Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Methods for Residual Generation Using Mixed Causality in Model Based Diagnosis

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan i Linköping
av

Johan Kingstedt, Magnus Johansson

LITH-ISY-EX--08/4882--SE

Linköping 2008

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Methods for Residual Generation Using Mixed Causality in Model Based Diagnosis

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan i Linköping
av

Johan Kingstedt, Magnus Johansson

LITH-ISY-EX--08/4882--SE

Handledare: **Carl Svärd**
NED, Scania AB

Examinator: **Mattias Nyberg**
ISY, Linköpings Universitet

Linköping, 18 January, 2008

Avdelning, Institution
Division, Department

Division of Vehicular Systems
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Datum
Date

2008-01-18

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ______________

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ ______________

ISBN
—

ISRN
LITH-ISY-EX--08/4882--SE

Serietitel och serienummer
Title of series, numbering

ISSN
—

URL för elektronisk version

http://www.fs.isy.liu.se

Titel
Title

Metoder för Residualgenerering med Mixad Kausalitet i Modellbaserad Diagnos

Methods for Residual Generation Using Mixed Causality in Model Based Diagnosis

Författare
Author

Johan Kingstedt, Magnus Johansson

Sammanfattning
Abstract

Several different air pollutions are produced during combustion in a diesel engine, for example nitric oxides, NO_x, which can be harmful for humans. This has led to stricter emission legislations for heavy duty trucks. The law requires both lower emissions and an On-Board Diagnosis system for all manufactured heavy duty trucks. The OBD system supervises the engine in order to keep the emissions below legislation demands. The OBD system shall detect malfunctions which may lead to increased emissions. To design the OBD system an automatic model based diagnosis approach has been developed at Scania CV AB where residual generators are generated from an engine model.

The main objective of this thesis is to improve the existing methods at Scania CV AB to extract residual generators from a model in order to generate more residual generators. The focus lies on the methods to find possible residual generators given an overdetermined subsystem. This includes methods to estimate derivatives of noisy signals.

A method to use both integral and derivative causality has been developed, called mixed causality. With this method it has been shown that more residual generators can be found when designing a model based diagnosis system, which improves the fault isolation. To use mixed causality, derivatives are estimated with smoothing spline approximation.

Nyckelord
Keywords

model based residual generation, derivative estimation, structural methods

Abstract

Several different air pollutions are produced during combustion in a diesel engine, for example nitric oxides, NO_x, which can be harmful for humans. This has led to stricter emission legislations for heavy duty trucks. The law requires both lower emissions and an On-Board Diagnosis system for all manufactured heavy duty trucks. The OBD system supervises the engine in order to keep the emissions below legislation demands. The OBD system shall detect malfunctions which may lead to increased emissions. To design the OBD system an automatic model based diagnosis approach has been developed at Scania CV AB where residual generators are generated from an engine model.

The main objective of this thesis is to improve the existing methods at Scania CV AB to extract residual generators from a model in order to generate more residual generators. The focus lies on the methods to find possible residual generators given an overdetermined subsystem. This includes methods to estimate derivatives of noisy signals.

A method to use both integral and derivative causality has been developed, called mixed causality. With this method it has been shown that more residual generators can be found when designing a model based diagnosis system, which improves the fault isolation. To use mixed causality, derivatives are estimated with smoothing spline approximation.

Sammanfattning

Vid förbränning av diesel bildas flera olika luftföroreningar, däribland kväveoxider som är skadliga för människor. Detta har lett till hårdare lagkrav gällande avgasutsläpp för tung fordonstrafik. Lagen kräver lägre emissioner men även att lastbilarna skall vara utrustade med ett diagnossystem (OBD). OBD-systemet skall upptäcka fel som kan öka avgaserna. För att designa OBD-systemet har en metod utvecklats på Scania CV AB som utifrån en motormodell automatiskt genererar residualgeneratorer.

Huvudsyftet med detta examensarbete är att förbättra den redan befintliga metoden på Scania CV AB för att hitta residualgeneratorer från en modell. Fokus ligger på att hitta fler residualgeneratorer givet ett överbestämt delsystem. För att göra detta måste derivator skattas från brusiga mätsignaler.

En metod för att använda både deriverande och integrerande kausalitet som kallas mixad kausalitet har tagits fram. Det har visats att fler residualgeneratorer kan genereras om mixad kausalitet används för att designa ett modellbaserat diagnossystem. Detta medför en förbättrad felisolering. För att använda mixad kausalitet skattas derivator med "smoothing spline approximation".

Acknowledgments

We would like to express our gratitude to a number of people:

First of all, we would like to thank our supervisor PhD student Carl Swärd, at Scania CV AB and Linköping University, for his dedicated support and for many good discussions. We would also like to thank our examiner docent Mattias Nyberg, at Scania CV AB and Linköping University, for all discussions and suggestions. The staff and our fellow master thesis workers at NED at Scania CV AB are acknowledge for helping us with our work and for many nice coffee breaks. Finally, we would like to thank our girlfriends, Emma and Therese, for their support.

Johan Kingstedt
Södertälje, December 2007
Magnus Johansson
Södertälje, December 2007

Contents

Part I

Introduction and Background Theory

Chapter 1

Introduction

This master thesis was performed at Scania CV AB in Södertälje, at the department of Diagnosis, NED. The department is responsible for the on board diagnosis system (OBD). Scania CV AB is a manufacturer of heavy duty trucks which are sold worldwide. The trucks has a gross vehicle weight of more than 16 tonnes and around 60 000 trucks was manufactured during 2006, see [18].

1.1 Background

Several different air pollutions are produced during combustion in a diesel engine, for example nitric oxides, NO_x, which can be harmful for humans. This has led to stricter emission legislations for heavy duty trucks. The law requires both lower emissions but also that all heavy duty trucks have an OBD system. The OBD system supervises the engine in order to keep the emissions below legislation demands. The OBD system shall detect malfunctions which may lead to increased emissions.

There are different approaches but one approach to design the OBD system is to use model based diagnosis. The idea with model based diagnosis is to build a model of the process, in this case the vehicle engine, and construct tests from the model. These tests run in a real-time control unit in the truck. The tests are typically based on the output from residual generators. The residual generator consists of a model of the system and the output is the difference between a modeled variable and the same measured variable. If they are not equal a fault has probably occurred.

If the model of the process is complex the residual generators will also become very complex. If the process and the model are changed the residual generators must also be changed. Therefore it is necessary to have reliable methods that can find and construct the residual generators automatically given a model of the process.

Such a tool is developed at Scania. The tool extracts overdetermined subsystems and produce residual generators from them. From all found subsystems there are only possible to create residual generators from less than a twentieth of them.

To increase the ability to detect faults and isolate them it is desirable to construct more residual generators.

1.2 Existing Work

This thesis is a part of a bigger project where much work already has been done. Algorithms to transform a SIMULINK model to analytical equations and extracting overdetermined subsystems from the equation systems has been carried out in [5]. The method to extract overdetermined subsystems has been improved in [19]. Based on the overdetermined subsystems a method to generate model based residual generators has been done in [6]. The overall method has been theoretically compared with a similar method and an approach to solve the instability issues is presents in [4].

1.3 Objectives

The main objective of this thesis is to improve the existing methods at Scania CV AB to extract residual generators from a model in order to generate more residual generators. The focus lies on the methods to find possible residual generators given an overdetermined subsystem. The main objective can be divided in two subparts

- Investigate and compare different methods to estimate derivatives of signals in order to use derivatives for realizing model based residual generators.
- Find a method that combines integration and differentiation to find model based residual generators.

1.4 Outline of the Thesis

Part I: Presents the background theory in control theory, model based diagnosis and structural analysis that are used trough out the thesis to the reader.

Part II: Presents a discussion of two methods used to construct residual generators and a third method that builds on the other two methods.

Part III: Present a method to realize residual generators without estimating derivatives and three different methods to estimate derivatives. A comparison is made between these two methods. An evaluation of the methods in previous part on a Scania engine model is carried out.

1.5 Contributions

Chapter 4: A discussion of how differential equations must be handled in structural analysis in order to use mixed causality.

Chapter 5-6: A discussion of when a correct initial condition is needed and a presentation of when derivative causality have advantages compared to integral causality. Also a discussion of when differential loops can be solved.

Chapter 7: A presentation of how integral and derivative causality can be used together and an algorithm for matching variables.

Chapter 8: A presentation of three different methods to estimate derivatives and a comparison between residual generators based on estimated derivatives and residual generators realized in state-space form.

Chapter 9: An evaluation of the methods described in previous chapters, which shows that the algorithm developed, gives a contribution to finding more model based residual generators.

1.6 Target Group

The target group for this thesis is undergraduate students and graduate engineers who have an interest in model based diagnosis. Knowledge in model based diagnosis, structural analysis, signal processing and control theory gives a better understanding of the thesis.

Chapter 2

Control Theory

The purpose with this chapter is to explain some theories in control theory, see [9]. The theories include system models, stability for a system and observer theory.

2.1 System Models

Systems in state-space form, either in linear or non-linear form are considered in this thesis. System (2.1) is a linear state-space system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.1a}$$
$$y(t) = Cx(t) + Du(t), \tag{2.1b}$$

which is a special case of the non-linear state-space system

$$\dot{x}(t) = f(x(t), u(t)) \tag{2.2a}$$
$$y(t) = h(x(t), u(t)), \tag{2.2b}$$

where $x(t)$ are states, $u(t)$ inputs and $y(t)$ outputs.

Semi-explicit differential algebraic equations are another frequently used system model in this thesis. The state-space model (2.2) is a special case of the non-linear semi-explicit differential algebraic equations

$$\dot{x}_1(t) = f(x_1(t), x_2(t), z(t)) \tag{2.3a}$$
$$0 = g(x_1(t), x_2(t), z(t)), \tag{2.3b}$$

where $x_1(t)$ are differential variables, $x_2(t)$ unknown algebraic variables and $z(t)$ known in and outputs.

To get a more readable thesis, the time index t will in the sequel be omitted when the time is not relevant, system (2.3) then becomes

$$\dot{x}_1 = f(x_1, x_2, z) \tag{2.4a}$$
$$0 = g(x_1, x_2, z). \tag{2.4b}$$

2.2 Stability

There are a number of different stability definitions for systems in state-space form, for example input-output stability, stability for equilibrium points and stability of a solution. The stability of a solution to system (2.2) is connected to how the initial condition, x_0, affect the solution.

Definition 2.1 *A solution x^* to the system of differential equations (2.2a) is stable if there for every ϵ exists a δ such that $|x_0^* - x_0| < \delta$ yield $|x^* - x| < \epsilon$ for every $t > 0$. The solution is unstable if not stable. The solution is asymptotic stable if it is stable and there exist a δ such that $|x_0^* - x_0| < \delta$ yields $|x^* - x| \to 0$ when $t \to \infty$. Where x_0 is the initial condition of the system and x_0^* is the initial condition of the solution.*

Theorem 2.1 provides a result that can be useful when investigating stability of a linear system.

Theorem 2.1 (Stability for linear system) *A linear system (2.1) is asymptotic stable if and only if the eigenvalues λ to the matrix A are in the closed left half plane, that is*

$$\Re\{\lambda(A)\} < 0, \tag{2.5}$$

where $\lambda(A)$ are the eigenvalues to the matrix A.

Proof
See [9]. □

There is no similar useful theorem when investigating stability for non-linear systems. However, there are methods to investigate the stability of the equilibrium points but that can not be used to determine if the system is globally stable or not.

2.3 Observers

Given a state-space model of a system, the states, x, are not usually observed but only the outputs, y. However, with an observer the states x can be reconstructed by using known in and outputs, u and y.

