 (2001). All functions used are included in the toolbox[1]. Included in Appendix 5.A is a full Matlab implementations of innovation filter design.

5.5.1 Design example: Aircraft dynamics

The model from Section 3.6 is here extended with noise models. The same set of faults is considered, i.e. additive actuator and sensor faults. Therefore, the

[1]The spectral factorization procedure used is (Kwakernaak, 2000a) which probably will be included in future versions of the Toolbox. The spectral factorization command shipped with the toolbox can not handle zeros on the imaginary axis.

total model including fault models and noise descriptions becomes:

$$\dot{x} = Ax + B_u u + B_f f + B_n n$$
$$y = Cx + D_u u + D_f f + D_n n$$

Details on fault models and numerical values for the state-space matrices is provided in Section 3.6. The noise is assumed white with unit covariance. The three different examples is based on this simulation model, each case with different noise assumptions, i.e. different B_n and D_n matrices.

The design goal in all the three examples based on this model is a residual generator $Q(s)$ that decouples faults in the elevator angle actuator, and produces a white residual in the fault-free case.

Process and measurement noise

In this first example, both measurement noise and process noise is considered and state-space matrices B_n and D_n is set to

$$B_n = [I_5 \quad 0_{5\times 3}] \qquad D_n = [0_{3\times 5} \quad I_3]$$

First, an innovation filter design is performed. Calculations in MATLAB give

$$N_M(s) = \begin{bmatrix} 0.0496s & 0.703s + 0.0378 & \dots & & \\ 0.421s^2 + 0.27s & -0.123 & \dots & & \\ \dots & 0.0643 & 0.0844 & -0.703 & 0 \\ \dots & 0.0185s^2 - 0.0174s - 0.306 & 0.582 & 0 & 0 \end{bmatrix} \tag{5.23}$$

Thus, the dimension of the null-space $\mathcal{N}_L(M(s))$ is 2, i.e. there exists exactly two linearly independent numerators that decouples f_6.

Step 2 from the summary in Section 5.4.1 was to compute matrix $Z(s)$ and checking full-rank condition. Matrix $Z(s)$ is shown to be

$$Z(s) = \begin{bmatrix} -0.5s^2 + 0.5 & 0.021s^3 - 0.012s^2 - 0.11s - 0.011 \\ -0.021s^3 - 0.012s^2 + 0.11s - 0.011 & 0.18s^4 - 0.26s^2 + 0.57 \end{bmatrix}$$

which has full rank. Performing a J-spectral co-factorization gives:

$$P(s) = \begin{bmatrix} -0.59s - 0.61 & -0.39s - 0.37 \\ -0.26s^2 - 0.57s - 0.38 & 0.33s^2 + 0.75s + 0.66 \end{bmatrix} \qquad J = I_2$$

The spectral factor $P(s)$ is strictly stable which can be seen by computing the zeros of the invariant polynomials. Computing the zeros of $P(s)$ in Matlab gives $s = -1.0196$ and $s = -1.1124 \pm j0.7305$.

Checking for existence of innovation filter according to Theorem 5.6 gives

$$\text{row-deg } N_{M_s}(s) = \{1, 2\}$$
$$\text{row-deg } N_{M_s}(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix} = \{1, 2\}$$

i.e. an innovation filter exists and can be formed as $Q(s) = P^{-1}(s)N_{M_s}(s)P_x$. Thus, the parameterization matrix $\eta(s)$ in Step 3 of the design summary in Section 5.4.1 is chosen to be the identity matrix.

Next, a scalar whitening residual generator is to be designed. The first 2 steps in the design summary in Section 5.4.2 has already been performed, left is to find an $\eta(s)$ in step 3 achieving unit variance in the fault-free residual. In this case it is easy since case i from step 3 applies and one choice of $\eta(s)$ that satisfies the design requirements is

$$\eta(s) = \frac{1}{\sqrt{2}}[1 \quad 1]$$

and the whitening residual generator can be formed as

$$Q(s) = \eta(s)P^{-1}(s)N_{M_s}(s)P_x$$

which is a 3:rd order realizable and strictly stable residual generator. The order of the residual generator is, due to the choice of $\eta(s)$, equal to the sum of row-degrees of $P(s)$. Figure 5.2 shows how the faults influence the residual and Figure 5.3 shows the fault-free spectrum $\Phi_r(j\omega)$ which is 1 for all ω as expected. Especially note that the desired decoupling of fault f_6 has succeeded while keeping the spectrum of r constant for all ω.

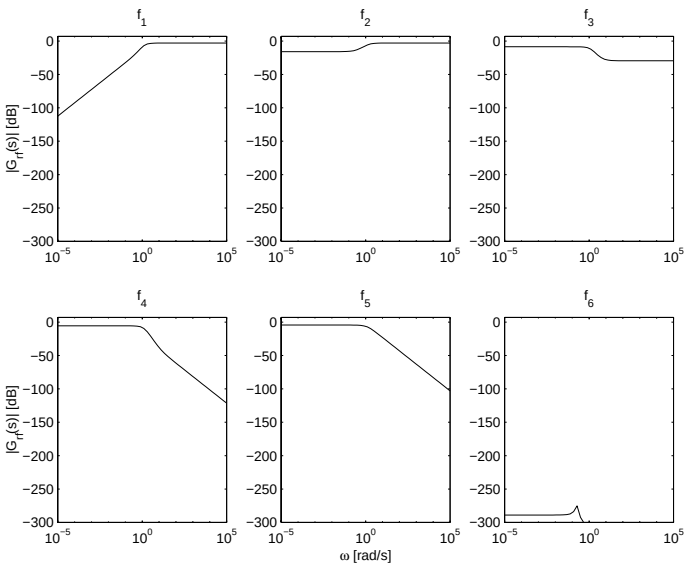


Figure 5.2: Magnitude bode plots for the gain from faults to the residual.

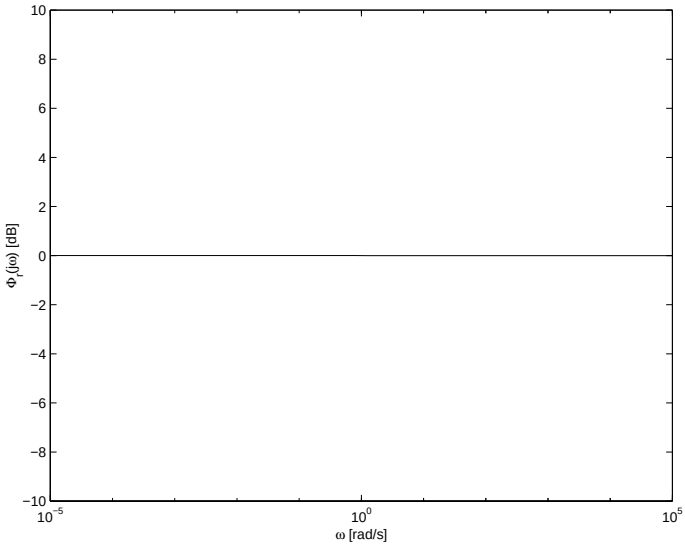


Figure 5.3: Spectrum $\Phi_r(j\omega)$.

Only process noise

In this second example, only process noise is considered and state-space matrices B_n and D_n is set to:

$$B_n = I_5 \qquad D_n = 0_{3\times 5}$$

The null-space basis $N_{M_s}(s)$ is identical to the first example (5.23). The row-degrees of $N_{M_s}(s)$ is $\{1, 2\}$ and the row-degrees of $N_{M_s}(s)\begin{bmatrix} D_n \\ B_n \end{bmatrix}$ is $\{0, 1\}$, i.e. no innovation filter exists according to Theorem 5.6. Next it will be shown that no whitening residual generator exists either. Performing the spectral factorization gives a spectral co-factor:

$$P(s) = \begin{bmatrix} -1 & 0 \\ 0 & 0.9975s + 1.611 \end{bmatrix}$$

Since the roots of $Z(s)$ is ± 1.6152, stability is ensured and only properness is left. The matrix quotient $W(s)$ of $P^{-1}(s)N_{M_s}(s)P_x$ from (5.14) becomes

$$W(s) = \left[\begin{array}{cccc} -0.0703s & -0.998s - 0.0537 & -0.0912 & \dots \\ 0.997s - 0.971 & -0.0703s + 0.0646 & 0.044s - 0.119 & \end{array}\right.$$
$$\left.\begin{array}{cccc} \dots & -0.12 & 0.998 & 0 \\ \dots & -0.00843 & 0.0703 & 0 \end{array}\right]$$

and it is easily checked that $W(s)$ is row-reduced. Also, computing $\mathcal{W}_0$ according to (5.15) gives

$$\mathcal{W}_0 = \begin{bmatrix} -0.0703 & -0.998 & 0 & 0 & 0 & 0 \\ 0.997 & -0.0703 & 0.044 & 0 & 0 & 0 \end{bmatrix}$$

which has full row-rank. Then, according to Lemma 5.4, no proper whitening residual generator exists.

Noise on all states and sensor 3

In this case the process is subjected to noise on all states and on sensor 3, i.e. the matrices B_n and D_n are given by

$$B_n = [I_5 \ 0] \qquad D_n = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, computing $N_{M_s}(s)$ and $Z(s)$ as before gives that $Z(s)$ is strictly stable and that the row-degrees of $N_{M_s}(s)$ is (as before) $\{1, 2\}$ and row-degrees of $N_{M_s}(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix}$ is $\{0, 2\}$. This gives that no innovation filter exists according to Theorem 5.6. But because $Z(s)$ has no imaginary zeros and the row-degree of the second row of $N_{M_s}(s)$ does not decrease when multiplied by the noise distribution matrices, Corollary 5.1 proves existence of a whitening residual generator. Computing $W(s)$ gives that

$$W(s) = \begin{bmatrix} 0.14s + 0.4 & 2s + 0.12 & 0.2 & 0.24 & -2 & 0 \\ 0.014s - 46 & 0.2s - 1.5 & -2 & 0.024 & -0.2 & 0 \end{bmatrix}$$

Matrix $W(s)$ has degree 1, and the existence of a whitening residual generator is proven by computing $\mathcal{W}_1$ for which a, non-empty, left null-space basis is computed. Using Lemma 5.3, a (here constant) $\eta(s)$ is found

$$\eta(s) = \begin{bmatrix} -0.0994 & 0.995 \end{bmatrix}$$

Then, a whitening residual generator can be formed by

$$Q(s) = \eta P^{-1}(s) N_{M_s}(s) P_x$$

Figures 5.4 and 5.5 shows the fault influence on the residual and the spectrum of the fault-free residual. Here it is clear that the residual still is able to detect all faults, besides from fault f_6 which should be decoupled according to design specifications. Thus, even though an innovation filter didn't exist, a whitening residual generator with satisfactory fault detectability properties existed. This residual generator could not have been designed algorithm proposed in (Nikoukhah, 1994).

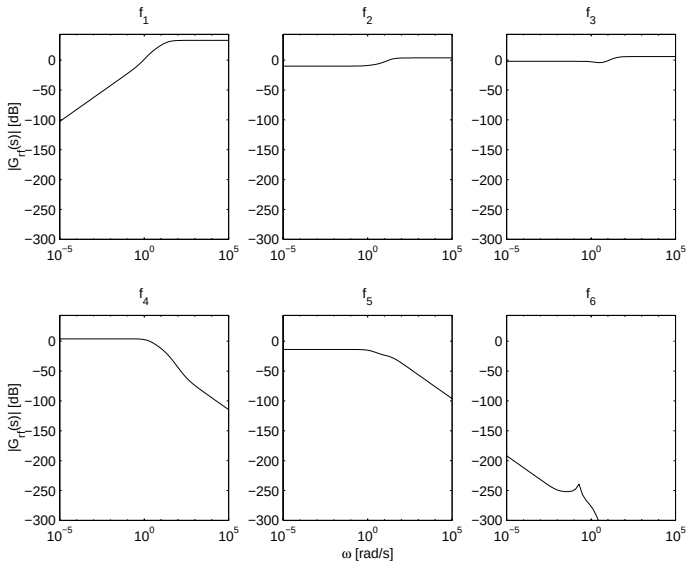


Figure 5.4: Magnitude bode plots for the fain from faults to the residual.

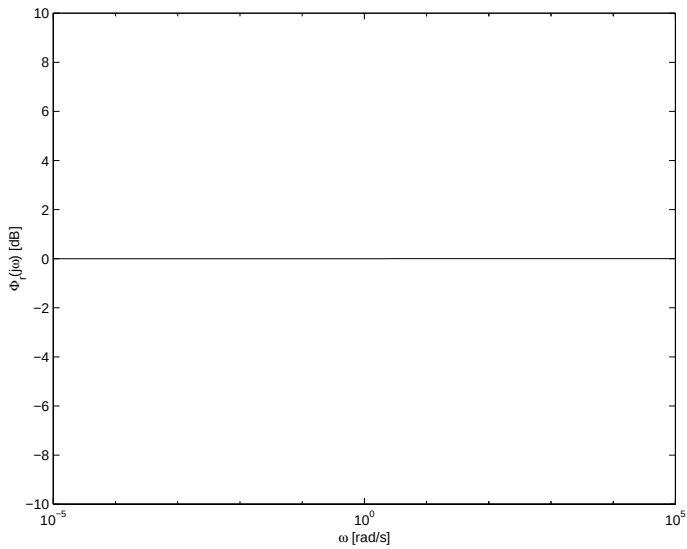


Figure 5.5: Spectrum $\Phi_r(j\omega)$.

It is straightforward to realize that innovation filters preserve any fault detectability properties since $\varphi(s)$ is invertible, i.e. the number of outputs of an innovation filter equals the dimension of $\mathcal{N}_L(M(s))$. However, if an innovation filter does not exist, there may very well exist a whitening residual generator with desirable fault detectability properties which was the case in this example.

5.5.2 Example with purely imaginary zeros

To study how purely imaginary zeros of $Z(s)$ influences the design procedure, consider the following system:

$$y = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{s(s+1)} \end{bmatrix} (u+f) + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} n$$

Simple calculations by hand gives that no innovation filter exists and all whitening residual generators can be written as

$$Q(s) = v(s)\frac{1}{\sqrt{5}}(y_1 - \frac{1}{s+1}u) \tag{5.24}$$

where $v(s)$ is any scalar all-pass link. Now, lets see how the design algorithm outlined in Section 5.4.2 arrives at the same conclusion.