Given a state-space model (2.2), the states x, are observed as

$$\dot{\hat{x}} = f(\hat{x}, u) \tag{2.6}$$

To measure how good the observed states, $\hat{x}$, corresponds to the states, x, the quantity $y - h(\hat{x}, u)$ can be used. This quantity is zero when $x = \hat{x}$ and there is no measurement noise. This quantity can also be used as feedback to make the observed states converge to the correct states. The observer is on the form

$$\dot{\hat{x}} = f(\hat{x}, u) + l(y - h(\hat{x}, u)), \tag{2.7}$$

where l is some observer function. The observer function, l, affects if and how fast the estimation error converge to zero and how sensitive the observer is for measurement noise. For linear systems the observer function, l, is replaced by an observer gain, K, which can be determined in a number of ways. The Kalman filter is used in this thesis to determine the observer gain K. The observer function, l, can be determined in many different ways for non-linear systems. Since no non-linear observers are used in this thesis they are not discussed further.

2.3.1 Kalman Filter

The Kalman filter is a well-known and efficient tool that can be used for observing states of a linear dynamic system given a system model and measurements, see [7]. The system (2.1) with process noise w and measurement noise v is denoted

$$\dot{x} = Ax + Bu + Nw \tag{2.8a}$$

$$y = Cx + Du + v. \tag{2.8b}$$

The noises w and v are assumed to be white noises with variances Q and R respectively. The cross-covariance, S, between w and v is assumed constant. The stationary Kalman filter minimizes the variance of the estimation error, $\tilde{x} = \hat{x} - x$, and is given by

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x} - Du), \tag{2.9}$$

where K is the Kalman gain given by

$$K = (PC^T + NS)S^{-1}, \tag{2.10}$$

and P is the positive semidefinite solution to

$$AP + PA^T - (PC^T + NS)R^{-1}(PC^T + NS)^T + NQN^T = 0. \tag{2.11}$$

Equation (2.11) is called the stationary Riccati equation and several approaches to solve the stationary Riccati equation exist. If w and v are Gaussian there are no better linear or non-linear state-estimator than the Kalman filter, and regardless of the distribution of the noises w and v it does not exist any better linear filter, see [11].

Chapter 3

Diagnosis

The purpose of this chapter is to present the basic diagnosis theory, see [8]. The concept of model based diagnosis is explained and a brief discussion of what quantities that makes a diagnosis system good is presented.

3.1 Introduction to Diagnosis

Diagnosis is to make a statement of a system given observations of the system that shall be diagnosed. That is, from observations and knowledge of the system detect if a fault is present and if so, it is desirable to isolate the fault.

A fault in a system is described as a deviation of the system structure or the system parameters from the nominal situation, see [1].

When monitoring technical systems, faults can be detected in several different ways. The traditional diagnosis method has been to check when certain measured variables go outside a predefined range. If a signal exceeds the limit, there is a fault present. Another method is to have hardware redundancy, which means that there are many sensors measuring the same variable. With two sensors measuring the same variable and one of the sensors diverge from the other, there is a fault in one of the two sensors. The problem is to know in which sensor the fault has occurred. With three sensors measuring the same variable and one sensor diverges from the other two, a fault has most likely occurred in the sensor that diverges. This method is reliable but very expensive.

If only one sensor is used to monitor a variable and at the same time the variable is modeled, it is maybe possible to detect and isolate the fault. This leads to model based diagnosis.

3.2 Model Based Diagnosis

The concept of model based diagnosis is to make a model of the system that shall be monitored and use that information along with information from sensors. The model of the system is

$$\dot{x} = f(x, u, f) \tag{3.1a}$$
$$y = h(x, u, f), \tag{3.1b}$$

where f represents arbitrary faults, for example actuator faults and sensor faults. The relation between the modeled variable and the measured variable is used in a residual generator, F, where the output, R, is zero in the fault-free case and nonzero when a fault that affects the residual generator is present. A residual generator is a system, F, with inputs u and y and output R.

Definition 3.1 *A residual generator is a function $F(u(t), y(t))$, such that*

$$u, y \in \Theta_{NF} \Rightarrow R = F(u(t), y(t)) \to 0, \; t \to \infty, \tag{3.2}$$

is satisfied.

The observed fault free set of signals, Θ_{NF}, and the observed set of signals, Θ_{F_i}, when a fault, f_i, is present are defined as

$$\Theta_{NF} = \{[u, y] | \exists\, x : \dot{x} = f(x, u, 0), \; y = h(x, u, 0)\} \tag{3.3a}$$
$$\Theta_{F_i} = \{[u, y] | \exists\, x, f_i : \dot{x} = f(x, u, f_i), \; y = h(x, u, f_i)\}. \tag{3.3b}$$

A fault f_i is detectable if and only if the observed signals u and y can not be explained by the fault free case,

$$\Theta_{F_i} \notin \Theta_{NF}. \tag{3.4}$$

The idea with model based diagnosis is seen in Figure 3.1.

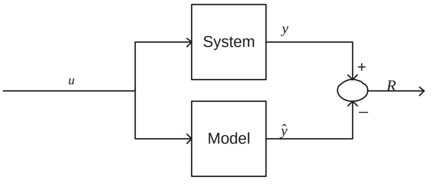

Figure 3.1. Idea with model based diagnosis.

Example 3.1

The system in Figure 3.1 could for example be

$$x = 3u \tag{3.5a}$$
$$y = x + 2 + f, \tag{3.5b}$$

where f is a sensor fault. A residual generator can be generated from system (3.5) as

$$R = F(u, y) = y - 3u - 2. \tag{3.6}$$

The fault, f, is detectable since $\Theta_{F_i} \notin \Theta_{NF}$ and $F(u, y)$ is a residual generator according to the definition since $u, y \in \Theta_{NF} \Rightarrow R = F(u, y) = 0$.

In system (3.5), there are two equations and one unknown variable. The fault f is not seen as an unknown variable since it shall be detected. A condition that must be satisfied if it shall be possible to generate a residual generator is that it shall be possible to determine an unknown variable in more than one way. That is, there must be at least one more equation than there are unknown variables, which means that the system is overdetermined. The residual generator (3.6) can be expressed as $R = x - x$ where the first x is calculated as $x = y - 2$ and the other as $x = 3u$.

A consistency relation (3.7) is an analytical relation between known or measured variables and their derivatives, which is zero in the fault-free case

$$c(y, \dot{y}, \ddot{y}, ..., u, \dot{u}, \ddot{u}, ...) = 0, \quad u, y \in \Theta_{NF}. \tag{3.7}$$

Therefore, consistency relations are often used as residual generators.

Example 3.2

Consider the following state-space model,

$$\dot{x} = -x + u$$
$$y = 2x.$$

A residual generator can be generated as

$$R = c(y, \dot{y}, u) = \frac{\dot{y}}{2} + \frac{y}{2} - u.$$

3.3 Evaluation of Residual Generators

Due to noisy measurements and model uncertainties a residual generator can differ from zero even in the fault free case. To get less noise sensitivity it is possible to construct tests based on residual generators.

3.3.1 Tests

A false alarm occurs if a residual generator gets above a certain threshold when there is no fault present. To minimize false alarms, a test quantity, $T(z)$, can be constructed based on a residual generator and observations, $z = [u, y]^T$. A test reacts if the corresponding test quantity is above or below certain thresholds, J_1 and J_2. The test quantity used in this thesis is a mean value filter,

$$T(z(t)) = \frac{1}{N+1} \sum_{k=0}^{N} R(z(t - kT_s), \tag{3.9}$$

where $N+1$ is the number of samples from the residual generator $R(z(t))$ and T_s the sample time. If the test reacts there is a fault present.

The test is less sensitive to noise but it is also less sensitive to small faults than the residual which the test origin from.

There are different ways to determine the thresholds J_1 and J_2. With the approach used in this thesis it is assumed that the test quantity $T(z)$ is a normal distributed stochastic variable. The thresholds can then be determined from

$$P(|T(z)| < J | z \in \Theta_{NF}) = 1 - P_{FA}, \tag{3.10}$$

where P_{FA} is the probability for false alarm and $J = J_1 = -J_2$.

3.3.2 Test Evaluation

A good test shall have small probability to false alarm and high probability to detect faults. To examine how the test behaves it is possible to make a power function. A power function, $\beta(\theta)$, is a measure for how good a test quantity, $T(z)$, is for a specific fault, θ. A power function is described as

$$\beta(\theta) = P(T(z) \geq J|\theta), \tag{3.11}$$

where θ is a parameter for a specific fault in the system, for example a sensor fault. In the fault free case, the power function, $\beta(\theta)$, shall have a small value since this is the probability for false alarm. When there is a fault present, the power function, $\beta(\theta)$, shall have a high value since this is the probability to detect the fault. A typical power function for a test, $T(z)$, is seen in Figure 3.2.

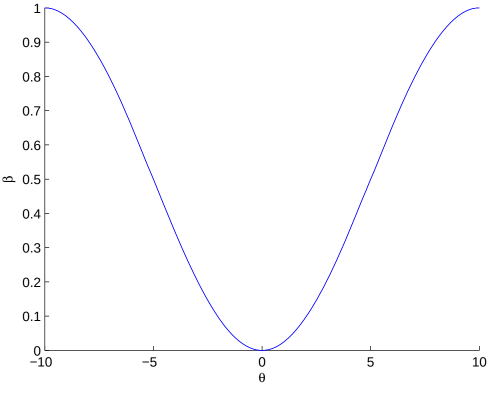

Figure 3.2. Typical power function $\beta(\theta)$ for a test T(z), $\theta = 0$ in the fault free case and $\theta \neq 0$ when there is a fault present.

3.3.3 Fault Detectability

When searching for residual generators it is good to know what qualities that makes a residual generator good. There are a number of factors that can be taken in to consideration, such as fault detectability, fault isolation, fault sensitivity, noise sensitivity and so on.

Fault detectability is the ability to detect certain faults and fault isolation is the ability to decide which fault that has occurred from the faults that are detectable. For fault isolation it is desirable that different residuals are sensitive for and can detect different faults. Fault sensitivity is how sensitive a residual is for a certain fault, that is how much the fault must differ from its nominal behavior before it is detected. Noise sensitivity is how much measurement and process noise affects the residual generator.

In this thesis it is first and foremost the fault detectability that is considered. The fault detectability is considered to be the same for residual generators that are computed from the same set of equations. If the fault detectability is the same for several residual generators, the fault isolation is examined. If several residual generators from the same set of equations are examined, the fault sensitivity is considered for a specific fault. Fault sensitivity is investigated with a power function.

Chapter 4

Structural Analysis

Structural analysis provides many tools for examine different properties of systems. In this chapter the basic theory of structural models, some graph theory and two different ways of handling dynamic systems are presented.

4.1 Structural Models

A structural model is a representation of a system where only the connections between variables and equations are seen and not the actual analytical equations. Structural models can be used in different ways. One way is to make a structural model of a system where the only knowledge is how variables and states are connected but not the actual analytical equations. This makes it easier to analyze the system and then make an analytical model. Another way is to make a structural model of a known analytical system and use the structural model because it is less complex to use when analyzing the system, instead of analyzing all equations. The second approach is used in this thesis.

In the structural model, the set of variables are denoted $\mathcal{Z}$ and the set of equations are denoted $\mathcal{C}$. The set $\mathcal{Z}$ is divided into known and unknown variables.

Consider the state-space system

$$e_1 : \dot{x} = f(x, u) \tag{4.1a}$$

$$e_2 : y = g(x, u), \tag{4.1b}$$

where e_i are equation names, x the states, u inputs and y outputs. The set with unknown variables are then $\mathcal{X} = \{x_1, \dots, x_n\}$ and the set with known variables are $\mathcal{Y} = \{y_1, \dots, y_m, u_1, \dots, u_l\}$. The sets of equations and variables are expressed as

$$\mathcal{Z} = \mathcal{X} \cup \mathcal{Y}$$
$$\mathcal{C} = \{e_1, \dots, e_p\}.$$

The structural model can be represented in different ways. In this thesis the structural matrix[1] and bi-partite graph are used, see [1], [2].