Forming $M_s(s)$, computing $N_{M_s}(s)$ and $Z(s)$ gives

$$Z(s) = \begin{bmatrix} -25s^2+5 & -11s^2+16s-5 \\ -11s^2-16s-5 & -5s^2+5 \end{bmatrix}$$

which has zeros $\{0, 0, \pm 1\}$. This gives that $Z(s)$ has zeros on the imaginary axis and $p_{\text{imag}}(s) = s$.

$$T(s) = \begin{bmatrix} s^2+s & 0 & -s \\ 0 & s^2+s & -1 \end{bmatrix} \quad \wedge \quad U(s) = \begin{bmatrix} 0.089975 & -1.1144 \\ -0.24782 & -2.4877 \end{bmatrix}$$

Since the first, but not the second row, is divided by $p_{\text{imag}}^{-1}(s)$, the numerator of $\eta(s)$ can be chosen using (5.21) and rule (5.22) as e.g.

$$n(s) = [1\ 0]U(s)$$

The scalar denominator $d(s)$ is easily obtained by scalar spectral factorization (here it is actually even more simple since $n(s)$ is constant). Performing the left division $p_{\text{imag}}^{-1}(s)\ \text{adj}\, P(s)N_{M_s}(s)P_x$ gives a zero remainder as expected and a quotient $N(s) = [(s+1)\ 0\ -1]$. Realization of the residual generator, where the imaginary zeros has been cancelled, is then

$$Q(s) = [0.4472 \quad 0 \quad -\frac{0.4472}{s+1}]$$

which, since $1/\sqrt{5} \approx 0.4472$ is identical to (5.24).

5.6 White residuals

Why is white residuals desirable and are there advantages with white residuals compared to low-pass filtered residuals to be thresholded?

One reason for white residuals is that basic change detection algorithms, e.g. cusum (Basseville and Nikiforov, 1993), is directly applicable. Although these algorithms are optimal, the algorithm relies on deep knowledge of amplitude distribution of the residual and also the type of change (step/ramp/etc.). Often, such knowledge is not available and a perhaps more important advantage with white residuals is that the simple thresholding test for testing if the residual is small can be replaced by a whiteness test of the residual.

The next section describes a basic whiteness test and Section 5.6.2 shows a few simulation results where a whiteness test based detection is compared with a low-pass and threshold approach. The simulation study does not provide any hard theoretical results on the merits of either method, but still indicates some properties.

5.6.1 Whiteness tests

A whiteness tests is a test to decide between the hypotheses

$$H_0 : n(t) \text{ is white} \qquad H_1 : n(t) \text{ is coloured} \tag{5.25}$$

where $n(t)$ is the signal being analyzed. This section is mainly based on (Söderström and Stoica, 1989) and (Mehra and Peschon, 1971).

Typically, a test quantity to separate H_0 from H_1 is based on an estimate of the covariance function c_k. A basic property of white noise processes is that $c_k = \sigma^2$ if $k = 0$ and $c_k = 0$ for $k \neq 0$. The covariance function of $n(t)$ is defined (for zero-mean stationary stochastic processes),

$$c_k = E\{n(t)n(t-k)\} \quad k = 0, 1, \ldots$$

and an estimate of c_k can be obtained from data as

$$\hat{c}_k = \frac{1}{N}\sum_{t=1}^{N} n(t)n(t+k) \quad k = 0, 1, \ldots \tag{5.26}$$

Let $\hat{c}$ denote the vector

$$\hat{c} = \begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_{n_k} \end{pmatrix},$$

then it holds that (asymptotically)

$$\frac{N\hat{c}^T\hat{c}}{\hat{c}_0} \sim \chi^2(n_k) \tag{5.27}$$

Thus, a test quantity for hypotheses (5.25) can be formed by estimating c_0 to c_{n_k} from M data points and calculate

$$T(x) = \frac{M\hat{c}^T\hat{c}}{\hat{c}_0} \tag{5.28}$$

The null hypothesis is rejected when $T(x) > J$. The thresholds are selected such that the false alarm rate is lower than a specified level α. The threshold selection is done by assuming asymptotic properties (5.27) of $T(x)$, i.e. assume M large enough.

One nice property of the test $T(x)$, assuming asymptotic properties, is that it is invariant to distribution and input noise power.

5.6.2 Simulations and comparisons

In this section, a few simulations is done to compare the whiteness test from the previous section with a simple LP-and-threshold test. The thresholds in both tests is set to achieve a false-alarm rate of $\alpha = 0.01$.

In the simulations, the system under consideration is given by

$$y = G_u(q)u + n + f$$

i.e. sensor noise and a sensor fault is considered. The "raw" residual is given by

$$r_{raw} = y - G_u(q)u = n + f$$

In the simulations, n is white Gaussian noise and y, u is collected during 100 seconds, sampled with sampling frequency 10 Hz. The residual is then evaluated, either with a white-noise test or a LP-and-threshold test.

The parameters in the whiteness is set to (without any excessive tuning) $M = 80$ and $n_k = 20$. In the LP-and-threshold test, a first order LP-link with unit DC-gain is used, i.e.

$$H_{lp}(z) = \frac{1-a}{z-a}$$

The parameter in the test quantity is therefore the placement of the pole in the filter. The threshold is set by (correctly) assuming Gaussian distribution and unit variance. In the simulations, four different pole placements with different cut-off frequencies are used, $a \in \{0.9, 0.95, 0.97, 0.99\}$.

Three kinds of faults are simulated

1. Step fault. The fault signal is

$$f(t) = \begin{cases} 0 & t < 50 \\ 0.85 & t \geq 50 \end{cases}$$

The simulation results is shown in Figures 5.6, 5.7, and 5.8.

2. Sinus fault. The fault signal is

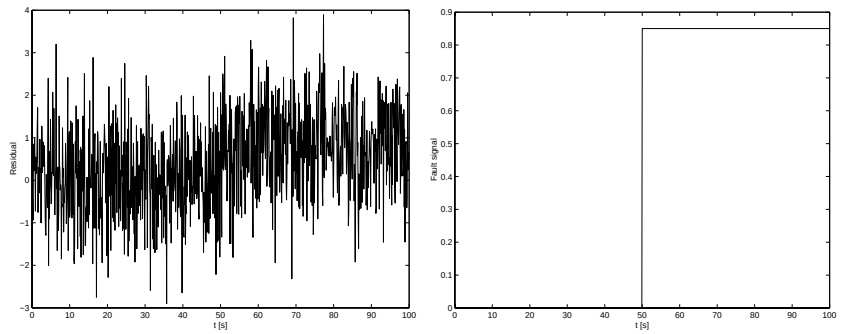
$$f(t) = \begin{cases} 0 & t < 50 \\ 1.5\sin(\pi t) & t \geq 50 \end{cases}$$

The simulation results is shown in Figures 5.9, 5.10, and 5.11.

3. Ramp fault. The fault signal is

$$f(t) = \begin{cases} 0 & t < 50 \\ \frac{1.5}{50}(t-50) & t \geq 50 \end{cases}$$

The simulation results is shown in Figures 5.12, 5.13, and 5.14.

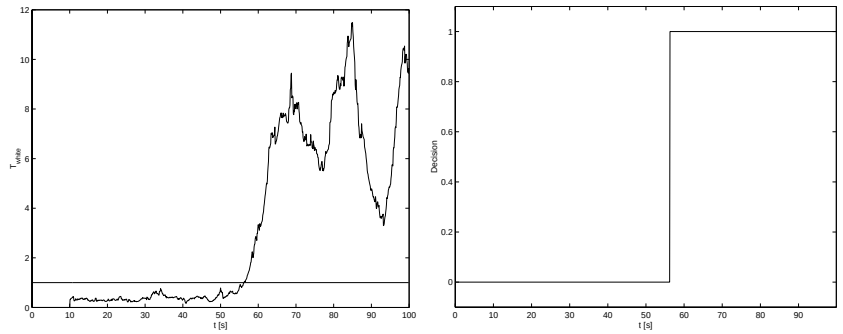


(a) Residual (b) Fault signal

Figure 5.6: The raw residual and fault signal in the step-fault simulation.

5.6.3 Simulation conclusions

From this simulation study it is not possible to draw any hard theoretical conclusions regarding the effectiveness of a whitening test compared to a low-pass and threshold test. The simulation study does however indicate that, in the case studied, the whiteness test is generally better to handle a wide variety of fault signal characteristics than the lp-and-threshold approach. This is achieved at the expense of relying on additional stochastic assumptions of signal properties.



(a) Test quantity (b) Decision

Figure 5.7: Test quantity and the decision, i.e. thresholded test quantity, for the whiteness based test in the step-fault simulation.

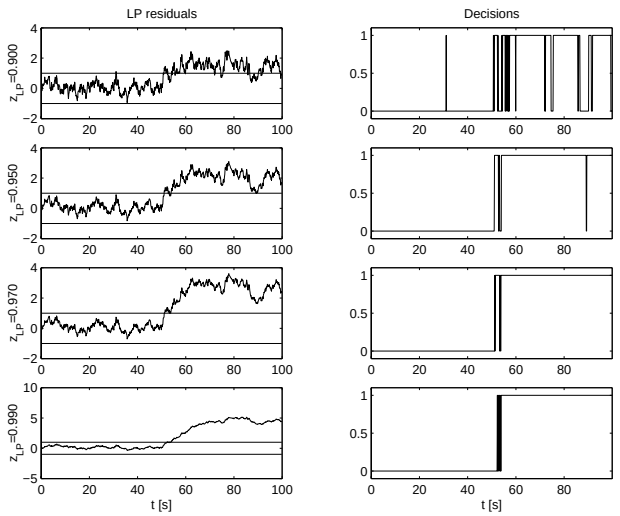
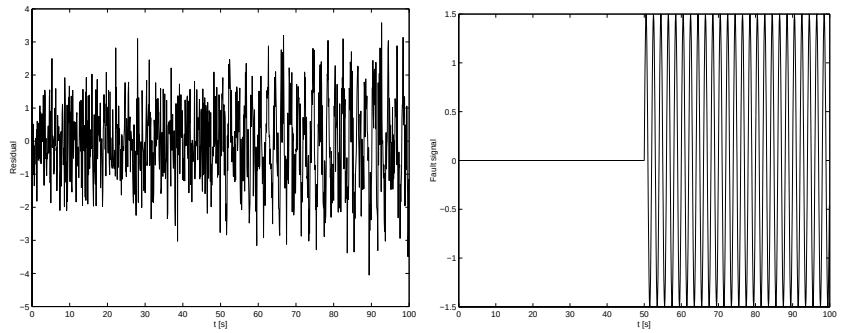
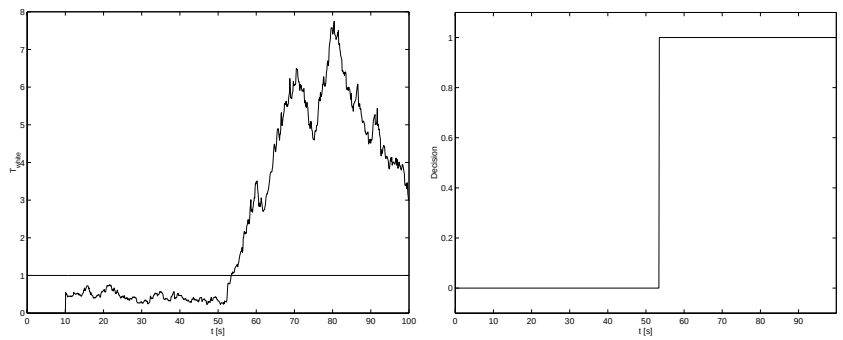


Figure 5.8: Residuals and decisions for the LP-and-threshold test in the step-fault simulation.



(a) Residual (b) Fault signal

Figure 5.9: The raw residual and fault signal in the sinus-fault simulation.



(a) Test quantity (b) Decision

Figure 5.10: Test quantity and the decision, i.e. thresholded test quantity, for the whiteness based test in the sinus-fault simulation.

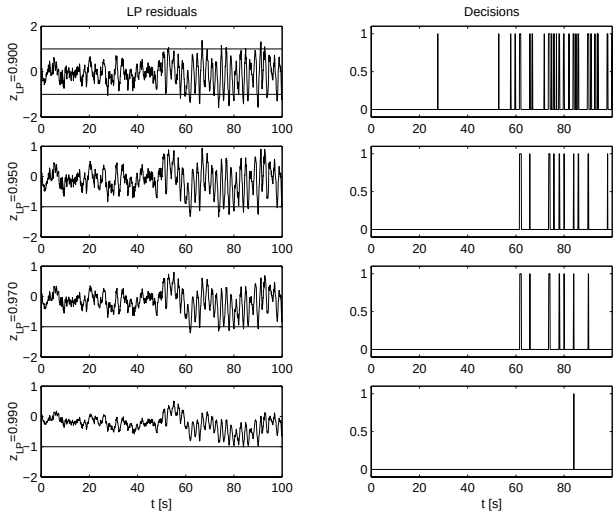
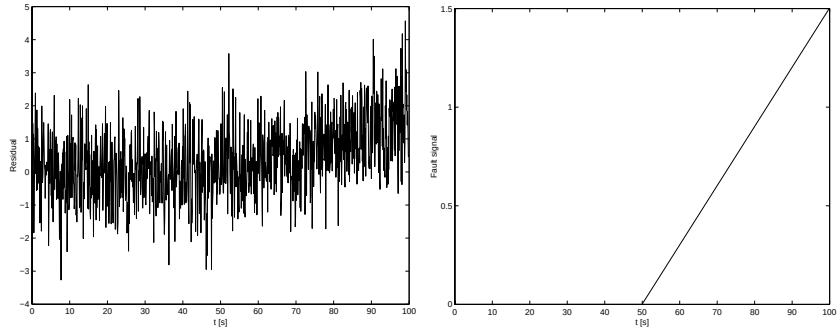
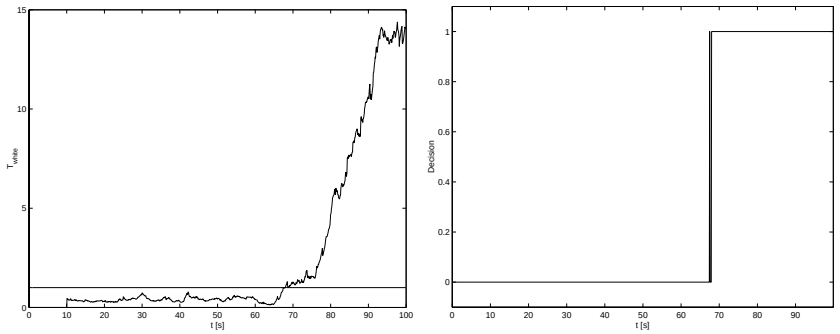


Figure 5.11: Residuals and decisions for the LP-and-threshold test in the sinus-fault simulation.



(a) Residual (b) Fault signal

Figure 5.12: The raw residual and fault signal in the ramp-fault simulation.



(a) Test quantity (b) Decision

Figure 5.13: Test quantity and the decision, i.e. thresholded test quantity, for the whiteness based test in the ramp-fault simulation.

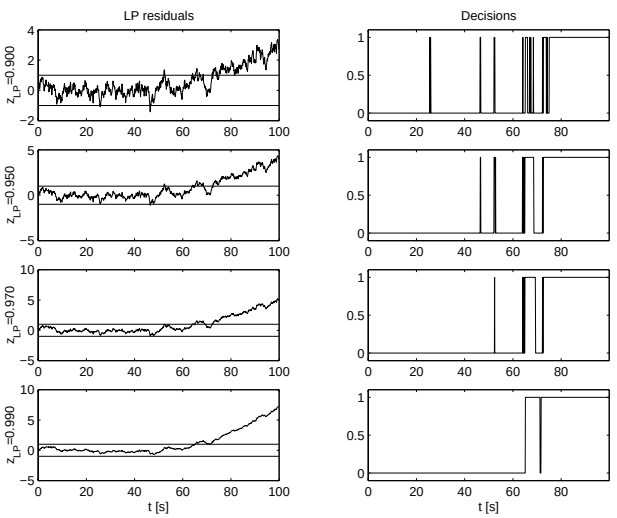


Figure 5.14: Residuals and decisions for the LP-and-threshold test in the ramp-fault simulation.

5.7 Time-discrete systems

In the deterministic case, time-discrete systems and time-continuous systems could be handled identically by just replacing s with z and proper with causal as shown in Chapter 3. In the stochastic case, small but important differences exists which are briefly discussed below.

First, a few words on the discrete time spectral factorization problem. Let $r = H(z)n$ with n unit covariance white noise, then the spectrum of r is given by

$$\Phi_r(\omega) = H(e^{j\omega T})H^T(e^{-j\omega T})$$

Thus, a time-discrete version of (5.8) becomes:

$$Z(z) = N_{M_s}(z)\begin{bmatrix} D_n \\ B_n \end{bmatrix}\begin{bmatrix} D_n \\ B_n \end{bmatrix}^T N_{M_s}^T(z^{-1})$$

In discrete time, a matrix $A(z)$ is para-hermitian if $A(z) = A^T(z^{-1})$. A J-spectral co-factorization of $Z(z)$ then becomes

$$Z(z) = P(z)JP^T(z^{-1})$$

where $P(z)$ have all zeros inside and on the unit circle. However, here only factorizations with signature $J = I$ and $Z(z)$ full rank on the unit-circle is considered. This because, to the authors knowledge, no numerically reliable factorization algorithm exists for the general case. In Polynomial Toolbox 2.5 for Matlab 5 (2001), an algorithm described in (Ježek and Kučera, 1985) is used.

The main difference between the time-continuous and time-discrete cases is that properness of the residual generator can always be achieved. This is immediate since a non-proper (non-causal) filter can always be made proper (casual) by inserting a number of time-delays and since z^{-1} is an all-pass link, the whiteness property is not violated. Thus, it is immediate to prove that the existence conditions for full-rank innovation filters and whitening residual generators is identical to the time-continuous case where the properness condition has been removed. For example, the existence of discrete time innovation filters is given by.

Theorem 5.7. *If $Z(z)$ is full rank, an innovation filter exists if and only if $Z(z)$ has no roots on the imaginary axis.*

Therefore, for the innovation filter design, the algorithmic restriction to $Z(z)$ with no zeros on the imaginary axis is no restriction. However in the whitening residual generator case this is a restriction as illustrated by the following small example which shows how the continuous algorithm directly transfers to the time-discrete case.

$$y = \begin{bmatrix} \frac{1}{z-a} \\ \frac{1}{(z-1)(z-a)} \end{bmatrix}(u+f) + n$$

with $|a| < 1$. It is clear that a residual generator $r = y_1 - \frac{1}{z-a}u$ produces white residuals in the fault-free case, an innovation filter does however not exist. Straightforward calculations give that for any residual generator it holds that the transfer function from noise to residual is given by

$$r = \varphi(z) \begin{bmatrix} z-a & 0 \\ 0 & 1-z \end{bmatrix} n \tag{5.29}$$

This gives that $\Phi_r(z)$ is given by

$$\Phi_r(z) = \varphi(z)Z(z)\varphi^T(1/z) = \varphi(z)P(z)P^T(1/z)\varphi^T(1/z)$$

with

$$P(z) = \begin{bmatrix} z-a & 0 \\ 0 & z-1 \end{bmatrix}$$

Therefore, $Q(z) = \eta(z)P^{-1}(z)N_{M_s}(z)P_x$ is a whitening residual generator if and only if there exists a $\eta(z)$ such that

$$Q(z) \text{ stable} \quad \wedge \quad \eta(z)\eta(1/z) = 1$$

One such choice is $\eta(z) = [-1 \;\; 0]$ which gives the residual generator

$$r = y_1 - \frac{1}{z-a}u$$

i.e. same as (5.29). However, to the authors knowledge, no computational tools seems to be available that can handle general cases with zeros on the unit circle.

5.8 The singular case

This section deals with the case when it is possible to perfectly decouple the noise. This is called the singular case since perfect decoupling of the noise is possible if and only if the matrix $Z(s)$ is singular.

5.8.1 Singular complications

Now follows a version of Lemma 5.1 where the non-singularity assumption is removed. After that, a discussion on the difficulties of formulating singular versions design methodologies for innovation filters and whitening residual generators using this polynomial approach.

Lemma 5.6. *There exists a $\varphi(s)$ such that the linear time-invariant filter $Q(s) = \varphi(s)N_{M_s}(s)P_x$ produces white residuals if and only if $\varphi(s)$ can be written*

$$\varphi(s) = [\eta(s) \;\; \zeta(s)]P^{-1}(s)$$

where $P(s)$ is a spectral co-factor of $Z(s)$, $\eta(s)\eta^T(-s) = \Psi$, and $[\eta(s) \;\; \zeta(s)]$ is partitioned according to the signature of the spectral co-factorization.

Proof. According to the proof of Theorem 5.1, the spectrum of r can be written

$$\Phi_r(s) = \varphi(s)Z(s)\varphi^T(-s) \tag{5.30}$$

Note that $Z(s)$ is a para-hermitian polynomial matrix. Now, let $P(s)$ be a spectral co-factor and J a signature of $Z(s)$, i.e.

$$Z(s) = P(s)JP^T(-s) \tag{5.31}$$

Since $Z(s)$ is positive semidefinite on the imaginary axis it is positive semidefinite in the whole complex plane. This implies that the signature of $Z(s)$ has the form

$$J = \begin{bmatrix} I_r & \\ & 0_{m-r} \end{bmatrix}$$

where $r = \text{rank } Z(s)$. Let $P_1(s)$ be the columns of $P(s)$ that corresponds to the non-zero part of the signature. Insertion of (5.31) into (5.30) and denoting $\eta(s) = \varphi(s)P_1(s)$ gives

$$\Phi_r(s) = \varphi(s)P(s)JP^T(-s)\varphi^T(-s) = \eta(s)\eta^T(-s)$$

which also gives that $\Phi_r(s) = \Psi$ according to the assumptions in the theorem. The parameterization matrix $\varphi(s)$ is found by solving for $\varphi(s)$ in the equation

$$\eta(s) = \varphi(s)P_1(s) \tag{5.32}$$

If $Z(s)$ is full-rank, $P_1(s) = P(s)$ will be a square full-rank matrix and (5.32) has only one unique solution $\varphi(s) = \eta(s)P^{-1}(s)$. If $Z(s)$ is rank deficient, several solutions exists, parameterized the matrix $\zeta(s)$ as

$$\varphi(s) = \eta(s)P_1^{-L}(s) + \zeta(s)N_{P_1}(s)$$

where $P_1^{-L}(s)$ is any left-inverse of matrix $P_1(s)$ and $N_{P_1}(s)$ is a basis for the left null-space of $P_1(s)$. The existence of such a left-inverse is ensured by the full column-rank property of $P_1(s)$. A stable inverse can be found by the inverse of $P(s)$. Let

$$P^{-1}(s) = \begin{bmatrix} P_{1i}(s) \\ P_{2i}(s) \end{bmatrix}$$

then it holds that $P_{1i}(s)$ is a stable left inverse of $P_1(s)$ and P_{2i} is a basis for the null-space of $P_1(s)$. Thus, in the rank deficient case, $\varphi(s)$ satisfies (5.32) if and only if it can be written

$$\varphi(s) = [\eta(s)\ \zeta(s)]P^{-1}(s)$$

■

Remark 1:An example of an inverse $P_1^{-L}(s)$ from the proof is the Moore-Penrose inverse. However, even though $\det P(s)$ is Hurwitz, the Moore-Penrose inverse of $P_1(s)$ is not necessarily stable.
Remark 2: One might argue that the parameterization from the theorem introduces unnecessary poles in $\varphi(s)$ since whole $P(s)$ is inverted, not only $P_1(s)$ as needed. However, as shown in Section 5.2.1, the spectral factor $P(s)$ can always be written on the form

$$P(s) = U(s) \begin{bmatrix} \bar{P}(s) & 0 \\ 0 & I \end{bmatrix}$$

where $P(s)$ is partitioned according to the signature J, matrix $U(s)$ is unimodular, and $\bar{P}(s)$ is a spectral factor of the non-singular part. Thus, all poles in $P^{-1}(s)$ origins from the non-singular part of the spectral factorization.

Now, existence conditions of innovation filters and whitening residual generators would be performed like in Section 5.4. Such conditions that are easily computable is difficult to state mainly because

1. The spectral factorization of a singular para-hermitian matrix is not canonical, i.e. we can not assume generic row-degrees of a spectral factor as in Theorem 5.5. To the authors knowledge, no canonical spectral factorization exists for the singular case and no characterization of the row-degrees of non-canonical spectral factors is available. This problem is also stated, in another problem setting, in the conclusions of (Šebek, 1990). Therefore, easily computable results like Corollary 5.1 and Theorem 5.6 is not available.

2. The freedom included in the design with $\zeta(s)$ from Lemma 5.6 to ensure properness and stability of the residual generator is not yet fully understood.

5.8.2 When does the non-singular case occur?

Now, a result giving necessary condition for the problem to be non-singular where the results from Section 5.4 is directly applicable.

Theorem 5.8. *A sufficient condition for $Z(s)$ to be non-singular is*

$$\mathit{rank} \begin{bmatrix} D_n \\ B_n \end{bmatrix} \geq m - \mathit{rank} \begin{bmatrix} D_d \\ B_d \end{bmatrix}$$

Proof. If $Z(s)$ is non-singular case it is possible to achieve full decoupling of the noise n. This is evident by following the proof of Lemma 5.6 and letting $\eta(s) = 0$ and $\zeta(s) \neq 0$ which will give a residual generator (always possible to get proper and stable) such that n is decoupled in the residual.

Following the proof of Theorem 5.1 it is seen that this is achievable if and only if the left null-space of

$$M(s) = \begin{bmatrix} C & D_d & D_n \\ -(sI - A) & B_d & B_n \end{bmatrix}$$

is non-empty. A dimensionality analysis of $M(s)$ immediately gives the theorem. ■

The following corollary follows directly:

Corollary 5.2. *Let all sensors in process (5.1) be subjected to measurement noise, then $Z(s)$ is non-singular.*

Proof. If all sensors are subjected to noise, rank $D_n = m$ and Theorem 5.8 is trivially fulfilled. ■

5.9 Stochastic descriptor example

In Section 3.7 it was shown how the minimal polynomial basis approach was applicable to also nominal descriptor systems. It is straightforward to realize that the extension described in this chapter for stochastic systems is equally applicable in the stochastic descriptor case.

The same calculations as in the proof of Theorem 5.1 gives that Equation (5.4) also holds for descriptor systems, i.e. the design algorithm is exactly as for the non-descriptor case with $M_s(s)$ given by (3.24).

Again, consider the descriptor example from Section 3.7 with sensor noise added, i.e. the stochastic descriptor model is given by

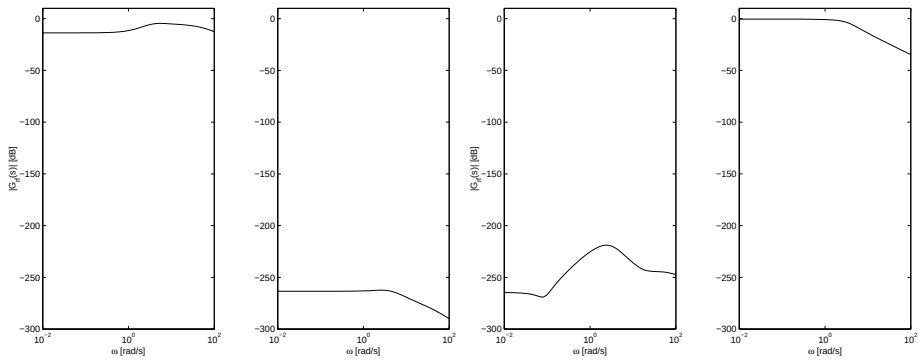
$$\begin{aligned} E\dot{x} &= Ax + B_u u + B_{f_1} f_1 + B_{f_2} f_2 \\ y &= Cx + n \end{aligned}$$

The design goal was to design residual generators $Q_1(s)$ and $Q_2(s)$, decoupling f_2 and f_1 respectively. In both cases, innovation filters exists and computing the spectral co-factor for $Z(s)$ in the first design gives

$$P(s) = \begin{bmatrix} -42s^2 - 2.6 \cdot 10^2 s - 6.5 \cdot 10^2 & 0.013s^2 + 0.99s - 42 & \cdots \\ 0.12s + 0.58 & -0.23s - 12 & \cdots \\ -0.38s - 1.8 & -0.9s - 44 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} \cdots & 0.0019s^2 - 0.13s - 84 \\ \cdots & 0.73s + 36 \\ \cdots & -0.42s - 20 \end{bmatrix}$$

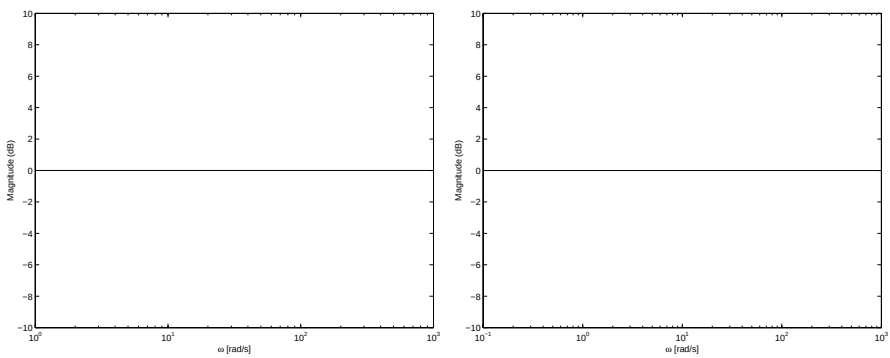
The free parameter $\eta(s)$ is chosen to be $\eta(s) = 1/\sqrt{3}[1\ 1\ 1]$. The procedure is repeated for the second residual generator $Q_2(s)$ and Figure 5.15 shows the magnitude of the transfer functions from fault f_1 and f_2 to residual 1 and 2 respectively. The figure also reveals that fault f_1 will be more difficult to detect than fault f_2 since both residuals have the same fault-free behavior and the gain from f_1 to r_1 is significantly lower than the f_2 to r_2 gain. Figure 5.16 demonstrates that the whiteness property of the residuals in the fault-free case holds since the spectra are constant for all ω.



(a) Fault sensitivity in residual 1

(b) Fault sensitivity in residual 2

Figure 5.15: Transfer functions from both faults to residual 1 and 2 respectively. The design specifications are met since fault f_2 is decoupled in residual 1, fault f_1 in residual 2 and the residuals are sensitive to the remaining faults.