4.2 Bi-Partite Graph

A bi-partite graph, which is a set of vertices and edges, can be used to represent the structural model. The vertex set consists of two sets, $\mathcal{Z}$ and $\mathcal{C}$, and the edges, Υ, representing the connection between a variable and an equation. An edge exists between vertex $z_i \in \mathcal{Z}$ and vertex $c_j \in \mathcal{C}$ if and only if the variable z_i occurs in the equation c_j, see [1].

Definition 4.1 *The structural model of the system* $(\mathcal{C}, \mathcal{Z})$ *is a bi-partite graph* $G(\mathcal{C}, \mathcal{Z}, \Upsilon)$ *where* $\Upsilon \subset \mathcal{C} \times \mathcal{Z}$

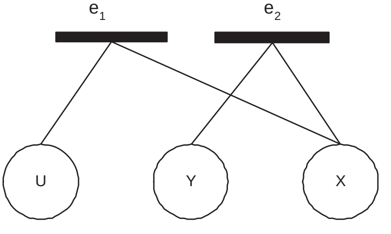

Figure 4.1. Bi-partite graph for system (4.3).

The bi-partite graph in Figure 4.1 is an example of a structural model of the system

$$e_1 : x = u \tag{4.3a}$$
$$e_2 : y = x, \tag{4.3b}$$

with two equations, e_1 and e_2, and three variables, u, y and x.

4.3 Structural Matrix

The structural matrix is, like the bi-partite graph, a way of representing how the equations, $\mathcal{C}$, and the variables, $\mathcal{Z}$, are connected. Consider the system

$$e_1 : f_1(x_1, x_2, u) = 0 \tag{4.4a}$$
$$e_2 : f_2(x_2, u) = 0 \tag{4.4b}$$
$$e_3 : g(x_1, x_2, y) = 0, \tag{4.4c}$$

where x_1, x_2 are unknown variables and u, y known variables. The structural matrix is shown in Table 4.1. Often it is only interesting to study the unknown

[1]Also called biadjacency matrix and incidence matrix

	Unknown		Known	
eq	x_1	x_2	y	u
e_1	X	X		X
e_2		X		X
e_3	X	X	X	

Table 4.1. Structural matrix for system (4.4).

variables in the structural matrix and therefore the structural matrix can be shown in two different ways, one as is in Table 4.1 and the other as in Table 4.2. In the sequel, no difference is made between these two representations.

eq	x_1	x_2
e_1	X	X
e_2		X
e_3	X	X

Table 4.2. Structural matrix for system (4.4) with only unknown variables.

4.4 System Canonical Decomposition

An overdetermined system is a system that contains more equations than unknown variables, that is, from which it is possible to calculate a variable in more than one way and construct a residual generator. From the bi-partite graph and the structural matrix it is possible to determine if the system is overdetermined. Let $(\mathcal{C}, \mathcal{Z})$ represent a system and let $M = (C_1, Z_1)$ where $C_1 \subseteq \mathcal{C}$, $Z_1 \subseteq \mathcal{Z}$ and X is the unknown variables in Z_1. The operator $|A|$ denotes the cardinality of A, which is the number of members in A.

Definition 4.2 *The set M is called structurally overdetermined, SO, if $|C_1| > |X|$, structurally just-determined if $|C_1| = |X|$ and structurally under-determined if $|C_1| < |X|$.*

One more equation than there are unknown variables is needed when searching for residuals, see Section 3.2, and with knowing that a proper subset is a subset S_p to S such that $S_p \subset S$, the following definition of a Minimal Structurally Overdetermined set is useful in this thesis.

Definition 4.3 *The structurally overdetermined set M is called Minimal Structurally Overdetermined, MSO, if there exists no proper structurally overdetermined subsets in M.*

An MSO set which contains differential equations is in semi-explicit form. This is because every differential equation in an MSO set introduces at least one unknown variable. Hence, there is at least one static equation in every MSO set.

Given the assumption that residual generators have the same fault detectability if they are from the same MSO set, best possible fault detectability and fault isolation is achieved if a residual for every possible MSO set is found, see [14].

A structural model can be decomposed in to three different subsystems, see [1]. The decomposition can be done in different ways but one commonly used method is Dulmage-Mendelsohn decomposition, see [2]. The subsystems are overdetermined, just-determined and under-determined. By doing this, it is possible to find the overdetermined part, if it exists, of any system. The interesting part of the decomposed structural matrix is the overdetermined part because it is only in the overdetermined part that residual generators can be found.

4.5 Matching

To construct a residual generator from an MSO set all the unknown variables must be calculated. An unknown variable can be calculated from different equations in an MSO set and to decide how the variable shall be calculated to get a residual generator a bi-partite graph can be used. With a bi-partite graph it is possible to see which variable that must be calculated from which equation. When it is decided from which equation an unknown variable is calculated from the variable is called the matched variable. The matched variable together with the equation that is used to calculate the variable is called a matching.

Definition 4.4 *A matching, Γ, in a bi-partite graph is a subset of edges such that not any edges are sharing the same vertex.*

Definition 4.5 *A complete matching, Γ_M, with respect to $\mathcal{C}$ is when $|\Gamma_M| = |\mathcal{C}|$ and a complete matching with respect to $\mathcal{Z}$ is when $|\Gamma_M| = |\mathcal{Z}|$.*

A matching that is complete with respect to both $\mathcal{C}$ and $\mathcal{Z}$ is called a perfect matching. Perfect matchings are of special interest when searching for matchings because the set of equations and variables is minimal in the sense that all information in the set is used.

A matching, Γ, is written as $\Gamma = \{(e_i, x_j), (e_j, x_i)\}$ which means that, x_j, is the matched variable from equation, e_i, and variable, x_i, is the matched variable from equation, e_j. In the structural matrix, a matching is seen as encircled crosses, $\otimes$.

In a bi-partite graph the edges has no direction. To show how variables and equations are connected in a matching it is possible to introduce an oriented graph.

An oriented graph has the same number of vertices and edges as the corresponding bi-partite graph and the direction of the edges comes from the matching.

Given a matching $\Gamma = \{(e_i, x_j), (e_j, x_i)\}$ and a bi-partite graph, G, the directions of the edges in the oriented graph are described as

$$\text{if } \{e_i, x_j\} \in \Gamma \text{ then } e_i \rightarrow x_j$$
$$\text{if } \{e_i, x_j\} \notin \Gamma \text{ then } x_j \rightarrow e_i.$$

An example of an oriented graph induced from the matching $\Gamma = \{(e_1, x), (e_2, y)\}$ and the bi-partite graph in Figure 4.1, is seen in Figure 4.2.

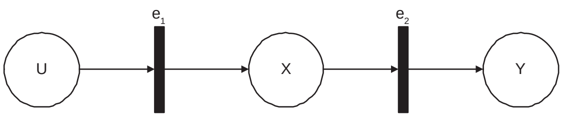

Figure 4.2. Oriented graph induced from the bi-partite graph in Figure 4.1 and the matching Γ.

4.6 Derivative and Integral Causality

There are two main approaches for how differential equations are handled, either with derivative causality or with integral causality.

Causality represents the calculation order that must be followed when variables are matched in a differential equation. This means that with integral causality a differential equation, for example $\dot{x}_1 = x_2$, can only be calculated as $x_1 = x_1(t_0) + \int_{t_0}^{t} (x_2(\tau))d\tau$. With derivative causality instead, only the variable x_2 can be matched and is calculated as $x_2 = \frac{d}{dt}x_1$.

A side effect of this is that with integral causality the initial condition of differentiated variables must be known and with derivative causality the derivative of a variable must be known or possible to estimate.

4.7 Handling Derivatives in Structural Models

In a structural model there are two main approaches for handling derivatives of variables, see [14]. When the differentiated variable is handled as the same variable as the non-differentiated variable, the structural model is called a Differentiated-Lumped Structural-Model (DLSM). When the differentiated variable and the non-differentiated variable are handled as different variables, the structural model is called a Differentiated-Separated Structural-Model (DSSM).

To show the difference between DSSM and DLSM consider the equation system

$$e_1 : \dot{x} = -x + u \tag{4.6a}$$
$$e_2 : y = 2x, \tag{4.6b}$$

	Unknown		Known	
eq	$\dot{x}$	x	y	u
e_1	X	X		X
e_2		X	X	

Table 4.3. Structural matrix with DSSM for system (4.6).

	Unknown	Known	
eq	x	y	u
e_1	X		X
e_2	X	X	

Table 4.4. Structural matrix with DLSM for system (4.6).

which contains the differentiated variable $\dot{x}$ and the non-differentiated variable x. The two structural models are seen in Table 4.3 and 4.4.

The system in Table 4.4 is an overdetermined system but the system in Table 4.3 is a just-determined system. This is a direct effect of the use of DLSM where there is no structurally difference between a variable x_i and its derivative $\dot{x}_i$. But with DSSM these two variables are handled as different variables and there is no information about the relation between them.

To avoid this, it is possible to introduce an extra equation that describes the relation between a variable and its time derivative. The differentiated variable $\dot{x}$ is replaced with x^d to avoid misunderstandings with the notation for time derivatives. The extra equation and the renamed variable are seen in (4.7).

$$e_1 : x^d = -x + u \tag{4.7a}$$

$$e_2 : y = 2x \tag{4.7b}$$

$$d_1 : x^d = \frac{d}{dt}x \tag{4.7c}$$

Introducing the extra information in the DSSM in Table 4.3 results in a new structural representation, called an Extended Differentiated-Separated Structural-Model (EDSSM), which is seen in Table 4.5. The systems in Table 4.4 and Table 4.5 are now both overdetermined systems with one more equation than unknown variables.

	Unknown		Known	
eq	x^d	x	y	u
e_1	X	X		X
e_2		X	X	
d_1	X	X		

Table 4.5. Structural matrix with EDSSM for system (4.6).

The extra equation is necessary to get an unambiguous decided calculation order if both integral and derivative causality are used. This is illustrated in Example 4.1.

Example 4.1

	x^d	x
e_1	×	×
d_1	×	×

Table 4.6. Structural matrix for (4.8) with EDSSM.

	x
e_1	×

Table 4.7. Structural matrix for (4.8) with DLSM.

Consider the differential equation

$$e_1 : \dot{x} = -x + u. \tag{4.8}$$

The structural model for the differential equation is represented with DLSM in Table 4.7 and with EDSSM in Table 4.6. From Table 4.6, the variable x can be matched in two different ways which gives two different ways to calculate the variable. Either as $x = \int (-x + u)\, dt$ or $x = -\dot{x} + u$. From Table 4.7, the variable x can only be matched in one way and now it is not decided how the variable x is calculated.

When only one of the two methods, integral and derivative causality, is used there must be something that marks which variables that can not be matched. Consider the equation system

$$x_i^d = f(X) \tag{4.9a}$$

$$x_i^d = \frac{d}{dt} x_i, \tag{4.9b}$$

where $X = [x_1, ..., x_n]^T$. With integral causality, only variable x_i can be matched in equation (4.9b) and with derivative causality, only the variable x_i^d can be matched in equation (4.9b). The variables that can not be matched are marked with a Δ in the structural matrix. This holds not only for the non-matchable variables in differential equations but for all variables that are not matchable, for example variables in non-invertible equations.

In the sequel only EDSSM is used, and to reduce the number of equations when a system is presented, an equation system is presented as in (4.6) but is handled as in (4.7). That is, the analytical equations are presented without the renamed variables and the extra equations but they are included in the structural model.

Since a new equation and a new variable are introduced, named d_i and x_i^d respectively, the sets $\mathcal{C}$ and $\mathcal{Z}$ have changed. Let $E = \{e_1, \ldots, e_p\}$, $D = \{d_1, \ldots, d_k\}$ and $\mathcal{X}^d = \{x_1^d, \ldots, x_k^d\}$, where p is the number of equations and k is the number of differential equations in the system. This gives new sets of variables and equations as

$$\mathcal{Z} = \mathcal{X}^d \cup \mathcal{X} \cup \mathcal{Y}$$
$$\mathcal{C} = E \cup D.$$

4.8 Strongly Connected Components

If a perfect matching exists for a system, it can be found from the structural matrix and this matching gives a computation sequence. Depending on system properties the sequence can sometimes not be unambiguous decided because the system contains strongly connected components (SCC). Strongly connected components are variables that depend on each other and must be calculated at the same time.