(a) $\Phi_{r_1}(j\omega)$

(b) $\Phi_{r_2}(j\omega)$

Figure 5.16: Spectrum of residuals r_1 and r_2 in the fault-free case.

5.10 Conclusions

Linear residual generator design for stochastic models on state-space, descriptor, and transfer-function form has been considered, both in continuous-time and discrete time. The basic design formulation is based on innovation filters as formulated by Nikoukhah (1994) where the basic stochastic design requirement is that the residuals should be zero mean and white in the fault-free case. The problem formulation is also further developed to whitening residual generators, a more general class of residual generators.

The main extension introduced by considering whitening residual generators instead of innovation filters is to consider the existence of *any* scalar residual generator producing white residuals. Previously with innovation filters, only the case where whiteness of the residual can be achieved without loosing *any* design freedom is considered. In such cases, the stochastic requirements can never reduce fault detectability in the residual. Here however, all residual generators that satisfies the whiteness requirement is parameterized. If there among these exist a residual generator with acceptable fault sensitivity, a feasible design has been found.

The algorithm is an extension of the deterministic design algorithm Chapter 3, where in a first step all residual generators for the deterministic problem is derived. Then, in a second step, all whitening residual generators are found by post filtering. Since the algorithm from Chapter 3 handled descriptor systems, so do this stochastic extension. In contrast to the deterministic design problem, the continuous and discrete-time problems is not analogous and a small, but important, difference exists. In the continuous-time case, an innovation filter or whitening residual generator may not exists because no proper whitening residual generator exists. For discrete-time, this is never an issue since a necessary number of time-delays always can be introduced and a time-delay is an all-pass link, i.e. the whiteness property is not violated. Computationally, the two main steps in the design algorithm is extraction of a polynomial basis for the left null-space of a polynomial matrix (as in the deterministic case) followed by a J-spectral co-factorization of a para-hermitian polynomial matrix. For both these operations there exists good numerical tools. To show the simplicity of the design methods, full Matlab implementation of innovation filter design is included in Appendix 5.A.

The design algorithm is successfully demonstrated on a number of non-trivial examples that in detail illustrates different properties of the algorithm and the design problems. One key observation is shown by example in which an innovation filter, as proposed by (Nikoukhah, 1994), does not exist. However, due to the extended class of residual generators considered, a residual generator do exist that fulfills all design specifications.

Appendix

5.A innovationfilter.m

```
function Q=innovationfilter(A,Bu,Bd,Bn,C,Du,Dd,Dn)
% INNOVATIONFILTER - Innovation filter design
%
%    Given a fault-free system desciption
%         x'= Ax + Bu u + Bd d + Bn n
%         y = Cx + Du u + Dd d + Dn n
%
%    Design (if one exists) an innovation filter Q(s).
%    Any innovation filter W(s) can then be parameterized by
%         W(s) = L(s)Q(s)
%    where L(s) is invertible and satisfies L(s)L'(s)=I
%
%    Syntax: Q=innovationfilter(A,Bu,Bd, Bn,C,Du,Dd,Dn)

tol=1e-5; % Tolerance for detecting purely imaginary zeros

nx  = length(A);    % Number of states
nmeas = size(C,1);  % Number of measurements
ndist = size(Bd,2); % Number of disturbances

Mx = [C Dd;-(s*eye(nx)-A) Bd];
Px = [eye(nmeas) -Du;zeros(nx,nmeas) -Bu];
Nmx = null(Mx.').';

Z = Nmx*[Dn;Bn]*[Dn;Bn]'*Nmx';

if rank(Z)<size(Z,1)
  error('Singular cases is not covered by this function');
end

if (max(deg(Nmx,'row')-deg(Nmx*[Dn;Bn],'row'))>0) | ...
      (sum(abs(real(roots(Z)))<tol)>0)
  error('No innovation filter exists');
end
[P,J] = spf(Z.'); P = P.'; %% Spectral factorization

[Qa,Qb,Qc,Qd] = lmf2ss(Nmx*Px,P);
Q = minreal(ss(Qa,Qb,Qc,Qd));
```

6

Residual Generation Based on Uncertain Linear Models

This chapter addresses the problem of synthesizing and analyzing robust residual generators in the presence of parametric uncertainties and deterministic additive disturbances that influences the process. This problem is well studied and there exists many papers with different design focuses on the subject, e.g. (Frank and Ding, 1994; Mangoubi et al., 1995; Edelmayer et al., 1994; Mangoubi et al., 1994, 1992; Eich and Oehler, 1997; Sauter et al., 1997). A common trait of these works is that an optimization problem is stated to synthesize an optimally robust residual generator. Among these optimization criterions two common classes of optimization criterions stand out, one can be stated on the form:

$$\min_{Q(s)} \frac{\|G_{rd}(s)\|}{\|G_{rf}(s)\|}$$

where $Q(s)$ is the residual generator. Different choices of norm $\|\cdot\|$ have been proposed. With such a criterion, model uncertainties are modeled as additive input signals. However, model uncertainty most often appears as parametric uncertainty where physical parameters can only be determined within a range. Theory and methodology for design of robust controllers for such systems is available in μ-synthesis of $\mathcal{H}_\infty/\mathcal{H}_2$ controllers (Zhou et al., 1995). This theory is also applicable to the filtering problem, and is therefore easily adopted to the residual generation problem. This leads to a second class of optimization criterions that is in common use:

$$\min_{Q(s)} \frac{\|r - f\|}{\|d\|}$$

i.e. the residual should be a fault estimator in the presence of uncertainties d. Since focus in this chapter is on parametric uncertainties, this type of optimization criterion is used. A main observation is that using the residual as a fault estimator is not always a good idea; a more refined criterion is often necessary. For this purpose, it is advantageous to introduce a *reference model* that describes the desired behavior of the residuals with respect to faults. A theory is developed where the reference model idea and a new optimization criterion are key elements. It is shown how this reference model is an intuitive design parameter in the synthesis problem, but that it has to be chosen with care.

6.1 Robust residual generation

The system under consideration is again assumed to be on the form

$$y = G_u^{\Delta}(s)u + G_d^{\Delta}(s)d + G_f^{\Delta}(s)f \tag{6.1}$$

like in (3.1) with a small difference, the superscript Δ. Matrices $G_u^{\Delta}(s)$, $G_d^{\Delta}(s)$, and $G_f^{\Delta}(s)$ are all rational transfer matrices and the superscripts Δ indicate that the model is subject to bounded parametric uncertainties. The class of uncertainties studied here is a quite general class where it is assumed that the uncertain system can be described by a linear fractional transformation (LFT). Section 6.3.2 further clarifies which types of systems that can be represented in this way. More details on linear fractional transformations can be found in e.g. (Zhou et al., 1995, Chapter 10).

The residual generator is, as in previous chapters, a finite dimensional linear filter $Q(s)$ that uses available known signals, i.e. y and u, to form a residual, r, that can be used to detect and isolate the different faults f.

$$r = Q(s)\begin{pmatrix} y \\ u \end{pmatrix} \tag{6.2}$$

The basic requirement on $Q(s)$, besides being $\mathcal{RH}_\infty$ (i.e. stable and proper), is that the residual should be insensitive to control actions, u, and disturbances, d, but it should also be sensitive to faults f.

The principle of fault isolation considered here is again structured residuals as described in Section 2.1.3, where a subset of faults are decoupled in each residual. Note that in Chapter 3, non-monitored faults were included in the d-vector. This re-organization of d and f vectors is, for the sake of convenience, not performed in this chapter. The fault-vector f will in this chapter include all modeled faults and the vector d all modeled disturbances.no

By generating a set of residuals where different subsets of faults are decoupled in each residual, fault isolation is possible. Inserting (6.1) into (6.2) gives

$$r = Q(s)\begin{bmatrix} G_u^{\Delta}(s) & G_d^{\Delta}(s) \\ I & 0 \end{bmatrix}\begin{pmatrix} u \\ d \end{pmatrix} + Q(s)\begin{bmatrix} G_f^{\Delta}(s) \\ 0 \end{bmatrix} f \tag{6.3}$$

Nominally, to achieve decoupling of u and d, the first term of (6.3) must be 0 while the second term must be $\neq 0$ which was discussed in detail in Chapter 3. This issue will be re-addressed in Section 6.5 where the freedom available in the robust design problem is explored. However, with uncertain models it is in most cases impossible to get the first term $= 0$ for all possible Δ, i.e. for all possible instances of uncertainties, without loosing some or all of the desired fault sensitivity. Note that it is not always so; in some cases parametric uncertainty can be transformed into unknown input signals that can be decoupled with methods based on nominal models, see e.g. (Patton and Hou, 1998). However, generally some tradeoff between sensitivity to faults and disturbance/uncertainty attenuation is required.

6.2 Reference model

When synthesizing a robust residual generator, it is desired that the design freedom available should be used to achieve both robust fault-free behavior and robust detection performance. The question arises how this performance should be formulated. A natural choice is to introduce a reference model, $R(s)$, to describe the desired behavior of the residual vector r, with respect to faults, f. Define *desired* residual behavior, $r_0(s)$, of the residual, via the *reference model* as

$$r_0 = R(s)f(s)$$

which then describes both fault-free behavior and the fault response in the residual. The matrix $R(s)$ is an arbitrary $\mathcal{RH}_\infty$ transfer matrix of appropriate dimensions. The idea of a reference model has successfully been used to describe signal behavior in other fields like controller design (Åström and Wittenmark, 1984) and adaptive control (Åström and Wittenmark, 1989).

It is of course necessary that the reference model, $R(s)$, contains the necessary structure for $Q(s)$ to be a residual generator corresponding to a desired influence structure.

Example 6.1
Suppose there are three modeled faults and it is desired to design a residual that:

1. Decouples the first fault
2. Responds to high frequency components ($\omega > \omega_h$) of the second fault
3. Responds to low frequency components ($\omega < \omega_l$) of the third fault

i.e. the residual should correspond to a row [0 1 1] in the residual structure. An example of $R(s)$ could then be

$$R(s) = \begin{bmatrix} 0 & \frac{s}{s+\omega_h} & \frac{1}{s/\omega_l+1} \end{bmatrix}$$

where ω_h and ω_l was chosen to reflect the frequency ranges of interest.

Later on in this chapter a design algorithm is outlined that tries to achieve the fault response given by the reference model while being as robust as possible against variation of the unknown parameters. However, it will later show in Section 6.4 that it is important to put some thought into the reference model selection since an ill specified reference model will lead to an optimization problem where the algorithm will spend unnecessary freedom on achieving the badly posed reference model. In the next section, the design algorithm and how to compute the solution is outlined and the reference model selection is revisited in Section 6.5.

6.3 Computation of a robust residual generator

This section will provide an algorithm, that is based on standard $\mathcal{H}_\infty$-optimization, that computes an optimally robust residual generator. First in Section 6.3.1, an optimization criterion is posed that incorporates the specified desired residual behavior, the reference model. Then in Section 6.3.2, the results needed to compute the optimally robust residual generator is presented.

6.3.1 Robustness criterion

The optimization criterion used here is formulated as a robust $\mathcal{H}_\infty$-filtering problem (Zhou et al., 1995), with an intuitive and appealing interpretation which is given after Eq. (6.6). The criterion is

$$J = \sup_{v \in \mathcal{L}_2} \frac{\|r_0(t) - r(t)\|_2}{\|v(t)\|_2} \tag{6.4}$$

where $v = [u^T \ f^T \ d^T]^T$. The norm $\|\cdot\|_2$ is defined as

$$\|u\|_2^2 = \int_0^\infty |u(t)|^2 dt$$

The optimization criterion J is thus the worst case distance between the residual r and the idealized residual r_0, defined by transfer matrix $R(s)$, normed by the size of the inputs. The optimal residual generator $Q(s)$ is the filter that minimizes J for all Δ such that $\|\Delta\| \leq \delta$ for some scalar $\delta > 0$.

The optimization criterion J can be rewritten as

$$J = \sup_{v \in \mathcal{L}_2} \frac{\|r_0 - r\|_2}{\|v\|_2} = \sup_{v \in \mathcal{L}_2} \frac{\|T_{zv}(s)v\|_2}{\|v\|_2} = \|T_{zv}(s)\|_\infty \tag{6.5}$$

where $z(t) = r_0(t) - r(t)$, and

$$T_{zv}(s) = \begin{bmatrix} -G_{ru}^\Delta(s) & \left(R(s) - G_{rf}^\Delta(s)\right) & -G_{rd}^\Delta(s) \end{bmatrix} \tag{6.6}$$

is the transfer matrix from $v(t)$ to $z(t)$. The transfer matrices from u to r, $G_{ru}^{\Delta}(s)$, from d to r, $G_{rd}^{\Delta}(s)$, and from f to r, $G_{rf}^{\Delta}(s)$ all depend on the residual generator $Q(s)$. Minimizing J, i.e. minimizing the ∞-norm of expression (6.6), has a simple interpretation. Keeping $\|T_{zv}(s)\|_\infty$ small ensures that the first and third element are small which makes sure that the influence from u and d on the residual are attenuated. The middle term keeps fault sensitivity, and also shapes the fault to residual transfer function $G_{rf}^{\Delta}(s)$ by minimizing the distance to the reference model $R(s)$.

The optimization/performance index minimizes the *absolute* difference between $R(s)$ and $G_{rf}^{\Delta}(s)$. A reasonable assumption is that it is the *relative* difference that need to be minimized, otherwise in e.g. high-gain models even very small relative errors will dominate the loss function and therefore move away optimization focus from robustness to fault sensitivity in an unwanted manner. Also, different signal levels between e.g. u and d have similar effects. Therefore is it important to normalize and/or weight the model appropriately to avoid such effects.

Remark

It is obvious that the optimal value of J also delivers constraints on the size (norm) of r in the fault-free and faulty case. Thus, the optimal value of the optimization criterion can be used to guide robust threshold selection. This issue will be briefly revisited in Section 6.8.

6.3.2 Computational framework

The above stated problem can be naturally formulated in an LFT (Linear Fractional Transformation)-framework, and then be solved by conventional methods from robust and optimal control, i.e. $\mathcal{H}_\infty$-filtering. Now follows a brief description on the formulation and computation of an optimal solution to (6.5). First, a standard definition:

Definition 6.1 (LFT). *Let M be a complex matrix partitioned as*

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{C}^{(p_1+p_2)\times(q_1+q_2)}$$

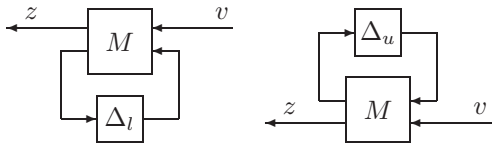
and let $\Delta_l \in \mathbb{C}^{q_2\times p_2}$ and $\Delta_u \in \mathbb{C}^{q_1\times p_1}$ be two other complex matrices. Then, provided that all inverses exists, the lower LFT of M with respect to Δ_l is defined as

$$\mathcal{F}_l(M, \Delta_l) \triangleq M_{11} + M_{12}\Delta_l(I - M_{22}\Delta_l)^{-1}M_{21}$$

and the upper LFT of M with respect to Δ_u is defined as

$$\mathcal{F}_u(M, \Delta_u) \triangleq M_{22} + M_{21}\Delta_u(I - M_{11}\Delta_u)^{-1}M_{12}$$

A graphical representation of these two transformations can be seen as:



where $\mathcal{F}_l(M, \Delta_l)$ and $\mathcal{F}_u(M, \Delta_u)$ are the transfer functions from v to z in Figure a and b respectively.

Now, to be able to use standard optimization theory, an LFT formulation of the optimization problem (6.6) stated in the Section 6.3.1 is needed. Figure 6.1 shows such a representation, where all parametric uncertainty is lumped into the block-diagonal matrix Δ, matrix $P(s)$ is an augmented plant description and $Q(s)$ is the residual generator. The LFT framework is quite general and

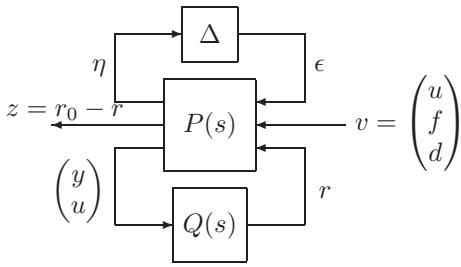


Figure 6.1: LFT formulation of residual generation problem

for example, all polynomial, rational functions can be represented by an LFT (Zhou et al., 1995). Thus, the class of uncertainties is that are covered by the LFT representation is quite large.

Now follows a small, static example illustrating an LFT formulation of a system with parametric uncertainties.

Example 6.2
Consider a rational, static transfer function with two uncertain parameters δ_1 and δ_2 given by

$$z = \frac{a + b\delta_1^2}{1 + c\delta_1\delta_2} w = Gw$$

Even though the influence from uncertainty is non-linear, it is straightforward

to realize that $G = \mathcal{F}_u(M, \Delta)$ with

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & b & 0 & -c & a \\ 0 & 0 & 1 & 0 & 0 \\ 0 & b & 0 & -c & a \end{bmatrix} \quad \Delta = \begin{bmatrix} \delta_1 I_3 & 0 \\ 0 & \delta_2 \end{bmatrix}$$

When the optimization problem is formulated according to Figure 6.1, there exist algorithms minimizing J with respect to $Q(s)$ by e.g. μ-synthesis. The algorithm used in this work is basic DK-iterations which, although heuristic and with no convergence guarantees, have in practice shown reliable performance (Balas et al., 1993). The parametric uncertainties are here assumed complex to be able to use standard tools in MATLAB. This gives a conservative solution since the parametric uncertainties modeled are physical constants which are real. A better solution would be to restrict the uncertainties corresponding to physical parameters to real values. This would lead to a mixed problem, with both real and complex uncertainties, which is addressed in e.g. (Helmersson, 1995). However, such topics are outside the scope of this thesis and does not alter the main ideas. The DK-iteration procedure finds a solution by iteratively solving for $Q(s)$ (the K-step) and iteratively solving for some scaling matrices (the D-step). Details on this procedure can be found in e.g. in (Zhou et al., 1995). Some details on the K-step is given here because it gives some insight in how to, in the residual generation problem, form the augmented plant description $P(s)$.

The standard $\mathcal{H}_\infty$-filtering problem

The K-step in the design of the residual generation is to solve a standard $\mathcal{H}_\infty$-filtering problem (Zhou et al., 1995).

Suppose a dynamic system is described by

$$\dot{x} = Ax + B_1 v \tag{6.7a}$$
$$r_0 = C_1 x + D_{11} v \tag{6.7b}$$
$$y_\infty = C_2 x + D_{21} v \tag{6.7c}$$

where y_∞ is the measurables/known variables and r_0 the signal that is to be estimated. To simplify expressions, the following standard notation is used

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = C(sI - A)^{-1}B + D$$

The filtering problem is then to find a linear estimator $r = Q(s)y_\infty$ that minimizes

$$\sup_{v \in \mathcal{L}_2} \frac{\|z\|_2}{\|v\|_2}$$

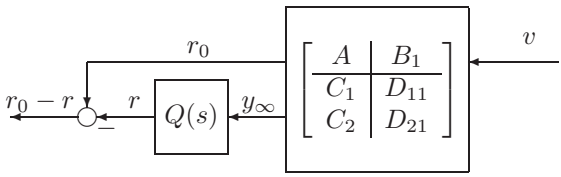
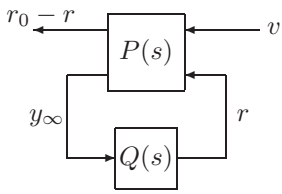


Figure 6.2: Block diagram of the $\mathcal{H}_\infty$-filtering problem.

where $z = r_0 - r$. Figure 6.2 shows a block representation of the filtering problem. The filtering problem in Figure 6.2 can then be reformulated in an LFT framework as is shown in Figure 6.3. The residual generation problem



$$P(s) = \left[\begin{array}{c|cc} A & B_1 & 0 \\ \hline C_1 & D_{11} & -I \\ C_2 & D_{21} & 0 \end{array}\right]$$

Figure 6.3: LFT formulation of the $\mathcal{H}_\infty$-filtering problem.

can be fitted into the problem formulation (6.7) by setting $v^T = [u^T \ f^T \ d^T]$, $y_\infty^T = [y^T \ u^T]$, and letting $r_0 = R(s)f$. Therefore the augmented matrix $P(s)$ is formed via (6.7) and Figure 6.3.

When $P(s)$ is formed, the problem is ready to be solved and a solution to the filtering problem is characterized by the following theorems, more or less, directly from (Zhou et al., 1995). The solution given by the theorems does not find the optimal solution because it is generally hard to obtain. However, a suboptimal solution where $\|\mathcal{F}_l(P,Q)\|_\infty < \gamma$ is more easy to come by. A *bisection* algorithm can then be used to iterate to a solution sufficiently close to optimum.

The first theorem gives an existence test of a solution for a given γ which can be used to iterate to a solution near optimum. The second theorem parameterizes all solutions that fulfills $\|\mathcal{F}_l(P,Q)\|_\infty < \gamma$ given that one exists. These results involve the solution of Ricatti equations and before the results are presented, some notation is needed. When solving Ricatti equations

$$A^*X + XA + XRX + Q = 0$$

it is useful to define an associated Hamiltonian matrix which has the form

$$H = \begin{bmatrix} A & R \\ -Q & -A^T \end{bmatrix}$$

where A, Q, and R are real $n \times n$ matrices with Q and R symmetric. In particular, we are interested in Ricatti equations where there exists a unique, stabilizing solution X. The set of all Hamiltonian matrices such that the associated Ricatti equation has such a unique, stabilizing solution is denoted $dom(Ric)$ and the solution with $X = Ric(H)$. Conditions for when $H \in dom(Ric)$ and other properties of Ricatti equations can be found in for example (Zhou et al., 1995).

Now, we are ready to state the results.

Theorem 6.1 (Existence). *Suppose (C_2, A) is detectable and*

$$\begin{bmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$$

has full row rank for all ω. Let D_{21} be normalized and D_{11} partitioned conformably as

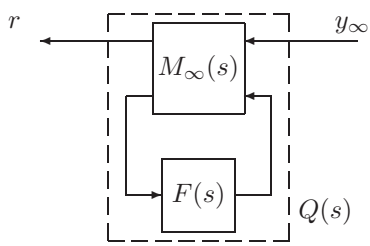
$$\begin{bmatrix} D_{11} \\ D_{21} \end{bmatrix} = \begin{bmatrix} D_{111} & D_{112} \\ 0 & I \end{bmatrix}.$$

Then there exists a causal filter $Q(s) \in \mathcal{RH}_\infty$ such that $J < \gamma$ if and only if $\overline{\sigma}(D_{111}) < \gamma$ and $J_\infty \in dom(Ric)$ with $Y_\infty = Ric(J_\infty) \geq 0$ where

$$\begin{aligned} \tilde{R} &= \begin{bmatrix} D_{11} \\ D_{21} \end{bmatrix} \begin{bmatrix} D_{11} \\ D_{21} \end{bmatrix}^* - \begin{bmatrix} \gamma^2 & 0 \\ 0 & 0 \end{bmatrix} \\ J_\infty &= \begin{bmatrix} A^* & 0 \\ -B_1 B_1^* & -A \end{bmatrix} - \begin{bmatrix} C_1^* & C_2^* \\ -B_1 D_{11}^* & -B_1 D_{21}^* \end{bmatrix} \tilde{R}^{-1} \begin{bmatrix} D_{11} B_1^* & C_1 \\ D_{21} B_1^* & C_2 \end{bmatrix}. \end{aligned}$$

Proof. See (Zhou et al., 1995). ■

Theorem 6.2 (Parameterization). *If the conditions in Theorem 6.1 are satisfied, then all rational causal filters $Q(s) = \mathcal{F}_l(M_\infty, F)$ satisfying $J = \|T_{zv}(s)\|_\infty < \gamma$ are parameterized by*



where $F(s) \in \mathcal{RH}_\infty$, $\|F(s)\|_\infty < \gamma$ and

$$M_\infty(s) = \left[\begin{array}{c|cc} A + L_{2\infty} C_2 + L_{1\infty} D_{112} C_2 & -L_{1\infty} D_{112} - L_{2\infty} & L_{1\infty} \hat{D}_{12} \\ \hline C_1 - D_{112} C_2 & D_{112} & \hat{D}_{12} \\ C_2 & I & 0 \end{array}\right]$$

where $\hat{D}_{12}$ *is any real matrix satisfying*

$$\hat{D}_{12}\hat{D}_{12}^* = I - \gamma^{-2} D_{111} D_{111}^*$$

and

$$\begin{bmatrix} L_{1\infty} & L_{2\infty} \end{bmatrix} = -\begin{bmatrix} B_1 D_{11}^* + Y_\infty C_1^* & B_1 D_{21}^* + Y_\infty C_2^* \end{bmatrix} \tilde{R}^{-1}.$$

Proof. Start with a general parameterization of $\mathcal{H}_\infty$ controllers, e.g. in (Zhou et al., 1995, p. 451) and utilize that $B_2 = 0$ and $D_{12} = -I$. Straightforward but lengthy calculations give the parameterization above. ■

6.4 Background example on reference model design

The idea of a reference model for residual behavior has been introduced, and was in the previous section used in the criterion for robust design. Before going into details of the design of the reference model, a simple stylized static example will illustrate the necessity of using a well formulated reference model. The discussion forms a background and leads to the detailed design in Section 6.5.

Example 6.3
Consider the static system

$$y_1 = (1 + \delta_1)x + 3f_1 \quad y_2 = (2 + \delta_2)x + 4f_2$$

where x is an unknown disturbance and $|\delta| < \frac{1}{2}$. Suppose it is desired that the disturbance x does not influence the residual while f_1 and f_2 has significant influence, which is the same as saying that r should respond to a row [1 1] in a residual structure.
The residual generator is formed as

$$r = q_1 y_1 + q_2 y_2$$

First attempt

A naive choice of $R(s)$ without any further thought might be to aim for unit gain from both faults, i.e. $R(s) = [1\ 1]$. A design is performed as described in the previous section. The resulting residual generator is denoted $q_a = [q_1^T\ q_2^T]^T$. Robustness is evaluated by determining the worst-case gain from the disturbance to the residual. This corresponds to measuring the size of the third term in (6.6). Since there is no control signal in this example, this measure indicates the performance of the filter in a no-fault situation.

$$\max_{|\delta|<\frac{1}{2}} \|k_{rx}\| = \max_{|\delta|<\frac{1}{2}} \|[(1 + \delta_1) \quad (2 + \delta_2)]q_a\| = 1.683$$

That is, the worst-case gain from x to r is $k_{rx} = 1.683$.

Second attempt

A little more thought on the problem leads to a second attempt based on a nominal design. If no uncertainty is present, then $\delta = 0$ and a natural choice of q would satisfy

$$q_2 = -\frac{1}{2}q_1$$

to completely decouple the disturbance x. The relation between the gain from f_1 to r, k_{rf_1}, and f_2 to r , k_{rf_2}, is then related by

$$\frac{k_{rf_1}}{k_{rf_2}} = \frac{3q_1}{4q_2} = -\frac{2 \cdot 3}{1 \cdot 4} = -\frac{3}{2}$$

Using this nominal information will significantly improve robustness properties of the design. Let the elements in $R(s)$ be proportional to the nominal gains from the faults to the residual k_{rf_1} and k_{rf_2},

$$R(s) = [-\frac{3}{2} \ 1] \tag{6.8}$$

Note that this choice of $R(s)$ also corresponds to a [1 1] residual structure. A new design with this $R(s)$ results in a q_b, normed to be of the same size as q_a, in the sense that $|q_a| = |q_b|$. This is important to enable a fair robustness comparison. The second design has robustness properties

$$\max_{|\delta|<\frac{1}{2}} \|[(1+\delta_1) \quad (2+\delta_2)]q_b\| = 0.7906$$

Comparison

Thus, a significant improvement on robustness properties is achieved in the second design compared to the first design. This is because in the first design, $R(s)$ reflected unrealistic performance specifications that inflicted unnecessary poor robustness properties on the residual generator. This issue will be further explored below and in the concluding example in Section 6.7.

It is worth noting that even in a no uncertainty case with $\delta_1 = \delta_2 = 0$, the first design attempt would not have found a solution where the unknown disturbance x is completely decoupled, even though such a solution exists. The reason for this, which is the same reason that made the second solution more robust, is that during optimization, freedom is spent to make $G_{rf}(s) - R(s)$ as small as possible. If an unrealistic $R(s)$ is used, freedom is spent on an impossible task that often is of no importance to the FDI problem. A suitable choice of $R(s)$ helps focusing on the robustness properties while keeping fault sensitivity and conforming to the specified performance. Note that it is straightforward to introduce weighting matrices for f, u, d, and Δ respectively to help the optimization procedure to get a proper trade-off between robustness- and

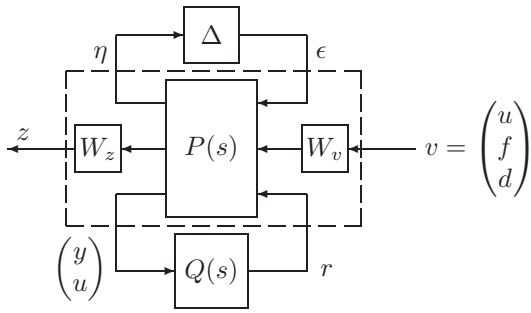


Figure 6.4: Weighting matrices

performance-focus in the optimization. This is an important non-trivial step in the design process. Examples of weighting matrices are shown in Figure 6.4 where $W_v(s) = \text{block-diag}[W_u(s), W_f(s), W_d(s)]$. The weighted counterpart of (6.6) is then

$$T_{zv}(s) = W_z(s)\left[-G^{\Delta}_{ru}(s)W_u(s) \quad [R(s) - G^{\Delta}_{rf}(s)]W_f(s) \quad -G^{\Delta}_{rd}(s)W_d(s)\right]$$

The issue illustrated above needs to be considered also when only one fault at a time is supposed to influence the residual, e.g. in a diagonal residual structure or when the notion of *don't care* (Nyberg, 1999b) is used. For example, consider a fault that is only *weakly* detectable (Nyberg and Nielsen, 1997), i.e. $G_{rf}(0) = 0$ for all residual generators. Figure 6.5 shows typical magnitude bode plot of the transfer function to a residual from a weakly detectable fault. Then, by specifying $R(s) = 1$, for low frequencies the second component in (6.6) will be large regardless of $Q(s)$, and the robustness properties will therefore be de-emphasized in the criterion. A more suitable reference model would then be something similar to

$$R(s) = \frac{s}{s + \omega_c}$$

6.5 Forming the reference model

From the example above it is clear that a poorly chosen reference model, i.e. a reference model with unrealistic performance properties, can result in a residual generator with unnecessary poor robustness properties. To form a methodology on how to select the reference model, *attainable* reference model is defined.