Strongly connected components are seen in the structural matrix for a just-determined system as blocks on the diagonal if the matrix is decomposed to a block upper triangular matrix. A structural matrix can always be decomposed to a block upper triangular matrix with some row and column permutations, see [2]. The decomposed matrix can for example be computed with MATLAB using the command *dmperm*.

Strongly connected components contain one or several algebraic loops, which is illustrated in Figure 4.3 and an algebraic loop is exemplified in Example 4.2.

Example 4.2

Consider the system of equations

$$e_1 : y_1 = x_1 + x_2 \tag{4.11a}$$
$$e_2 : y_2 = 2x_1 - x_2. \tag{4.11b}$$

The structural matrix for this system is seen in Table 4.8 which contains strongly connected components. The matching, $\Gamma = \{(e_2, x_1), (e_1, x_2)\}$, gives a computation sequence

$$x_1 = \frac{y_2 + x_2}{2}$$
$$x_2 = y_1 - x_1,$$

which contains an algebraic loop because to calculate x_1, x_2 is needed and vice verse.

eq	x_1	x_2
e_1	×	×
e_2	×	×

Table 4.8. Structural matrix for system (4.11) with an algebraic loop.

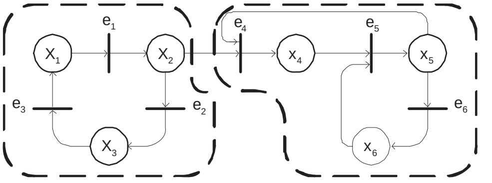

Figure 4.3. Oriented graph with two SCC.

The oriented graph in Figure 4.3 consists of three different algebraic loops but two different strongly connected components.

Strongly connected components can induce three different forms of algebraic loops. The algebraic loops can either be static, differential or both static and differential. The purely static loops contain no differential equations and to solve them, some sort of equation solver is needed. If static loops occur they are handled as not solvable. If differential loops, or loops with both static and differential equations occur, they can be solved in some cases, which are further discussed in Sections 5.4 and 6.4.

Part II

Residual Generation with Different Causality

Chapter 5

Integral Causality

The aim of this chapter is to discuss the existing method to find residual generators. The discussion includes the need of initial conditions when solving differential equations and how strongly connected components are handled with integral causality.

5.1 Finding Residual Generators from Mathematical Models

The process to find residual generators from mathematical models can be divided to a number of steps, see [4]. The first step is to transform the mathematical model to a structural matrix. The overdetermined part is extracted from the structural matrix because it is only when redundant information exists, residual generators can be created.

To create a residual generator only one more equation is needed than there are unknown variables. Hence, MSO sets are searched for in the overdetermined part of the structural matrix. In each MSO set it is possible to remove one equation at the time and use that one as residual equation. When an equation is removed from the MSO set, the new set is just-determined and if it exists a perfect matching the residual generator can be realized unless it contains non-solvable strongly connected components. From the perfect matching all variables can be calculated and used in the residual equation. The chain from a mathematical model to a residual generator is illustrated in Figure 5.1.

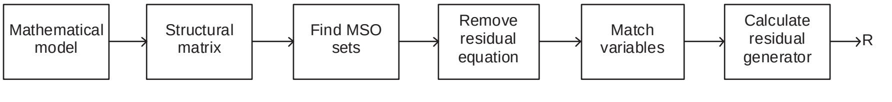

Figure 5.1. Process of finding a residual generator from a mathematical model.

All equations except the extra equation, see Section 4.7, can be used as residual equations. The extra equation is not used because this is an equation that is added afterwards and is considered to be a dummy-equation. However, there are no loss of residual generators when not using this equation because every residual generator that is found with the extra equation as residual equation can be found in the same MSO set with another equation as residual equation, see [4].

Integral causality is used when variables are matched and calculated, which represents the two last steps of the chain in Figure 5.1. In the following two chapters, modifications that can be made in these two steps when residual generators are searched for are presented.

5.2 Structure of Residual Generators with Integral Causality

The structure of a residual generator generated with integral causality, as described in Section 5.1, is on the form

$$\dot{x} = f(x, u, y) \tag{5.1a}$$

$$R_{IC} = g(x, u, y). \tag{5.1b}$$

To calculate x an initial condition is needed.

5.3 Initial Conditions with Integral Causality

When using integral causality the initial condition for the differentiated variable must be known if it shall be possible to calculate the variable. This assumption can be partly modified. If the system is stable, the initial condition can be chosen arbitrary. This will work because the solution will converge to the correct value after some time. The difference with knowing the correct initial condition is that the residual generator will be zero from the beginning if the system is fault-free.

For an unstable system with correct initial condition, the residual generator will not converge to zero in many cases, due to process and measurement noise. This is illustrated in Example 5.1.

Example 5.1

Consider the unstable system

$$\dot{x} = x + u + w \tag{5.2a}$$

$$y = x + v, \tag{5.2b}$$

which yields the residual generator

$$\dot{x} = x + u \tag{5.3a}$$

$$R = y - x. \tag{5.3b}$$

Three simulations of the system and the residual generator were done and in Figure 5.2 the results from the three simulations are shown. The first was done without noise and with correct initial condition, $x_0 = 0$. The second simulation was done without noise and with a faulty initial condition, $x_0 = 0.001$. The third simulation was done with process and measurement noise, where the noises are Gaussian with variance 0.05 and 0.1 respectively, and with correct initial condition. Only one residual generator converges to zero and that residual generator gets its values from the model without noise and with the correct initial condition.

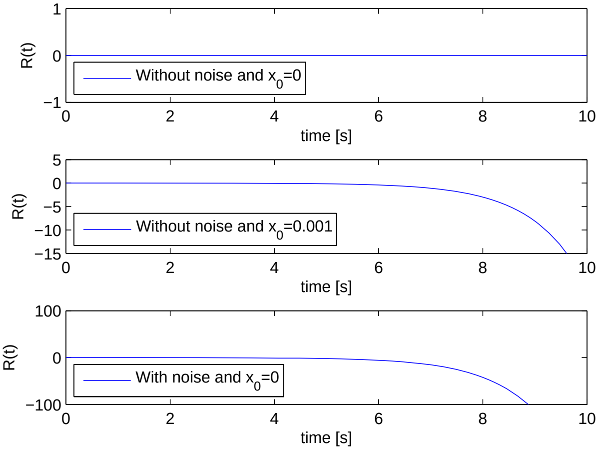

Figure 5.2. Three simulations of residual generator (5.3b) with different initial conditions.

The fact that a stable system converge to the correct value and an unstable system does not can be seen by writing the solution of a linear state-space system (2.1) as

$$x(t) = e^{At}x_0 + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau,$$

where the first term $e^{At}x_0$ converge to zero if the system is stable, independently of x_0, see Section 2.2. For an unstable system with a correct initial condition and without noise the residual generator is zero. An unstable residual generator will not converge if there is noise present, see Figure 5.2. There are different methods to stabilize unstable residual generators and one method is to use observer theory, see [4]. This method has already been investigated and is not discussed further.

For non-linear systems there are many different methods to investigate stability, see [9]. Even if all equilibrium points to a non-linear system are stable there is no guarantee that the solution will converge to the correct value if the initial condition has been chosen badly. This is shown in Example 5.2.

Example 5.2

Consider the non-linear state-space system

$$\dot{x} = -\sin x + u \tag{5.4a}$$
$$y = x, \tag{5.4b}$$

for which the residual generator

$$\dot{x} = -\sin x + u \tag{5.5a}$$
$$R = y - x, \tag{5.5b}$$

can be designed. The residual generator is sensitive for the initial condition x_0. The system was simulated in MATLAB/SIMULINK with two different initial conditions. Both simulations were done with faulty initial conditions. The correct initial condition for the system is $x_0 = 0$ and the initial conditions in the simulation was $x_0 = 2$ and $x_0 = 3.5$ respectively. The two simulations are shown in Figure 5.3. The simulations were fault-free but the residual generator converges to two different values.

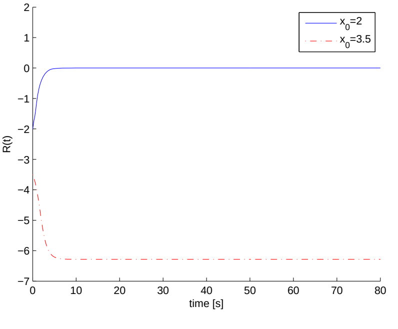

Figure 5.3. Two simulations of system (5.5) with different initial conditions.

For a stable linear system a faulty initial condition will work satisfactory. Stability of linear systems is easy to examine, see Section 2. For non-linear systems both the stability and to what value the solution converge must be examined. However, this is out of scope of this thesis and will not be studied further.

The conclusion is that correct initial conditions are needed for non-linear systems but for linear stable systems the initial conditions can be chosen arbitrary.

5.4 Solvability of Strongly Connected Components with Integral Causality

When strongly connected components only contain differential equations they induce a differential loop on the form

$$e_1 : x^d = f(x, z) \tag{5.6a}$$

$$d_1 : x^d = \frac{d}{dt}x. \tag{5.6b}$$

The structural model for these equations is seen in Table 5.1 with matched variables marked, which also is the only possible matching. The matching in Table 5.1 is a loop and this loop can be solved numerically with the Euler forward method as

$$x(t+T) = x(t) + Tf(x(t), z(t)), \tag{5.7}$$

where T is the sample time. The loop and the solved loop are illustrated in Figure 5.4 and Figure 5.5.

eq	x^d	x
e_1	$\otimes$	$\times$
d_1	Δ	$\otimes$

Table 5.1. Structural matrix for SCC containing differential equations with matched variables.

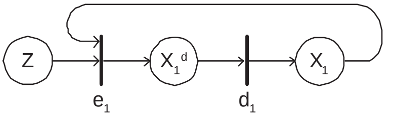

Figure 5.4. An oriented graph which contains a differential loop.

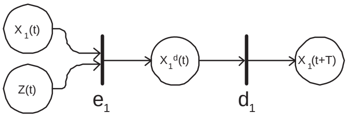

Figure 5.5. An oriented graph which contains a solved differential loop.

If the strongly connected components contain both differential and static equations the induced loop can be solved if the static variables do not induce a static loop. A solvable and a non-solvable loop that contains both differential equations and static equations are illustrated in Examples 5.3 and 5.4.

Example 5.3

Consider the semi-explicit system

$$e_1 : \dot{x}_1 = -x_1 + 2x_2 \quad (5.8a)$$

$$e_2 : y = 3x_1 + x_2. \quad (5.8b)$$

The structural matrix for system 5.8 is seen in Table 5.2 and contains strongly connected components. The circles show a matching that induce an algebraic loop that contains both static and differential variables but can be solved numerically with the Euler forward method.

eq	x_1^d	x_1	x_2
e_1	$\otimes$	$\times$	$\times$
e_2		$\times$	$\otimes$
d_1	Δ	$\otimes$	

Table 5.2. Structural matrix for system 5.8 containing both differential and static equations with matched variables.

Example 5.4

Consider another semi-explicit system

$$e_1 : \dot{x}_1 = -x_1 + 2x_2 + 5x_3 \quad (5.9a)$$

$$e_2 : y_1 = 3x_1 + x_2 + 3x_3 \quad (5.9b)$$

$$e_3 : 0 = x_2 + 4x_3. \quad (5.9c)$$

The structural matrix for system 5.9 is seen in Table 5.3 and contains strongly connected components. The unknown variables can not be matched without inducing a loop which contains both static and differential variables that can not be solved. This is because the static variables will induce a static loop which can not be solved, see Section 4.8.

eq	x_1^d	x_1	x_2	x_3
e_1	$\times$	$\times$	$\times$	$\times$
e_2		$\times$	$\times$	$\times$
e_3			$\times$	$\times$
d_1	Δ	$\times$		

Table 5.3. Structural matrix for system 5.9 containing both differential and static equations.