Definition 6.2 (Attainable Reference Model). *A reference model is said to be attainable if, with no model uncertainty, there exists a residual generator with the specified fault response.*

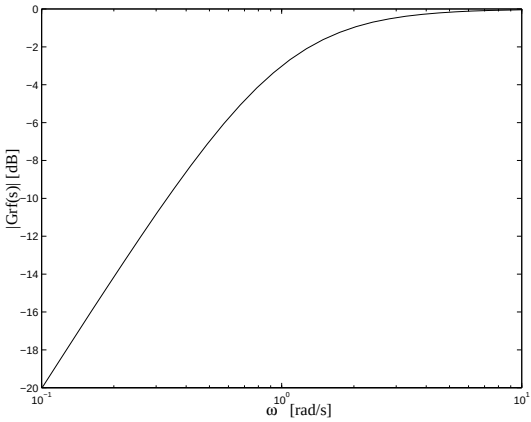


Figure 6.5: Typical transfer function (amplitude plot) from a weakly detectable fault to the residual

In other words, a reference model is attainable if, for some fixed values of the uncertain parameters, there exists a residual generator that produces the fault response specified by the reference model. In the examples used here, the fixed values of the uncertain parameters are set to their nominal values.

The main idea is thus to use a *nominal* design of the residual generator to shape the reference model when synthesizing the robust residual generator, thus assuring attainable reference models. This is to avoid specifying an unrealistic performance criterion and thereby inflicting unnecessary poor robustness properties on the residual generator.

The formation of the criterion for the robust design is straightforward, given that a nominal residual generator, i.e. a $Q_{nom}(s)$, has been derived that nominally fulfills all demands. The reference model $R(s)$ is then selected as

$$R(s) = Q_{nom}(s) \begin{bmatrix} G_f(s) \\ 0 \end{bmatrix} \tag{6.9}$$

since this is the nominal fault to residual transfer function, compare with Eq. (6.3).

Of course, if no design based on a nominal model is available that meets the requirements of the application, then no feasible design with an uncertain model is available either.

6.5.1 Nominal design

The residual generator design problem based on nominal linear models can be solved by any of a number of methods. The design method used here to shape the reference model, is the algorithm from Chapter 3. The reason for this is that

the design freedom available in the nominal design, and therefore in the design of the reference model, is clear, explicit and condensed into the two polynomial matrices $\phi(s)$ and $c(s)$ from (3.6) and (3.7).

Remark

It is possible that, even with nominal models, no disturbance decoupling residual generator can be achieved that conforms to the desired residual structure, i.e. dim $\mathcal{N}_L(M(s)) = 0$. Then a residual generator can be synthesized where disturbances and faults (according to the residual structure) are approximately decoupled. This is often stated as an optimization problem and solved in different ways by many different methods, e.g. by $\mathcal{H}_\infty$ methods (Frank and Ding, 1994; Qiu and Gertler, 1994), singular value truncation (Lou et al., 1986) and other methods (Wünnenberg, 1990). If the resulting residual generator, which doesn't perfectly decouple disturbances, still complies with the requirements on the diagnosis system, the nominal residual generator can be used to form a reference model according to (6.9) and robust synthesis following this chapter can be performed.

Example of reference model design

Below follows an example on how to, given a system description, form the reference model.

Example 6.4

Assume a 1-input, 2-output, second order process with sensor and actuator faults:

$$y = \begin{bmatrix} \frac{\beta}{s+\alpha} \\ \frac{\beta}{s(s+\alpha)} \end{bmatrix} (u + f_3) + \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = G(s)u + [I \; G(s)] \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = G(s)u + L(s)f$$

A (minimal) polynomial basis for $\mathcal{N}_L(M(s))$ can, in this no-disturbance case, be obtained by a row-reduced and irreducible left MFD of $G_u(s) = D^{-1}(s)N(s)$ according to Theorem 3.2. It is straightforward to verify that

$$G(s) = \begin{bmatrix} \frac{\beta}{s+\alpha} \\ \frac{\beta}{s(s+\alpha)} \end{bmatrix} = \begin{bmatrix} s+\alpha & 0 \\ -1 & s \end{bmatrix}^{-1} \begin{bmatrix} \beta \\ 0 \end{bmatrix} = D(s)^{-1}N(s)$$

Then it holds that

$$N_M(s) = [D(s) \; -N(s)] = \begin{bmatrix} s+\alpha & 0 & -\beta \\ -1 & s & 0 \end{bmatrix}$$

Now it is possible to form a matrix $K(s)$ that characterizes all attainable fault to residual transfer functions, i.e. it characterizes all possible residual structures. The matrix $K(s)$ becomes

$$K(s) = N_M(s) \begin{bmatrix} L(s) \\ 0 \end{bmatrix} = \begin{bmatrix} s+\alpha & 0 & \beta \\ -1 & s & 0 \end{bmatrix}$$

Here $K(s)$ has full row-rank and it is therefore clear that residuals corresponding to rows in the residual structure with maximum 1 zero is possible, i.e. it is possible to perfectly decouple maximum 1 fault. If a residual with structure [1 0 1] is desired, then $Q_{nom}(s)$ and $R(s)$ become e.g.

$$Q_{nom}(s) = \frac{\gamma}{s+\gamma}[s+\alpha \quad 0 \quad -\beta] \tag{6.10}$$

$$R(s) = Q_{nom}(s)\begin{bmatrix} L(s) \\ 0 \end{bmatrix} = \frac{\gamma}{s+\gamma}[s+\alpha \quad 0 \quad \beta] \tag{6.11}$$

This is achieved by choosing a $\phi(s) = [1\ 0]$ that selects the first row in $N_M(s)$ and adding arbitrary first order dynamics $D_F(s) = \frac{1}{\gamma}s+1$, i.e. γ is free to shape the fault response in the residual which specifies desired response time.

6.5.2 Discussion of design choices

When choosing $Q_{nom}(s)$, and thereby the reference model $R(s)$, two degrees of freedom have to be chosen by the designer (here it is assumed that in the first-step nominal design, perfectly decoupling residual generators exist):

- The numerator may need to be chosen, especially since the space of decoupling numerators can have dimensionality greater than one, which means that there are several numerators to choose from.
- The denominator need to be chosen. The dynamics (poles) of the residual generator is completely free, conditioned that the degree is greater or equal to the numerator degree.

These choices are then held fix during optimization. Instead of specifying the poles individually, they could of course be obtained almost automatically using a band-width requirement together with a Butterworth structure (Åström and Wittenmark, 1984).

It might be argued that by fixating these two degrees of freedom limits the freedom in the optimization to achieve robustness against parametric variations and that these two variables should be optimized over. However, any fixed set of uncertain parameters Δ would only influence the numerator of the residual generator. This is clear from Chapter 3, since when fixating Δ the problem is a nominal problem as in Chapter 3 where the poles, i.e. matrix $D_F(s)$ was completely free to choose, except for degree constraints. Therefore, it is believed that fixating the poles does not severely influence the robustness properties of the optimal residual generator. If $\dim \mathcal{N}_L(M(s)) > 1$, different base vectors (numerators) might generate different robustness properties of the resulting residual generator. Therefore is it desired to optimize over all linear combinations $\phi(s)$ with some constraint on $\phi(s)$ such that fault sensitivity is kept. If no such constraint is enforced, $\phi(s) = 0$ ($R(s) = 0$) and $Q(s) = 0$

would be optimal, which obviously is a useless design. One possible heuristic way, of course with no convergence guarantees, of performing this optimization is to iteratively solve for $Q(s)$ and $\phi(s)$, much like the use of D-K iterations when performing μ-synthesis. The search for $\phi(s)$ could be constrained to e.g. all constant vectors with $|\phi| = 1$.

6.6 Summary of design method

An advantage with the problem formulation used here is that it is possible to incorporate desired fault to residual performance in the optimization criterion, i.e. the reference model. This is not included in any other optimization criterion used for synthesizing robust residual generators.

Recall the optimization criterion

$$J = \sup_{v \in \mathcal{L}_2} \frac{\|r_0 - r\|_2}{\|v\|_2} = \|T_{zv}(s)\|_\infty = \\ = \| \begin{bmatrix} -G_{ru}^{\Delta}(s) & [R(s) - G_{rf}^{\Delta}(s)] & -G_{rd}^{\Delta}(s) \end{bmatrix} \|_\infty$$

where $G_{ru}^{\Delta}(s)$ is the transfer function from u to r, $G_{rf}^{\Delta}(s)$ is the transfer function from f to r, and $G_{rd}^{\Delta}(s)$ is the transfer function from d to r. These are given directly by Eq. (6.3) and they are all dependent on the residual generator $Q(s)$. The optimization goal is then to find the $Q(s)$ that minimizes J under bounded parametric uncertainties Δ. The first and third components of J then decouples u and d and the second shapes the residual response of the faults.

The synthesis procedure is as follows

1. Perform a nominal design $Q_{nom}(s)$ as in Chapter 3.
2. Select the reference model as $R(s) = Q_{nom}(s) \begin{bmatrix} G_f(s) \\ 0 \end{bmatrix}$
3. Introduce weighting matrices to focus on important frequency ranges
4. Optimize (minimize) J under structured uncertainties Δ with μ-synthesis.
5. Apply model reduction techniques on the resulting residual generator and re-evaluate robust performance via μ-analysis.

The main designer interaction is in step 1 where the nominal design, and thereby the reference model, is selected and step 3 where knowledge of the process or demands on the diagnosis system is used to shape the optimization criteria. Design choices during step 1 were discussed in Section 6.5.2.

A special case of Eq. (6.4) is treated in e.g. (Niemann and Stoustrup, 1997), where integrated residual generator and controller design is presented. There, the residual generator design is performed by letting $r_0(t) = f(t)$ in (6.4), i.e. strive for a diagonal residual structure with unit gain from fault i to residual i,

the optimization criteria becomes the same. This may be a good design objective if it is possible. However, as shown previously in this chapter, this is not always possible and might lead to a poor design. A simple, generic, condition for nominal models on the existence of a residual generator with diagonal influence structure is (assuming detectability of faults)

$$m \geq \text{rank } [G_d(s)\ G_f(s)]$$

where m is the number of measurements.

6.7 Illustrative dynamic example

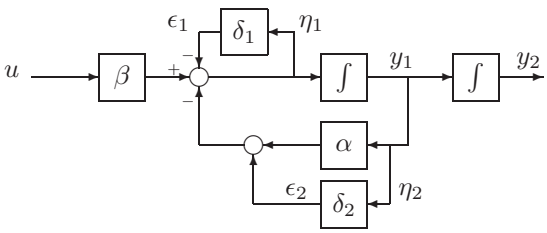
In this section, the concepts introduced in this chapter are applied to a small dynamic example with parametric uncertainties. Three designs are made, one nominal, one with an ad-hoc reference model and one design with a reference model based on the nominal design. Robustness properties of the three designs are then compared.

6.7.1 Model

Reconsider Example 6.4, inspired from a second order DC-servo model. Assume uncertainties in moment of inertia, modeled by δ_1, and in the viscous friction, modeled by δ_2. The model can then be written on a state-space form as

$$\begin{aligned}(1+\delta_1)\dot{x}_1 &= -(\alpha+\delta_2)x_1 + \beta(u+f_3)\\ \dot{x}_2 &= x_1\\ y_1 &= x_1 + f_1\\ y_2 &= x_2 + f_2\end{aligned}$$

Block representation of the fault-free system where the input and output signal to/from δ_i is denoted η_i and ϵ_i respectively:

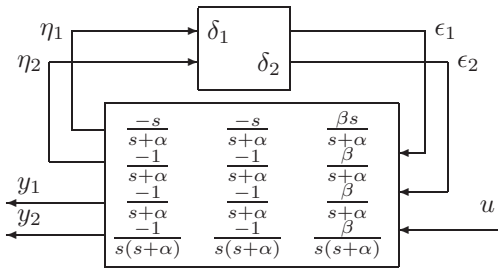


Straightforward calculations give the transfer-functions from ϵ_i and u to η_i and

y.

$$\eta_1 = \frac{s}{s+\alpha}(\beta u - \epsilon_1 - \epsilon_2)$$
$$\eta_2 = \frac{1}{s+\alpha}(\beta u - \epsilon_1 - \epsilon_2)$$
$$y_1 = \eta_2 = \frac{1}{s+\alpha}(\beta u - \epsilon_1 - \epsilon_2)$$
$$y_2 = \frac{1}{s}y_1 = \frac{1}{s(s+\alpha)}(\beta u - \epsilon_1 - \epsilon_2)$$

An upper LFT formulation of the uncertain model (without faults) is then



6.7.2 Residual generator specifications

Suppose a residual generator is to be synthesized to conform to a [1 0 1] structure, i.e. decouple sensor fault 2 while keeping sensitivity to the other two faults. The parameters are $\alpha = 1$ and $\beta = 4$. Three different designs are considered:

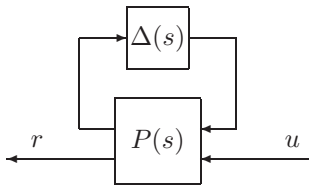
- The nominal design is made via polynomial methods as in Example 6.4. The nominal residual generator $Q_{nom}(s)$ was given by (6.10).
- The robust residual generator with a non-attainable $R(s) = [1\ 0\ 1]$ is synthesized using the method in Section 6.6. The resulting residual generator is denoted $Q_{na}(s)$. This choice of $R(s)$ is related to the first naive attempt in Example 6.3.
- A robust design with an attainable $R(s)$, based on the nominal design $Q_{nom}(s)$, is performed using the method in Section 6.6 and is denoted $Q_{att}(s)$.

6.7.3 Robustness comparison

It is hard to make a fair comparison of robustness because of the many different definitions of robust residual generators. Here, the designs have been normed to have equal *static* fault-to-residual gains (the 2-norm of $G_{rf}(0)$ is equal for all designs). Then, the robustness comparison measure used is how large $G_{ru}(s)$

gets under bounded parametric uncertainties in a worst-case situation. This is the same robustness evaluation criterion as in the static Example 6.3 and corresponds to the operation of the residual generators in a fault-free situation.