When there are strongly connected components containing both differential and static equations, a solvable matching can be found if the structural matrix, with the differential equations and the differential variables removed, does not contain any strongly connected components.

Chapter 6

Derivative Causality

When using integral causality the initial conditions of a system have to be known for certain systems. In many cases this is not a realistic assumption. Instead derivative causality can be used, which do not need the initial conditions but instead derivatives are needed. Derivatives of measurements are supposed to be known in this chapter, but how they are estimated is further discussed in Section 8.1.

In this chapter the use of derivative causality is motivated and there is a discussion of how some arising difficulties are handled.

6.1 Structure of Residual Generators with Derivative Causality

The structure of a residual generator generated with derivative causality is different than the structure that was given with integral causality. The structure of a residual generator with derivative causality is on the form

$$x = f(u, y) \tag{6.1a}$$

$$R_{DC} = g(x, \dot{x}, \dots), \tag{6.1b}$$

where the derivatives are estimated with the methods described in Section 8.1. With derivative causality, the last two steps in Figure 5.1, see Section 5.1, are changed. Variables are matched as described in Section 4.6 and residual generators are calculated as (6.1).

6.2 Introduction to Derivative Causality

The use of derivative causality is motivated in Example 6.1 where an unstable system is considered.

Example 6.1

Consider the following unstable state-space system

$$\dot{x} = x + u + w \tag{6.2a}$$
$$y = x + v + f_y, \tag{6.2b}$$

where v and w are Gaussian noise and f_y is an additive fault on sensor y. The system is kept stable by a control loop but it is only the uncontrolled system that is diagnosed. With integral causality a found residual generator is

$$\dot{x} = x - u \tag{6.3a}$$
$$R_{IC} = y - x. \tag{6.3b}$$

With derivative causality a found residual generator is

$$R_{DC} = \frac{d}{dt}y - y - u. \tag{6.4a}$$

By simulating the system and the residual generator in MATLAB/SIMULINK, the behavior of the residuals was investigated. In the simulation the process noise w has variance 0.05 and the measurement noise v variance 0.1. Both noises have zero mean value. A bias fault in sensor y occurred after $5s$. The result of the simulation is seen in Figure 6.1. The residual R_{DC} detects the bias fault but the residual R_{IC} is of no use because it is unstable and therefore very sensitive to noise.

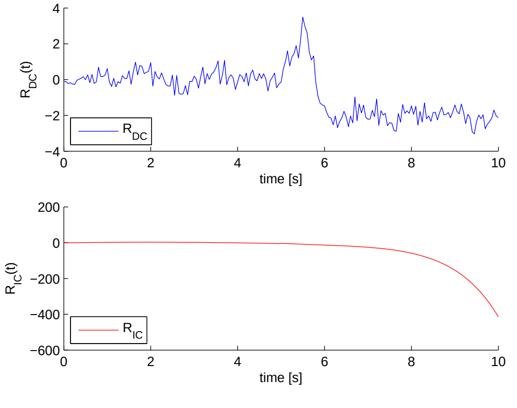

Figure 6.1. Simulation of residual R_{IC} and R_{DC}.

The use of derivative causality has advantages compared to integral causality. One advantage is seen in Example 6.1 where an unstable system can be handled better than with integral causality. On the other hand, there are also disadvantages, namely that derivatives of noisy signals are difficult to estimate and the occurrence of differential loops. The incidence of differential loops is investigated in Example 6.2.

Example 6.2

Consider the following state-space system

$$e_1 : \dot{x}_1 = -x_1 - 5x_2 + u \quad (6.5a)$$
$$e_2 : \dot{x}_2 = -x_2 + u \quad (6.5b)$$
$$e_3 : y_1 = x_1 + x_2 \quad (6.5c)$$
$$e_4 : y_2 = x_2. \quad (6.5d)$$

It is possible to find three MSO sets from the system and in two of the sets it is possible to find a residual generator without induced differential loops. The structural matrices of the MSO sets from which it is possible to generate residuals are seen in Tables 6.1 and 6.2. The two realizable residuals are seen in (6.6).

eq	x_1^d	x_1	x_2
e_1	Δ	$\times$	$\times$
e_3		$\otimes$	$\times$
e_4			$\otimes$
d_1	$\otimes$	Δ	

Table 6.1. Structural matrix for MSO 1 from system (6.5).

eq	x_2^d	x_2
e_2	Δ	X
e_4		$\otimes$
d_2	$\otimes$	Δ

Table 6.2. Structural matrix for MSO 2 from system (6.5).

$$R_1 = \dot{y}_1 - \dot{y}_2 + y_1 + 4y_2 - u \quad (6.6a)$$
$$R_2 = \dot{y}_2 + y_2 - u \quad (6.6b)$$

In total there are three MSO sets and in the third MSO set there are four different ways of matching variables but all lead to differential loops[1], which are non-solvable and further discussed in Section 6.4. If it would be possible to find a residual generator from each MSO set, higher fault detectability and better fault isolation could possible be achieved.

[1]All found MSO sets and residual generatos can be seen in Appendix A

6.3 Initial Conditions for Derivative Causality

An advantage with derivative causality is that initial conditions for the system states are not needed, see [1]. This is a truth with modifications. When derivatives are estimated an initial condition is usually needed, see Section 8.1. When consistency relations are realized in state-space form to remove derivatives of signals as inputs, an initial condition is also needed, see Section 8.2. An initial condition is therefore needed in many cases, even with derivative causality. Since the methods to estimate derivatives and realization in state-space form are stable the initial condition can be chosen arbitrary.

The conclusion is that correct initial conditions are not needed when derivative causality is used.

6.4 Solvability of Strongly Connected Components with Derivative Causality

Strongly connected components which consist of differential equations will form a differential loop, see Section 5.4. The difference with derivative causality is that different variables are matchable, see Section 4.6.

There are different methods to solve differential loops but they rely on previous time samples in some way, see [17]. This means that the initial condition is needed and the loop is solved in the same way as the loop is solved with integral causality, see Section 5.4. The variables are no longer matched as they should when derivative causality is used, see Section 4.6. Instead the variables are matched in the same way as when integral causality is used and this is not a way of solving differential loops with derivative causality. Hence, differential loops are considered non-solvable when derivative causality is used.

Chapter 7

Mixed Causality

Previously in this thesis either integral or derivative causality have been used when searching for residual generators. Each of these lead to different ways of handling differential equations, that is, limits in the possible matchings for the differential equations. If there exists systems were two differential equations have to be handled differently to be able to find any residual generators, neither integral nor derivative causality would find any residual generators. Hence, it is desirable to have a method where differential equations can be handled in different ways in the same system.

The purpose of this chapter is to discuss mixed causality and present an algorithm that extracts all possible residual generators.

7.1 Structure of Residual Generators with Mixed Causality

The structure of a residual generator generated with mixed causality is on the form

$$\dot{x}_1 = f_1(x_1, x_2, u, y) \tag{7.1a}$$

$$x_2 = f_2(x_1, x_2, u, y) \tag{7.1b}$$

$$R_{MC} = g(x_1, x_2, \dot{x}_2, \ddot{x}_2, \ldots, u, y), \tag{7.1c}$$

where the derivatives are estimated with the methods described in Section 8.1. With mixed causality the last two steps when extracting residual generators are changed, see Figure 5.1 in Section 5.1. How variables are matched is presented in Section 7.4 and the residual generator is computed as (7.1).

7.2 Introduction to Mixed Causality

The advantage with mixed causality is illustrated in three examples, Example 7.1, 7.2 and 7.3. In the first example no restrictions of possible matchings are made, which means that it is assumed that the initial conditions for all states are known and that derivatives can be estimated.

Example 7.1

Consider the following MSO set that is in semi-explicit form

$$e_1 : \dot{x}_1 = f_1(x_1, x_2, x_3) \tag{7.2a}$$
$$e_2 : \dot{x}_2 = f_2(x_1) \tag{7.2b}$$
$$e_3 : y_1 = x_1 \tag{7.2c}$$
$$e_4 : y_2 = f_3(x_2, x_3), \tag{7.2d}$$

where f_3 is a non-invertible function for both x_2 and x_3. Since no variable can be matched from equation e_3 that equation is used as residual equation. The structural matrix of (7.2) is seen in Table 7.1. From the structural matrix it can be seen that only one possible residual generator exists and it has no differential loops. This residual generator is derived with equation, e_4, as residual equation and matching, $\Gamma = \{(e_1, x_3), (e_2, x_2^d), (e_3, x_1), (d_1, x_1^d), (d_2, x_2)\}$. The matching Γ would not have been found with integral or derivative causality separately. Figure 7.1 shows the oriented graph corresponding to the matching, Γ.

eq	x_1^d	x_2^d	x_1	x_2	x_3
e_1	$\times$		$\times$	$\times$	$\otimes$
e_2		$\otimes$	$\times$		
e_3			$\otimes$		
e_4				Δ	Δ
d_1	$\otimes$		$\times$		
d_2		$\times$		$\otimes$	

Table 7.1. Structural matrix of (7.2) without restrictions on the differential equation.

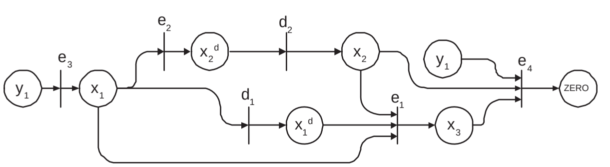

Figure 7.1. Oriented graph for the matching Γ and the structural matrix in Table 7.1.

In Example 7.1 the use of mixed causality is motivated because a possible matching, which can not be found using integral or derivative causality separately, is found.

Another form of mixed causality is presented in Example 7.2, where derivative causality is used and for the induced differential loops, which is discussed in Chapter 6, integral causality is used to avoid the loops.

Example 7.2

Consider the following MSO set that is in semi-explicit form

$$e_1 : \dot{x}_1 = f_1(x_1, x_2, x_3, u) \tag{7.3a}$$
$$e_2 : \dot{x}_2 = f_2(x_2, x_3) \tag{7.3b}$$
$$e_3 : y_1 = h_1(x_2) \tag{7.3c}$$
$$e_4 : y_2 = h_2(x_1, x_3), \tag{7.3d}$$

where h_2 is a non-invertible function. The structural matrix for (7.3) with derivative causality is seen in Table 7.2. The decomposed matrix for the structural matrix in Table 7.2 without equation, e_4, is seen in Table 7.3. In the top left corner of Table 7.3 there are strongly connected components that induce a differential loop. The problem with this loop can be avoided by switching causality and a solution can be computed numerically, see Section 5.4. The found matching is $\Gamma = \{(e_3, x_2), (d_2, x_2^d), (e_2, x_3), (e_1, x_1), (d_1, x_1^d)\}$ and Figure 7.2 shows the oriented graph corresponding to the matching, Γ.

eq	x_1^d	x_2^d	x_1	x_2	x_3
e_1	Δ		$\times$	$\times$	$\times$
e_2		Δ		$\times$	$\times$
e_3				$\times$	
e_4			Δ		Δ
d_1	$\times$		Δ		
d_2		$\times$		Δ	

Table 7.2. Structural matrix of (7.3) with derivative causality.

eq	x_1^d	x_1	x_3	x_2^d	x_2
d_1	$\times$	Δ			
e_1	Δ	$\times$	$\times$		$\times$
e_2			$\times$	Δ	$\times$
d_2				$\times$	Δ
e_3					$\times$

Table 7.3. Decomposed structural matrix of (7.3) without residual equation, e_4, with the strongly connected components marked.

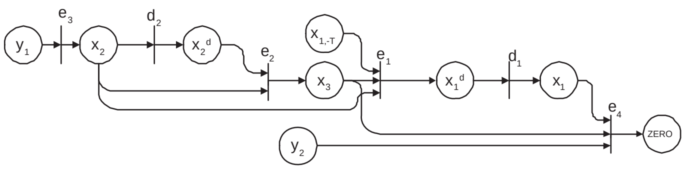

Figure 7.2. Oriented graph for the matching Γ and the structural matrix in Table 7.2, where $x_{1,-T}$ symbolize the variable x_1 in the previous sample.