Since, compared to Example 6.3, this is a dynamic example, we need some theory to perform the robustness comparison. What is needed is a theorem regarding robust performance from (Zhou et al., 1995) where the worst-case size of $G_{ru}^{\Delta}(s)$ can be computed. Consider the following model setup, where the LFT describes the transfer function from u to r:



Then, it holds that

Theorem 6.3 (Robust Performance). *Let $\beta > 0$. For all $\|\Delta(s)\|_\infty < \frac{1}{\beta}$, the loop shown above is well-posed, internally stable and $\|\mathcal{F}_u(P,\Delta)\|_\infty \leq \beta$ if and only if*

$$\sup_{\omega\in\mathbb{R}} \mu_{\Delta_P}(P(j\omega)) \leq \beta$$

where $\mu_{\Delta_P}(P(j\omega))$ is the structured singular value and

$$\Delta_P = \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_f \end{bmatrix} : \Delta \in \mathbf{\Delta}, \Delta_f \in \mathbb{C}^{q_2\times q_2} \right\}$$

So the smaller β, i.e. peak-value of μ, the smaller is the worst-case value of $\|G_{ru}(s)\|_\infty$ and thereby a more robust fault-free behavior of the residual generator is achieved.

Figure 6.6 shows the μ-plot, $\mu_{\Delta_P}(G_{ru}(j\omega))$ for the three designs. The interpretation of the plot is, according to the theorem above, when $\mu_{\Delta_P}(G_{ru}(j\omega)) < \beta$ then for all $\|\Delta\|_\infty < 1/\beta$ it holds that $\|G_{ru}^{\Delta}(s)\| < \beta$. The plot gives that $Q_{nom}(s)$,$Q_{na}(s)$, and $Q_{att}(s)$ gives $\beta = 0.6, 0.55, 0.5$ respectively. Thus, $Q_{na}(s)$ suffers from up to 15% unnecessary loss of robustness compared to $Q_{att}(s)$. In fact, $Q_{na}(s)$ is even worse than the nominal design for large frequencies. This means that a robust design not considering a proper reference model can in fact be less robust than a nominal design.

This example, and the example in Section 6.4, show that a significant increase in robustness was achieved by robust design with a reference model chosen via a nominal design.

6.8 Robust fault detection

The performance index J has up to now only been used as an optimization index, used to synthesize the residual generator. However, the optimal value of

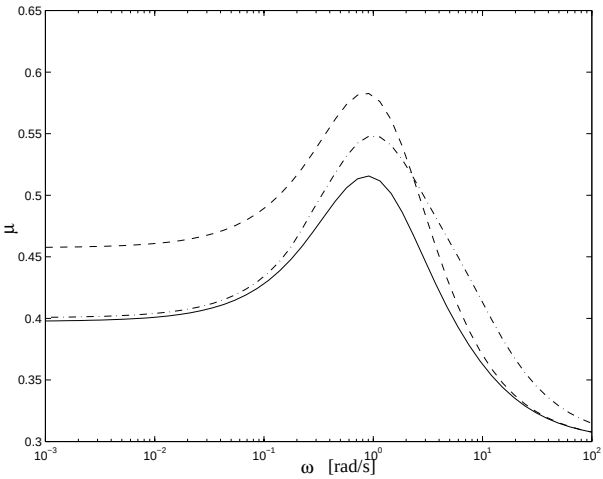


Figure 6.6: Plot of $\mu_{\Delta_P}(G_{ru}(j\omega))$ to evaluate robustness properties. The dashed, dash-dotted, and the solid lines corresponds to $Q_{nom}(s)$, $Q_{na}(s)$ and $Q_{att}(s)$ respectively.

the optimization function can also be used to evaluate absolute performance. Below follows a theorem presenting sufficient conditions on J such that robust fault detection is possible. This means that it is possible to find a threshold which (some norm of) the residual always exceeds when a fault, of a certain size, is present and never exceeds when no fault is present.

Theorem 6.4 (Robust Fault Detection). *Let the problem be normalized so that*

$$\begin{aligned} \|f_i\|_2 &\geq 1 \quad i = 1, \ldots, q \quad \text{if } f_i \neq 0 \\ \|d\|_2 &\leq 1 \\ \|u\|_2 &\leq 1 \end{aligned}$$

and denote $\kappa_f = \min_\omega \underline{\sigma}(R(j\omega))$ *and* $\beta = \sup_\omega \mu_{\Delta_p}(T_{zv}(j\omega))$.

Then, if

$$\beta < \frac{\kappa_f}{\sqrt{2}+\sqrt{3}}$$

there exists a $J_{th} > 0$ *such that*

$$\begin{aligned} \|r\|_2 &< J_{th} \quad \text{if } f = 0 \\ \|r\|_2 &> J_{th} \quad \text{if } \|f_i\| \geq 1 \end{aligned}$$

for all $\|\Delta\|_\infty < 1/\beta$.

Proof. By definition:

$$\forall v \in \mathcal{L}_2 : \frac{\|r_0 - r\|_2}{\|v\|_2} \leq \beta \tag{6.12}$$

In the *fault-free* case $r_0 = 0$ and $\|v\|_2 = \sqrt{\|d\|_2^2 + \|u\|_2^2}$. With (6.12), an upper bound on $\|r\|_2$ can be derived as:

$$\|r\|_2 \leq \beta\sqrt{\|d\|_2^2 + \|u\|_2^2} \leq \beta\sqrt{2} \tag{6.13}$$

thus $\|r\|_2 \leq \beta\sqrt{2}$ in the fault-free case.

With fault(s), $r_0 = R(s)f$ and $v = \begin{bmatrix} u(t) \\ f(t) \\ d(t) \end{bmatrix}$. With (6.12) a lower bound on $\|r\|_2$ can be derived as:

$$\begin{aligned} \|r\|_2 \geq \|r_0\|_2 - \|r_0 - r\|_2 \geq \|R(s)f\|_2 - \beta\|v\|_2 \geq \\ \geq \kappa_f\|f\|_2 - \beta\sqrt{\|f\|_2^2 + 2} = J_1 \end{aligned} \tag{6.14}$$

If $\min_{\|f\|_2 \geq 1} J_1 > \beta\sqrt{2}$ there exists a J_{th}, $\beta\sqrt{2} < J_{th} < \min_{\|f\|_2 > 1} J_1$ such that

$$\begin{aligned} \|r\|_2 &< J_{th} \quad \text{if } f = 0 \\ \|r\|_2 &> J_{th} \quad \text{if } \|f\| \geq 1 \end{aligned}$$

is fulfilled. From (6.14) it follows that

$$\min_{\|f\|_2 \geq 1} J_1 = \begin{cases} \kappa_f - \beta\sqrt{3} & \text{if} \quad \beta \leq \kappa_f \\ -\infty & \text{if} \quad \beta > \kappa_f \end{cases} \tag{6.15}$$

The condition on β,

$$\beta < \frac{\kappa_f}{\sqrt{2} + \sqrt{3}} \tag{6.16}$$

ensures that $\beta < \kappa_f$. Equation 6.16 is equivalent with

$$\kappa_f - \beta\sqrt{3} > \beta\sqrt{2}$$

from which it follows, by inspection of (6.13) and (6.15), that

$$\min_{\|f\|_2 \geq 1} J_1 > \beta\sqrt{2}.$$

■

Example 6.5

Consider a design where $R(s) = I$, i.e. r is an estimation of the fault-vector. Then any fault, of size larger than 1, can be robustly detected if

$$\beta < \frac{1}{\sqrt{2} + \sqrt{3}}$$

6.9 Conclusions

A theory for robust residual design has been developed where the key element is the use of a reference model. The reference model represents desired performance of the synthesized residual generator. It is a condensed formulation including structural requirements, to make the synthesized residual generator fit in a larger diagnosis system based on structured residuals. It also includes performance issues such as fault response in the residual. Without considering structural constraints, it is possible to form unrealistic performance demands and it is shown by examples how this can de-emphasize the robustness parts of the optimization and lead to a design with unnecessary poor robustness properties. A methodology how to select realistic reference models is presented where all design freedom available is explicit and intuitive. The optimization algorithms used to synthesize the residual generator rely on established and efficient methods. The designer of a diagnosis system is thus provided with a tool where it is easy to specify desired behavior without violating structural requirements. Finally it is shown how the optimization procedure, theoretically, provides an absolute measure on the size of disturbances/model uncertainty that is acceptable to be able to robustly detect a fault.

7

Conclusions

The objective when supervising technical processes is to alarm an operator when a fault is detected and also identify one, or possibly a set of components, that may have been the cause of the alarm. Diagnosis is an expansive subject, both in the academic research community and in industry. This is partly due to the fact that nowadays, more applications have more embedded computing power and more available sensors than before. This provide means for, in addition to more advanced control strategies, advanced diagnosis algorithms and on-line supervision by the process itself.

A fundamental part of many model-based diagnosis algorithms are so called *residuals*. A residual is a signal that reacts to a carefully chosen subset of the considered faults and by generating a suitable set of such residuals, fault detection and isolation can be achieved. Design of such residual generators has been the topic of this work.

A common thread throughout this dissertation is the development of systematic design and analysis methods for residual generators based on a number of different model classes. In particular, exploration of available design freedom and simple, intuitive, parameterization of that freedom is pursued. In addition, it is considered important that there exist readily available computer tools for all design algorithms where as little as possible diagnosis specific code need to be developed. Of course, the numerical performance of the algorithms is important to be able to cope with large or ill-conditioned design problems and still produce feasible solutions.

The model classes that were considered are:

- deterministic linear systems on state-space, transfer-function, and descrip-

tor form

- non-linear systems described by polynomial differential-algebraic equations
- stochastic linear systems on state-space, transfer-function, and descriptor form
- linear systems on state-space and transfer-function form with bounded parametric model uncertainties

The simplest case of these are deterministic linear models on state-space or transfer-function form. It is beneficial to study these fundamental systems to thoroughly investigate the residual generation problem and explore fundamental properties of the design problem. It is shown in Chapter 2 how consistency relations, representing the most local relations in the model, forms a suitable basis for residual generator design. The main reason for this is good robustness properties of the residual generator. Based on this, a design algorithm, the minimal polynomial basis approach, is formed by using theory and algorithms for polynomial matrices. The design algorithm mainly consist of two standard operations on polynomial matrices, computation of a basis for the left null-space of a polynomial matrix and realization of an MFD on state-space form. All design freedom is contained in the row-vector $\phi(s)$ from (3.6) and the scalar polynomial $c(s)$ in (3.7). A consequence of the formulation of the design algorithm is that the extension of the algorithm to also cover descriptor systems is immediate. The numerical performance is shown to be good on relatively large state-space and descriptor systems.

To further help guide the selection of the available design freedom, additional modeling is necessary. Two natural choices of model classes to consider are then models with parametric uncertainties or models with noise descriptions of uncertainty. When considering linear systems with uncertain parameters, a design algorithm is formulated based on an optimization criterion and standard $\mathcal{H}_\infty$-filtering techniques. A fundamental part of the optimization criterion is the *reference model* that specifies desired performance of the residual generator. A key observation is that a natural, and often used, reference model can impose unnecessary poor robustness properties of the residual generator because the criterion violates structural properties of the model. Based on this observation, a methodology to select a well formed reference model is developed in Chapter 6 that is based on the minimal polynomial basis approach from Chapter 3.

For the stochastic linear systems systems, design of innovation filters is studied and also an extension to a more general class of residual generators, whitening residual generators. Also here, the nominal design procedure forms a foundation for the design steps and only one additional operation, spectral factorization, is needed for the design of residual generators. Because of this, also stochastic descriptor systems can be handled without any modification of the algorithm. In a series of non-trivial examples, both state-space and descriptor,

it is shown how the design works and what design freedom that is still available. To show the merits of the extended problem formulation, a design example is included where an innovation filter does not exist while a whitening residual generator exist that meets all design specifications.

The last model class considered is non-linear models. To achieve the rather ambitious goals, the class of non-linear models is restricted to models described by polynomial differential-algebraic equations. For this, relatively large, class of non-linearities, powerful computational tools exists and a design algorithm with great similarities with the linear design algorithm is developed. The key step is here, instead of computing a basis for the left null-space of a polynomial matrix, to compute a Gröbner basis for a specified elimination ideal. A main problem with the approach is the inherent algorithmic complexity which makes it difficult to handle anything but small models. Structural analysis of the model equations, which can be performed for huge models, is then shown to provide a way to manage this complexity. To study this, it is shown how a relatively small 4 state model poses an intractable problem on a standard personal computer. However, after a structural analysis is used to reduce the problem, a computer algebra package provide a solution in only 0.02 seconds.

In conclusion, for the four classes of models studied, new methods have been developed. The methods fulfills requirements generation of all possible solutions, availability of computational tools, and numerical soundness. The methods also provide the diagnosis system designer with a set of tools with well specified and intuitive design freedom.

Notation

$\mathbb{R}$,$\mathbb{C}$	The field of real/complex numbers.
$\mathcal{H}_\infty$	The Hardy space that consists of all complex-valued functions that are analytical and bounded in the open left half-plane.
$\mathcal{RH}_\infty$	The real rational subspace of $\mathcal{H}_\infty$ is denoted $\mathcal{RH}_\infty$ which then consists of all proper and real rational stable transfer matrices.
rank $A(s)$	Denotes normal rank of $A(s) \in \mathbb{R}^{m\times n}[s]$.
$\mathbb{R}^{m\times n}[s]$	$M(s) \in \mathbb{R}^{m\times n}[s]$ if and only if $M(s)$ is a matrix of dimension $m \times n$ with polynomial elements in variable s with real coefficients.
$\mathbb{R}^{m\times n}(s)$	$M(s) \in \mathbb{R}^{m\times n}(s)$ if and only if $M(s)$ is a matrix of dimension $m \times n$ with rational elements in variable s with real coefficients.
$\mathcal{N}_L(M(s))$	The rational left null-space of a matrix $M(s) \in \mathbb{R}^{m\times n}[s]$ or $M(s) \in \mathbb{R}^{m\times n}(s)$.
$N_M(s)$	The rows of $N_M(s)$ forms a minimal polynomial basis for $N_{M_s}(s)$.
u, y, f, d	Signals representing control signals, measurements, faults, and disturbances respectively. Signals are denoted in lower-case, both in time-domain and frequency domain. When it is clear from context if the signals is specified in time or frequency domain, argument t or s dropped for notational convenience.
$G_{uv}(s)$	The transfer function from v to u.
$\left[\begin{array}{c\|c} A & B \\ \hline C & D \end{array}\right]$	The transfer function $C(sI - A)^{-1}B + D$

References

H. Aling and J.M. Schumacher. A nine-fold canonical decomposition for linear systems. *Int. J. of Control*, 39(4):779–805, 1984.

K. Åström and B. Wittenmark. *Computer Controlled Systems - Theory and Design*. Prentice Hall, 1984.

K. Åström and B. Wittenmark. *Adaptive Control*. Prentice Hall, 1989.

G.J. Balas, J.C. Doyle, K. Glover, A. Packard, and R. Smith. *μ-Analysis and Synthesis Toolbox*. Math Works Inc., 1993.