Example 7.2 shows that a differential loop can be solved with integral causality, this is further discussed in Section 7.3. A third form of mixed causality is presented in Example 7.3.

Example 7.3

Consider the following linear system

$$e_1 : \dot{x}_1 = -x_1 + u \tag{7.4a}$$

$$e_2 : \dot{x}_2 = x_1 \tag{7.4b}$$

$$e_3 : y = x_2. \tag{7.4c}$$

Independent of the initial condition, x_{1,t_0}, the state, x_1, converge to the right value. However, the state, x_2, does not converge to the right value if the initial condition, x_{2,t_0}, is chosen badly or there is noise present in the system. Therefore it is desirable to match state x_1 but not x_2 with integral causality. Assume, that from some system knowledge it is well-known that it is difficult to differentiate state x_1 but not x_2. It is actually y that is differentiable and due to high frequency components in the signal it is difficult to get any good information from higher order derivatives. Hence, it is desirable to match state x_2 but not x_1 with derivative causality. The structural matrix of (7.4) and the knowledge presented above is seen in Table 7.4. Only one possible perfect matching, $\Gamma = \{(e_3, x_2), (e_2, x_2^d), (d_1, x_1), (e_1, x_1^d)\}$, is found from the structural matrix when neither equation d_1 or d_2 is used as residual equation. Figure 7.3 shows the oriented graph corresponding to the matching, Γ.

eq	x_1^d	x_2^d	x_1	x_2
e_1	$\otimes$		$\times$	
e_2		$\times$	$\times$	
e_3				$\otimes$
d_1	Δ		$\otimes$	
d_2		$\otimes$		Δ

Table 7.4. Structural matrix of (7.4).

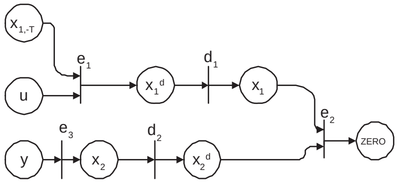

Figure 7.3. Oriented graph for the matching Γ and the structural matrix in Table 7.4, where $x_{1,-T}$ symbolize the variable x_1 in the previous sample.

In Example 7.3 one form of mixed causality is shown, which can be useful if there are knowledge about which states the initial conditions are known for and which states that can be differentiated or not.

Consider the linear system studied in Example 7.3 with added measurement noise. Integral, derivative and mixed causality each generates one residual generator.

$$\dot{x}_1 = -x_1 + u \tag{7.5a}$$
$$\dot{x}_2 = x_1 \tag{7.5b}$$
$$R_{IC} = y - x_2 \tag{7.5c}$$

$$R_{DC} = u - \dot{y} - \ddot{y} \tag{7.6a}$$

$$\dot{x}_1 = -x_1 + u \tag{7.7a}$$
$$R_{MC} = x_1 - \dot{y} \tag{7.7b}$$

The initial conditions for the system can not be determined and are therefore assumed to be zero. Power functions for tests based on the different residual generators are presented in Figure 7.4 with varying actuator bias fault. The probability for the test to react when no fault has occurred is set to 5%. According to Figure 7.4 the residual generator found with mixed causality has the best fault detectability. The residual generator found with integral causality detects a fault even when no fault is present, this is because the initial condition for the system has not been estimated correctly and there is noise present in the system. This result shows that mixed causality is useful when trying to detect faults in a system where derivatives can be estimated and some differential equations are unstable.

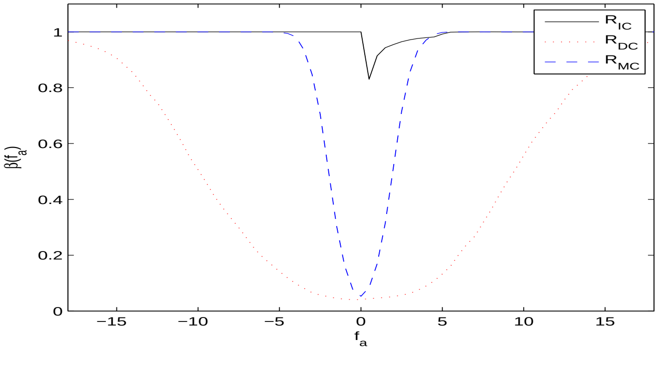

Figure 7.4. Power functions for R_{IC}, R_{DC} and R_{MC} with actuator bias fault.

All three Examples 7.1, 7.2 and 7.3 presented in this section are different forms of mixed causality, each with different characteristics. The first example motivates the use of mixed causality because no residual generators are found with either integral or derivative causality separately. The second example shows how differential loops are solved with integral causality. The third and last example presented above shows that with some system knowledge it is desirable to handle different differential equations with different methods.

7.3 Solvability of Strongly Connected Components with Mixed Causality

Strongly connected components can induce two different forms of loops, see Section 4.8. If they are static they are considered not solvable. Differential loops are considered solvable if they are matched with integral causality, see Section 5.4, and considered not solvable if they are matched with derivative causality, see Section 6.4. Hence, differential loops are matched with integral causality in this chapter.

7.4 Structural Methods for Finding Residual Generators with Mixed Causality

An algorithm for finding residual generators with mixed causality is presented and illustrated with an example in this section. The algorithm does not have any matching restriction of the type that is introduced in Section 7.2.

A brief outline of the algorithm is:

1. Transform the simulink model to a structural model.
2. Find all possible MSO sets.
3. Find all realizable residual generators in every MSO set.
4. Based on an evaluation of the residual generators, remove all without useful properties, for example unstable residual generators.

Algorithm 1 is illustrated in Example 7.4. Since no restrictions are made in this algorithm for how differential equations are handled, all residual generators found with either integral or derivative causality are subsets of the set of residual generators found with this algorithm.

Algorithm 1: Residual generation with mixed causality

```
Input   : A SIMULINK-model file, model.mdl
R:= ∅
ME:= Sim2Me (model.mdl)
ME:= AddExtraEquations (ME)
SM:= Me2Se (ME)
SM_MSO:= FindAllMso (SM)
forall MSO sets M ∈ SM_MSO do
    forall equations e ∈ M \(extra equations) do
        m:= M \ e
        SCC:= FindAllStronglyConnectedComponents (m)
        m':= m \ SCC
        Γ':= FindPerfectMatching (m')
        forall strongly connected components S_i ∈ SCC do
            Γ_{S_i}:= FindSolvablePerfectMatching (S_i)
        end
        Γ_SCC:= ⋃_i Γ_{S_i}
        Γ:= Γ' ∪ Γ_SCC
        if |Γ | ≠ 0 then
            R':= CalculateResidualGenerator (Γ, e)
            R:= R ∪ R'
        end
    end
end
R:= EvaluateResidualGenerators (R)
Output: A set of residual generators, R
```

The first step where a Simulink model is transformed to a structural model is divided in three sub steps. All model equations are first extracted from the simulink model with `Sim2Me`. The extra equations, see Section 4.7, are then added with `AddExtraEquations`. The model equations are in the third sub step transformed to a structural model with `Me2Sm`. The two functions `Sim2Me` and `Me2Sm` are based on algorithms described in [5].

The second step to find all possible MSO sets is performed by function `FindAllMso`, see [5].

The third step to find all solvable residual generators in a MSO set is divided in four sub steps. All strongly connected components are in the first sub step found with `FindAllStronglyConnectedComponents`. A perfect matching is searched for with `FindPerfectMatching` on the structural model without strongly connected components. For every strongly connected components is the solvable perfect matching found with `FindSolvablePerfectMatching` if it exists. In the last sub step the found solvable matching is realized with `CalculateResidualGenerator`.

The fourth and final step in the algorithm is performed in function `EvaluateResidualGenerators`, where the residual generators are evaluated in a stability and fault sensitive sense.

Example 7.4

Consider the linear system

$$e_1 : \dot{x}_1 = -x_1 - 5x_2 + u \tag{7.8a}$$
$$e_2 : \dot{x}_2 = -x_2 + u \tag{7.8b}$$
$$e_3 : y_1 = x_1 + x_2 \tag{7.8c}$$
$$e_4 : y_2 = x_2. \tag{7.8d}$$

In Example 6.2 residual generators for two of in total three MSO sets for system (7.8) are presented. Consider the third MSO set, M, with equations $\{e_1, e_2, e_3, d_1, d_2\}$, where no residual generators with derivative causality is found. Use Algorithm 1 on the third MSO set, M.

The structural matrix of M is found in Table 7.5. Every equation, e_i, except the extra equations are removed once at a time and the resulting matrices, m_i, are seen in Table 7.6, 7.7 and 7.8 with the strongly connected components marked.

eq	x_1^d	x_2^d	x_1	x_2
e_1	$\times$		$\times$	$\times$
e_2		$\times$		$\times$
e_3			$\times$	$\times$
d_1	$\times$		$\times$	
d_2		$\times$		$\times$

Table 7.5. Structural matrix of the third MSO set, M, equations $\{e_1, e_2, e_3, d_1, d_2\}$ of system (7.8).

eq	x_1^d	x_1	x_2	x_2^d
d_1	$\otimes$	$\times$		
e_3		$\otimes$	$\times$	
d_2			$\otimes$	$\times$
e_2			$\times$	$\otimes$

Table 7.6. Decomposed structural matrix of MSO 3, without equation e_1, m_1, containing one SCC.

eq	x_2^d	x_1	x_2	x_1^d
d_2	$\otimes$		$\times$	
e_3		$\times$	$\otimes$	
e_1		$\times$	$\times$	$\otimes$
d_1		$\otimes$		$\times$

Table 7.7. Decomposed structural matrix of MSO 3, without equation e_2, m_2, containing one SCC.

eq	x_1^d	x_1	x_2	x_2^d
d_1	×	⊗		
e_1	⊗	×	×	
d_2			⊗	×
e_2			×	⊗

Table 7.8. Decomposed structural matrix of MSO 3, without equation e_3, m_3, containing two SCC

All strongly connected components found are differential and are therefore solvable if they are matched with integral causality. The resulting matchings for every MSO set are

$$\Gamma_1 = \{(d_1, x_1^d), (e_3, x_1), (d_2, x_2), (e_2, x_2^d)\} \tag{7.9a}$$

$$\Gamma_2 = \{(d_2, x_2^d), (d_1, x_1)(e_3, x_2), (e_1, x_1^d)\} \tag{7.9b}$$

$$\Gamma_3 = \{(d_1, x_1), (e_1, x_1^d), (d_2, x_2), (e_2, x_2^d)\}. \tag{7.9c}$$

The found matchings (7.9) results in three residual generators, see Figure 7.5, 7.6 and 7.7.

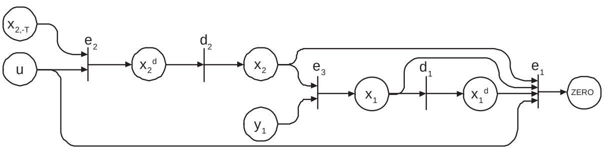

Figure 7.5. Oriented graph for residual generator R_1 yield by matching Γ_1 and the structural matrix in Table 7.6, where $x_{2,-T}$ symbolize the variable x_2 in the previous sample.

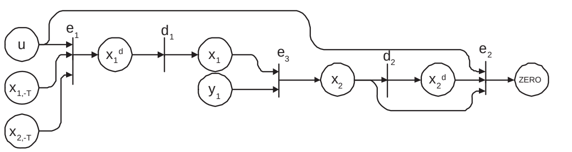

Figure 7.6. Oriented graph for residual generator R_2 yield by matching Γ_2 and the structural matrix in Table 7.7, where $x_{i,-T}$ symbolize the variable x_i in the previous sample.

The last step in the algorithm is to evaluate the realized residual generators and remove all without useful properties. The residuals are evaluated according to their ability to detect a bias fault in sensor y_1 and in a stability sense. According to the power function in Figure 7.8, both residual R_1 and R_3 have the ability to detect a bias fault in sensor y_1. The fault free simulation in Figure 7.9 shows that residual R_2 does not converge to zero and is therefore rejected. The resulting residual generators are R_1 and R_3.

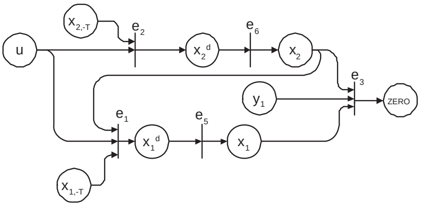

Figure 7.7. Oriented graph for residual generator R_3 yield by matching Γ_3 and the structural matrix in Table 7.8, where $x_{i,-T}$ symbolize the variable x_i in the previous sample.

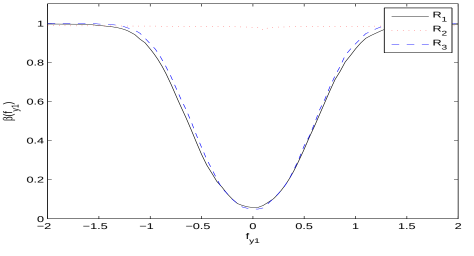

Figure 7.8. Detectability evaluation of residual R_1, R_2 and R_3.

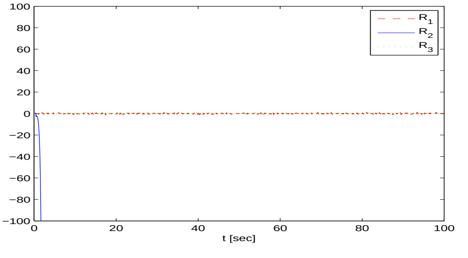

Figure 7.9. Stability evaluation of residual R_1, R_2 and R_3.

Part III

Estimating Derivatives and Evaluation of Methods

Chapter 8

Realizing Consistency Relations

Time derivatives of signals are in Chapter 6 and 7 used in residual generators and they are in general not known. Hence, it is either necessary to have methods that estimate the derivatives or realize the residual generator in a way that derivatives of signals are avoided. In this chapter different methods to estimate derivatives and realize consistency relations are presented and discussed.

8.1 Methods for Estimating Derivatives

Numerical differentiation in a noisy environment is an ill-posed problem in the sense that small measurement errors may cause large estimation errors. In this section, three different methods to estimate derivatives are presented and evaluated.

8.1.1 Approximately Differentiating Filter

A basic and straightforward method to estimate derivatives is to use an approximately differentiating filter and the most simple is, see [8]

$$\dot{y} \approx \frac{s}{1 + sT_d} y. \tag{8.1}$$

This is a first order low-pass differentiating filter with T_d as a design parameter. This filter is a good approximation of the derivative for signals with low frequencies in relation to the noise frequencies but is not for signals with high frequencies in relation to the noise frequencies. More advanced filter theory can be applied in order to get a better result.

If information about the noise and signal frequencies exists, a filter, $H(s)$, that passes the signal frequencies and attenuates the noise frequencies can be used to derive a differentiating filter, $D(s)$, as

$$D(s) = sH(s). \tag{8.2}$$

The filter, $H(s)$, needs to be flat in the passband in order to gain good differentiation capacity in the passband and therefore a Butterworth filter is used since the passband is flat, see [20].

Design Parameter

When an approximately differentiating filter is used to estimate derivatives the cut-off frequencies for the filter and the order of the filter have to be decided, which means that good knowledge about the signals is needed.

8.1.2 Smoothing Spline Approximation

Another method to estimate derivative is to use smoothing spline approximation, see [10]. In this approach analytical functions are used to approximate the measured signal. The derivative is then computed analytically from the analytical function. This method is adjusted to a smoothing spline approximation on a sliding window in this section.

Let the sliding window consist of n equidistant measurement points $t_1, t_2, \ldots, t_n$ such that $t_i = t_0 + Ti$ and let k be an integer satisfying $2 \leq k \leq n$, where k is the order of the smoothing spline. Assume that $y(t)$ is a noisy measurement of the function $x(t)$ and that there exists a δ, called noise level, satisfying

$$\frac{1}{n}\sum_{i=1}^{n}(y(t_i) - x(t_i))^2 \leq \delta^2. \tag{8.3}$$

The noise level δ can be interpreted as the deviation of the measurement noise, σ_v. The main objective with this method is to find a smooth function $f(t)$ such that the derivative $f^{(j)}(t)$ approximates the function $x^{(j)}(t)$ where $1 \leq j \leq k-1$ is a positive integer. The problem can be written as a minimizing problem to be solved in every sampling point.

$$\min_f \Phi_k(f) = \min_f \frac{1}{n}\sum_{i=1}^{n}(f(t_i) - y(t_i))^2 + \alpha \parallel f^k \parallel^2_{L^2(\mathbb{R})}, \tag{8.4}$$

where α is a regularization parameter. The minimizing problem, (8.4), can be separated in two sub-problems:

(1) Find a regularization parameter α.
(2) Find a minimizer f of the minimizing problem (8.4).

Consider the first problem, two methods to find the regularization parameter α exist. One priori choice method and one posteriori method, see [10]. In this thesis the priori choice method will be used to find the regularization parameter α so it can be used to calculate the coefficient vectors c and d in (8.9). The priori choice strategy takes $\alpha = \delta^2 \approx \sigma_v^2$.

Consider the second problem defined above, which is to find a minimizer f. Denote

$$f_\alpha(t) = \sum_{j=1}^{n} c_j |t - t_j|^{2k-1} + \sum_{j=1}^{k} d_j t^{j-1} \tag{8.5}$$

where coefficients $\{c_j\}_1^n$ and $\{d_j\}_1^n$ satisfy

$$f_\alpha(t_i) + 2(2k-1)!(-1)^k \alpha n c_i = y(t_i), \qquad i = 1, \ldots, n \tag{8.6a}$$

$$\sum_{j=1}^{n} c_j t_j^i = 0, \qquad i = 0, \ldots, k-1. \tag{8.6b}$$

The problem to find a minimizer f is actually a problem to find the coefficients for the polynomial in (8.5). The linear system of equations (8.6) can be written as

$$\begin{pmatrix} A + (-1)^k D & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix} \tag{8.7}$$

where $c = (c_1, c_2, \ldots, c_n)^T$, $d = (d_1, d_2, \ldots, d_k)^T$, $A = (|t_i - t_j|^{2k-1})_{n \times n}$, $D = 2(2k-1)! \alpha n I_{n \times n}$, $y = (y(t_1), y(t_2), \ldots, y(t_n))^T$ and P is a Vandermonde matrix

$$P = \begin{pmatrix} 1 & t_1 & \ldots & t_1^{k-1} \\ 1 & t_2 & \ldots & t_2^{k-1} \\ \ldots & \ldots & \ldots & \ldots \\ 1 & t_n & \ldots & t_n^{k-1} \end{pmatrix}.$$

Then f_α is a unique solution to the minimizing problem (8.4) and it exists a unique solution for the linear system of equations (8.7), see [10].

Assume that $t_0 = 0$ then the matrices P and A can be rewritten as $A = (|T(i-j)|^{2k-1})_{n \times n}$ and

$$P = \begin{pmatrix} 1 & T & \ldots & T^{k-1} \\ 1 & 2T & \ldots & (2T)^{k-1} \\ \ldots & \ldots & \ldots & \ldots \\ 1 & nT & \ldots & (nT)^{k-1} \end{pmatrix}.$$

If

$$\begin{pmatrix} A + (-1)^k D & P \\ P^T & 0 \end{pmatrix} \tag{8.8}$$

is invertible the coefficient-vectors c and d can be calculated as a function of the measurement data.

$$\begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} A + (-1)^k D & P \\ P^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} y \\ 0 \end{pmatrix} \tag{8.9}$$

The determinant of matrix (8.8) is not equal to zero, which means that it is invertible.

$$det \begin{pmatrix} A + (-1)^k D & P \\ P^T & 0 \end{pmatrix} = det(P^T) det(P) = det(P)^2 = (\prod_{1 \le i < j \le n} (T(j-i)))^2 \neq 0 \tag{8.10}$$

The derivative can be computed as

$$\frac{f_\alpha}{dt}(t) = (2k-1) \sum_{j=1}^{n} c_j |t - t_j|^{2k-2} + \sum_{j=2}^{k} (j-1) d_j (t - t_0)^{j-2}. \tag{8.11}$$

Example 8.1

Consider the trigonometric function

$$y(t) = \sin t. \tag{8.12}$$

The Maclaurin series expansion of $\sin(t)$ is

$$\sin(t) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} t^{2n+1} \quad = t - \frac{t^3}{3!} + \frac{t^5}{5!} \cdots. \tag{8.13}$$

If smoothing spline is used to estimate $y(t)$ the coefficient-vector d should have the same characteristic as the Maclaurin expansion coefficients. Smoothing spline on a window with the length of one period and $k = 8$ yields

$$\begin{aligned} f(t) = {} & 0.000018 + 1t + 0.0018t^2 - 0.171t^3 + 0.0048t^4 + 0.0050t^5 \\ & + 0.0014t^6 - 0.00058t^7, \end{aligned} \tag{8.14a}$$

which not is far from (8.13)

$$\sin(t) \approx t - 0.17t^3 + 0.0083t^5 - 0.0002t^7 \tag{8.14b}$$

Design Parameter

With smoothing spline approximation three design parameters are to be tuned.

σ_v^2	The variance of the measurement noise
k	The smoothing parameter
n	The length of the sliding window

8.1.3 Kalman Filter

The Kalman filter is a well studied method and it is used to estimate states for a dynamical system, see Section 2.3.1. If an integrator is used as the dynamical system a Kalman filter can be used to estimate the states of the system, which means the derivative of the measured signal.

A state-space model for a triple integrator can be described as

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u \tag{8.15a}$$

$$y = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} x, \tag{8.15b}$$

with u as input signal and y as output. Suppose that y is measured with added noise v that can be described as white noise with standard deviation σ_v and input u in some sense can be described as white noise with standard deviation σ_u. Let $u = w$ then (8.15) can be written on the form

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} w \tag{8.16a}$$

$$y = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} x + v. \tag{8.16b}$$

This model yields the state-vector

$$x = \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \end{pmatrix}.$$

If a Kalman filter is used in order to estimate the state-vector x, an estimator for the measured signal y and its first and second order derivatives have been created. The transfer function for the steady-state Kalman filter, with cross-covariance $S = 0$, from input y to its derivative $\dot{y}$ is

$$G_{\dot{y}y}(s) = \frac{2(Q/R)^{1/3}s^2 + (Q/R)^{1/2}s}{s^3 + 2(Q/R)^{1/6}s^2 + 2(Q/R)^{1/3}s + (Q/R)^{1/2}} \tag{8.17}$$

Design Parameters

When using the Kalman filter as an estimator of the derivative of the measured signal there are three design parameters to tune, Q, R and S. Q and R can be interpreted as one parameter because it is only the quotient Q/R that affect the filter, see transfer function (8.17). The cross-covariance S is zero if w and v both are Gaussian variables. Now this three design parameters can be reduced to one, the quotient Q/R.

Q/R The quotient between process noise and measure noise

8.1.4 Evaluation of the Estimating Methods

The trade-off between noise reduction and time delay has to be considered when estimating derivatives from a noisy environment. The connections between this trade-off and the design parameters for the different estimation methods are described below.

Approximately differentiating filter

The cut-off frequencies for the filter have to be decided in a way that signal frequencies passes and noise frequencies attenuates. The choice of filter order yields different phase-characteristic which gives a time-delay. A filter of higher order can attenuates noise better but yields a longer time-delay.