M. Basseville, A. Benveniste, G. Moustakides, and A. Rougée. Detection and diagnosis of changes in the eigenstructure of nonstationary multivariable systems. *Automatica*, 23(4):479–489, 1987.

M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice Hall, 1993.

R.V. Beard. *Failure Accommodation in Linear System Through Self Reorganization*. PhD thesis, MIT, 1971.

M. Blanke, R. Izadi-Zamanabadi, S. Bøgh, and C. Lunau. Fault-tolerant control systems - a holistic view. *Control Eng. Practice*, 5(5):693–701, 1997.

W. Boege, R. Gebauer, and H. Kredel. Some examples for solving systems of algebraic equations by calculating groebner bases. *Journal of Symbolic Computing*, 2(1):83–98, 1986.

F.M. Callier. On polynomial matrix spectral factorization by symmetric factor extraction. *IEEE Trans. Automatic Control*, 30(5):453–464, 1985.

C.T. Chen. *Linear System Theory and Design*. Holt, Rinehart and Winston, New York, 1984.

J. Chen and R.J. Patton. A re-examination of fault detectability and isolability in linear dynamic systems. In *Proc. IFAC Safeprocess'1994*, pages 567–573, Espoo, Finland, 1994.

J. Chen and R.J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer, 1999.

E.Y. Chow and A.S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. on Automatic Control*, 29(7):603–614, 1984.

D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag, 1991.

A.J.van der Schaft. On realization of nonlinear systems described by higher order differential equations. *Math. Systems Theory*, 19:239–275, 1987.

S.X. Ding, E.L. Ding, and T. Jeinsch. A numerical approach to analysis and design of FDI systems. IEEE Conf. on Decision and Control, 1998.

S.X. Ding, E.L. Ding, and T. Jeinsch. An approach to analysis and design of observer and parity relation based FDI systems. volume P of *Proc. of IFAC 1999 World Congress*, pages 37–42, Beijing, P.R. China, 1999a.

X. Ding and P.M. Frank. Fault detection via factorization approach. *Systems & control letters*, 14(5):431–436, 1990.

X. Ding, L. Guo, and T. Jeinsch. A characterization of parity space and its application to robust fault detection. *IEEE Trans. on Automatic Control*, 44 (2):337–342, 1999b.

A. Edelmayer, J. Bokor, and L. Keviczky. An H_∞ filtering approach to robust detection of failures in dynamic systems. In *Proceedings Decision & Control*, pages 3037–3039, 1994.

J. Eich and R. Oehler. On the application of the generalized structured singular value to robust fdi system design. In *Proc. IFAC Safeprocess'1997*, pages 885–890, Hull, UK, 1997.

G.D. Forney. Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM J. Control*, 13(3):493–520, May 1975.

K. Forsman. *Constructive Commutative Algebra in Nonlinear Control Theory*. PhD thesis, Linköping University, 1991.

P.M. Frank and X. Ding. Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. *Automatica*, 30(5):789–804, 1994.

P.M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results. *Automatica*, 26 (3):459–474, 1990.

E. Frisk and L. Nielsen. Robust residual generation for diagnosis including a reference model for residual behavior. In *Proc. IFAC World Congress*, volume P, pages 55–60, Beijing, China, 1999.

E. Frisk and M. Nyberg. Using minimal polynomial bases for fault diagnosis. In *Proc. European Control Conference*, 1999.

E. Frisk and M. Nyberg. A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems. *Automatica*, 37(9):1417–1424, 2001.

E. Frisk and M. Nyberg. Residual generation for fault diagnosis of systems described by general linear differential algebraic equations. In *submitted to Proc. 15th IFAC World Congress*, Barcelona, Spain, 2002.

E. Frisk. Order of residual generators – bounds and algorithms. In *Proc. IFAC Safeprocess'2000*, pages 599–604, Budapest, Hungary, 2000a.

E. Frisk. Residual generatior design for non-linear, polynomial systems – a gröbner basis approach. In *Proc. IFAC Safeprocess'2000*, pages 979–984, Budapest, Hungary, 2000b.

E. Frisk. Residual generatior in linear stochastic systems - a polynomial approach. In *Proc. IEEE Conference on Decision and Control*, Orlando, USA, 2001.

J. Gertler, X. Fang, and Q. Luo. Detection and diagnosis of plant failures: the orthogonal parity equation approach. *Control and Dynamic Systems*, 37: 159–216, 1990.

J. Gertler and D. Singer. A new structural framework for parity equation-based failure detecation and isolation. *Automatica*, 26(2):381–388, 1990.

J. Gertler. Analytical redundancy methods in fault detection and isolation; survey and synthesis. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 9–21, Baden-Baden, Germany, 1991.

J. Gertler. *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker, 1998.

M. Green, K. Glover, D.J.N. Limebeer, and J. Doyle. A J-spectral factorization approach to $\mathcal{H}_\infty$-control. *SIAM J. Contr. Opt.*, 28:1350–1371, 1990.

C. Guernez, J.Ph. Casar, and M. Staroswiecki. Extension of parity space to non-linear polynomial dynamic systems. In *Proc. IFAC Safeprocess 1997*, pages 861–866, Hull, UK, 1997.

W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan Kaufmann Publishers, 1992. ISBN 1-55860-249-6.

A. Helmersson. *Methods for Robust Gain Scheduling*. PhD thesis, Linköping University, 1995.

M. Hou and P.C. Müller. Tracking control for a class of descriptor systems. In *Proc. 13th IFAC World Congress*, pages 109–114, San Fransisco, USA, 1996.

M. Hou. *Fault detection and isolation for descriptor systems*, chapter 5. In , Patton et al. (2000), 2000.

R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing - tutorial paper. *Automatica*, 29(4):815–835, 1993.

A. Isidori. *Nonlinear control systems*. Springer, 3rd edition, 1995.

V.A. Jakubovič. Factorization of symmetric matrix polynomials. *Dokl. Akad. Nauk SSSR*, 194(3), 1970.

J. Ježek and V. Kučera. Efficient algorithm for matrix spectral factorization. *Automatica*, 21(6):663–669, 1985.

M. Jirstrand. Nonlinear control design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):137–152, August 1997.

M. Jirstrand. *Constructive Methods for Inequality Constraints in Control*. PhD thesis, Linköping University, 1998.

B. Kågström. RGSVD - an algorithm for computing the kronecker structure and reducing subspaces of singular $A-\lambda B$ pencils. *SIAM J. Sci. Stat. Comput*, 7(1):185–211, 1986.

T. Kailath. *Linear Systems*. Prentice-Hall, 1980. ISBN 0-13-536961-4.

R. Karcanias and G. Kalogeropoulos. Right, left characteristic sequences and column, row minimal indices of a singular pencil. *Int. J. of Control*, 47(4): 937–946, 1988.

F. Kratz, S. Bousghiri, and W. Nuninger. A finite memory observer structure of continuous descriptor systems. In *Proc. American Control Conference*, pages 3900–3904, Seattle, USA, 1995.

V. Krishnaswami, G.C. Luh, and G. Rizzoni. Fault detection in IC engines using nonlinear parity equations. In *Proceedings of the American Control Conference*, pages 2001–2005, Baltimore, Maryland, 1994.

V. Krishnaswami and G. Rizzoni. Nonlinear parity equation residual generation for fault detection and isolation. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 305–310, Espoo, Finland, 1994.

M. Krysander and M. Nyberg. Diagnosis of large scale differential-algebraic systems utilizing structural properties. In *Submitted to IFAC World Congress*, Barcelona, Spain, 2002.

H. Kwakernaak and M. Šebek. Polynomial J-spectral factorization. *IEEE Tran. on Automatic Control*, 39(2):315–328, 1994.

H. Kwakernaak. A descriptor algorithm for the spectral factorization of polynomial matrices. In *Proc. Robust Control Design*, Prague, Czech Republic, 2000a. IFAC.

H. Kwakernaak. $\mathcal{H}_2$-optimization – theory and applications to robust control design. In *Proc. Robust Control Design*, Prague, Czech Republic, 2000b. IFAC.

P. Lancaster and M. Tismenetsky. *The theory of matrices*. Academic Press, second ed. edition, 1985.

M. Larsson. *Behavioral and Structural Model Based Approaches to Discrete Diagnosis*. PhD thesis, Linköping University, Sweden, 1999.

P. Lindskog. *Methods, Algorithms and Tools for System Identification Based on Prior Knowledge*. PhD thesis, Linköping University, 1996.

L. Ljung and T. Glad. On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265–276, 1994.

X.C. Lou, A.S. Willsky, and G.C. Verghese. Optimally robust redundancy relations for failure detection in uncertain systems. *Automatica*, 22(3):333–344, 1986.

J.M. Maciejowski. *Multivariable Feedback Design*. Addison Wesley, 1989.

J.F. Magni and P. Mouyon. On residual generation by observer and parity space approaches. *IEEE Trans. on Automatic Control*, 39(2):441–447, 1994.

R. Mangoubi, B.D. Appleby, and J. Farrell. Robust estimation in fault detection. In *Proceedings Decision & Control*, pages 2317–2322, 1992.

R. Mangoubi, B.D. Appleby, G.C. Verghese, and W.E. VanderVelde. A robust failure detection and isolation algorithm. In *Proceedings Decision & Control*, pages 2377–2382, 1995.

R. Mangoubi, B.D. Appleby, and G.C. Verghese. Stochastic interpretation of H_∞ and robust estimation. In *Proceedings Decision & Control*, pages 3943–3948, 1994.

M.A. Massoumnia, G.C. Verghese, and A.S. Willsky. Failure detection and identification. *IEEE Trans. on Automatic Control*, AC-34(3):316–321, 1989.

R.K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7:637–640, 1971.

L.A. Mironovskii. Functional diagnosis of linear dynamic systems. *Automation and Remote Control*, pages 1198–1205, 1980.

P. Misra, P. Dooren, and A. Varga. Computation of structural invariants of generalized state-space systems. *Automatica*, 30(12):1921–1936, 1994.

P.C. Müller and M. Hou. On the observer design for descriptor systems. *IEEE Trans. on Automatic Control*, 38(11):1666–1671, 1993.

P.C. Müller. Descriptor systems: pros and cons of system modelling by differential-algebraic equations. *Mathematics and computers in simulation*, 53:273–279, 2000.

H. Niemann and J. Stoustrup. Integration of control and fault detection: nominal and robust design. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 341–346, Hull, UK, 1997.

R. Nikoukhah. Innovations generation in the presence of unknown inputs: Application to robust failure detection. *Automatica*, 30(12):1851–1867, 1994.

M. Nyberg and L. Nielsen. Parity functions as universal residual generators and tool for fault detectability analysis. In *IEEE Conf. on Decision and Control*, 1997.

M. Nyberg and L. Nielsen. A universal Chow-Willsky scheme and detectability criteria. *IEEE Trans. on Automatic Control*, 45(1):152–156, 2000.

M. Nyberg. Model based diagnosis of both sensor-faults and leakage in the air-intake system of an si-engine. In *Proc. SAE World Congress*, number 1999-01-0860 in SAE, Detroit, USA, 1999a.

M. Nyberg. *Model Based Fault Diagnosis: Methods, Theory, and Automotive Engine Applications.* PhD thesis, Linköping University, May 1999b.

M. Nyberg. Criterions for detectability and strong detectability of faults in linear systems. In *Proc. IFAC Safeprocess'2000*, 2000.

R.J. Patton, P.M. Frank, and R.N. Clark, editors. *Issues of Fault Diagnosis for Dynamic Systems.* Springer, 2000.

R.J. Patton and M. Hou. Design of fault detection and isolation observers: A matrix pencil approach. *Automatica*, 34(9), 1998.

R.J. Patton. Robust model-based fault diagnosis:the state of the art. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 1–24, Espoo, Finland, 1994.

C. De Persis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Trans. on Automatic Control*, 46(6):853–865, 2001.

The Polynomial Toolbox 2.5 for Matlab 5. Polyx, Czech Republic. URL: http://www.polyx.com, 2001.

Z. Qiu and J. Gertler. Robust FDI systems and H_∞-optimization. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 247–252, Espoo, Finland, 1994.

J. Ritt. *Differential Algebra*. Dover Publications, 1950.

N. Sadegh. Minimal realization of nonlinear systems described by input-output difference equations. *IEEE Trans. on Automatic Control*, 46(5):698–710, 2001.

M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D.C. Teneketzis. Diagnosability of discrete event systems. *IEEE Trans. on Automatic Control*, 40(9):1555–1575, 1995.

M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D.C. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Trans. on Control Systems Technology*, 4(2):105–124, 1996.

D. Sauter, F. Rambeaux, and F. Hamelin. Robust fault diagnosis in an H_∞ setting. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 879–884, Hull, UK, 1997.

D.N. Shields. Robust fault detection for descriptor systems. In *Proc. IASTED International Conf., modelling, identification, and control.*, pages 301–304, Anaheim, USA, 1994.

R.F. Sincovec, A.N. Erisman, E.L. Yip, and M.A. Epton. Analysis of descriptor systems using numerical algorithms. *IEEE Trans. on Automatic Control*, 26 (1):139–147, 1981.

T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989.

E.D. Sontag and Y. Wang. Input/output equations and realizability. In *Proc. Int. Symposium MTNS-89*, volume 1, pages 125–132, 1990.

M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5): 687–699, 2001.

R.C.W. Strijbos. Calculation of right matrix fraction descriptions; an algorithm. In *Proceedings of the 4th IEEE Mediterranean Symposium on New Directions in Control and Automation,*, pages 478–482, Maleme, Krete, Greece, 1996.

M. Šebek. An algorithm for spectral factorization of polynomial matrices with any signature. Technical Report ISSN 0169-2690, Faculty of Applied Mathematics, University of Twente, The Netherlands, 1990.

N. Viswanadham, J.H. Taylor, and E.C. Luce. A frequency-domain approach to failure detection and isolation with application to GE-21 turbine engine control systems. *Control - Theory and advanced technology*, 3(1):45–72, March 1987.

S.H. Wang, P. Dorato, and E.J. Davison. On the computation of minimal indices for linear multivariable systems. *International Journal of Control*, 21 (4):537–543, 1975.

W.M. Wonham. *Linear Multivariable Control: a Geometric Approach*. Springer-Verlag, 2nd edition, 1979.

J. Wünnenberg. *Observer-Based Fault Detection in Dynamic Systems*. PhD thesis, University of Duisburg, 1990.

E.L. Yip and R.F. Sincovec. Solvability, controllability, and observability of continuous descriptor systems. *IEEE Trans. on Automatic Control*, 26(2): 702–707, 1981.

Q. Zhang, M. Basseville, and A. Benveniste. Fault detection and isolation in nonlinear dynamic systems: A combined input-output and local approach. *Automatica*, 34(11):1359–1373, 1998.

A.N. Zhirabok. Parity space approach to fault diagnosis in nonlinear systems. In *European Control Conference*, 1999.

K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1995.

Author Index

Index

S

U

V

W