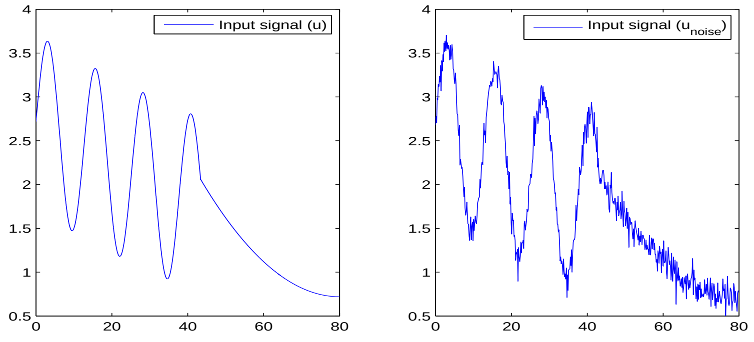

Figure 8.1. Signal with and without noise.

Smoothing Spline Approximation

Both k and n can be used to adjust the trade-off. The window length, n, adjust the time-delay and the smoothing parameter, k, adjust the noise reduction.

Kalman Filter

The quotient Q/R is the direct trade-off between noise reduction and time-delay. High quotient yields good noise reduction but long time-delay, on the other hand low quotient gives short time-delay but bad noise reduction.

In this thesis the estimated derivatives are used in residual generators based on consistency relations. In other words, if the derivatives are time delayed, then all other signals in the consistency relation have to be delayed for the consistency to hold. Two ranking quantities to evaluate an estimate, $\dot{f}(t)$, of a signal derivative, $\dot{y}(t)$, are defined as follow.

Definition 8.1 *The time delay, τ, of a estimation and the square estimation error, SEE, with the time delay in mind are defined as*

$$SEE = \frac{1}{n}\sum_{i=1}^{n}(\dot{f}(t_i) - \dot{y}(t_i - \tau))^2 \tag{8.18a}$$

$$\tau = \arg\min_{\tau} SEE. \tag{8.18b}$$

When comparing different estimation methods to each other, one estimation method is better than another if it have smaller τ and SEE. The signal derivative, $\dot{y}(t)$, is usually not known and therefore a signal with known derivative is used in order to get an evaluation about the estimation methods.

Figure 8.1 present a signal that is used to evaluate the different methods presented in this chapter to estimate derivatives. To get some information on how

Estimate	τ [s]	SEE
Kalman	0.45	0.019
$F_1(s)$	0.45	0.066
$F_2(2)$	0.45	0.067
Spline	0.45	0.006

Table 8.1. Ranking quantities for four different estimation methods, with focus on short time-delay.

Estimate	τ [s]	SEE
Kalman	0.45	0.014
$F_1(s)$	1.45	0.002
$F_2(2)$	1.7	0.018
Spline	0.95	0.001

Table 8.2. Ranking quantities for four different estimation methods, with focus on noise reduction.

the noise affects the different estimation methods, Gaussian noise with suitable variance is added to the signal.

In order to evaluate the different methods two simulations were made in MATLAB/SIMULINK where the derivative was estimated with 2 different approximately differentiating filter (Section 8.1.1), smoothing spline (Section 8.1.2) and with a Kalman filter (Section 8.1.3). The approximately differentiating filters are

$$F_1(s) = \frac{s}{(T_{d_1}s + 1)^2} \tag{8.19a}$$

$$F_2(s) = \frac{s}{(T_{d_2}s + 1)}. \tag{8.19b}$$

The design parameters for the different estimation methods, were tuned ad-hoc to get satisfying results, one simulation with focus on short time-delay and one with focus on good noise reduction. In Figure 8.2 the results from the two simulations, where the derivative was estimated with four different methods, are presented together with the analytical derivative.

Consider the estimations presented in Figure 8.2, the ranking quantities, τ and SEE, can be calculated and analyzed for every estimation methods for the two simulations to get a ranking among the different estimation methods.

According to the ranking quantities presented in Table 8.1 and 8.2 smoothing spline has best SEE when the estimation methods have been tuned to get a short time-delay. When the estimation methods have been tuned to get a good noise reduction has smoothing spline shortest time-delay and also lowest SEE. Hence, smoothing spline is the best estimation method according to these two properties.

Methods to generate residual generators using derivatives are presented in Part II. Since derivatives that may contain other derivatives are used, higher order derivatives have to be estimated. This leads to that estimates of derivatives

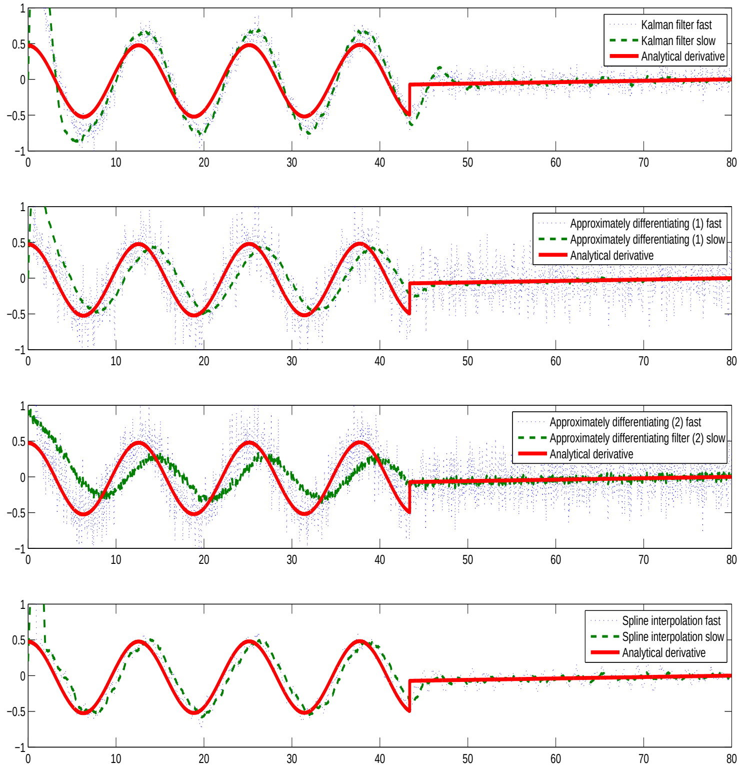

Figure 8.2. The derivative of signal estimated with three different methods.

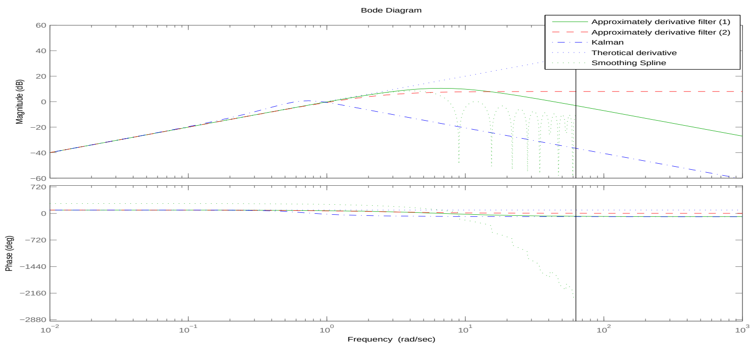

Figure 8.3. Bode plot of the evaluated methods.

of different order have different time-delays and all signals have to be delayed so all signals in the residual generator gets the same time-delay. It is important to keep the time-delay, τ, small, because it is desirable to detect a fault as soon as it occurs.

A bode plot for all estimation methods, with the parameters used above to get the ranking quantities in Table 8.1, is presented in Figure 8.3. The breakpoints for the different methods can be moved along the theoretical derivative if the design parameters for each filter are changed. It can be seen that filter F_1, F_2 and smoothing spline have the most accurate bode plot and F_1, smoothing spline and the Kalman filter have the best reduction for higher frequencies.

The conclusion that can be made from the evaluation presented in this section is that smoothing spline is the best method to estimate derivatives. Hence, smoothing spline is used in the sequel to estimate derivatives.

8.2 Realize in State-Space Form

This aim with this section is to present a method to avoid the need of numerical estimation of derivatives. The main idea is to add dynamics to the consistency relation in a way that a state-space realization without derivatives as inputs exists. Conditions for when this realization exists and how it can be done are presented and discussed.

8.2.1 Linear Systems

Given a linear system a residual generator based on a consistency relation, with the highest order of derivative $n-1$, can be written as

$$R = \sum_{i=1}^{n} C_i u^{(n-i)} + \sum_{i=1}^{n} D_i y^{(n-i)}, \tag{8.20}$$

where C_i and D_i are row-vectors. Let $\dot{u} = pu$, where p is the derivative operator, and add stable dynamics, $d(p)$, to the residual generator. In other words let $d(p) = p^n + a_1 p^{n-1} + \ldots + a_{n-1}p + a_n$ be a polynomial with all roots in the open left half plane. Then (8.20) can be written as

$$R = \frac{1}{d(p)} \left(\sum_{i=1}^{n} C_i p^{n-i} \right) u + \left(\sum_{i=1}^{n} D_i p^{n-i} \right) y = G_{ru}(p)u + G_{ry}(p)y, \qquad (8.21)$$

where $G_{ru}(p)$ is the transfer function from u to r and $G_{ry}(p)$ is the transfer function from y to r. The polynomial, $d(p)$, can be chosen such that $d(p)$ have no roots in common with the numerators in the polynomial matrices $G_{ru}(p)$ or $G_{ry}(p)$. This can be done since there exist only a finite number of roots in the numerators and there exists infinite numbers of choices for the roots to $d(p)$. It can be proved that a minimal realization of input-output behavior (8.21), with the characteristic described above, have dimension n, see [13]. The system (8.21) can be realized in observer companion form

$$\dot{x} = \begin{pmatrix} -a_1 & 1 & 0 & \ldots & 0 \\ -a_2 & 0 & 1 & 0 & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ -a_{n-1} & 0 & \ldots & 0 & 1 \\ -a_n & 0 & \ldots & 0 & 0 \end{pmatrix} x + \begin{pmatrix} C_1 & D_1 \\ C_2 & D_2 \\ \vdots & \vdots \\ C_{n-1} & D_{n-1} \\ C_n & D_n \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix} \qquad (8.22a)$$

$$R = \begin{pmatrix} 1 & 0 & \ldots & 0 \end{pmatrix} x, \qquad (8.22b)$$

which have dimension n and therefore is a minimal realization of the input-output behavior (8.21).

8.2.2 Non-Linear Systems

For non-linear systems it is not easy to determine the conditions for when it is possible to realize a non-linear consistency relation in state-space form, see [12]. In certain cases a possible approach to handle non-linear consistency relations is to find a variable substitution such that the non-linear problem is transformed to a linear problem. When the problem is transformed to linear problem the methods described in Section 8.2.1 can be used.

Example 8.2

Consider the non-linear consistency relation

$$c(\dot{y}, y) = 2y_1\dot{y}_1 + y_2. \qquad (8.23)$$

This consistency relation is not straightforward to realize in state-space form but since $\frac{d}{dt}(y_1^2) = 2y_1\dot{y}_1$, the consistency relation can be written in linear form as

$$c(\dot{y}, y) = \frac{d}{dt}(y_1^2) + y_2 = \dot{v} + y_2, \qquad (8.24)$$

with variable substitution $v = y_1^2$. The method for the linear case can now be applied.

The problem to find a suitable variable substitution in a way that the non-linear consistency relation is transformed to a linear problem, as in Example 8.2, is not a trivial problem. Furthermore, the existence of such a variable substitution is not obvious, see [12].

In general the problem can be described as to find two transformations, $\Psi(c(\dot{y}, y), y)$ and $\psi(r, y)$, and residual dynamics, $h(r, y)$, such that there exist a state-space realization

$$\dot{\omega} = f(\omega, y) \tag{8.25a}$$

$$r = \psi^{-1}(\omega, y), \tag{8.25b}$$

with the variable substitution $\omega = \psi(r, y)$, for the residual generator

$$\dot{r} + h(r, y) = \Psi(c(\dot{y}, y), y). \tag{8.26}$$

There exist several conditions on the consistency relation, $c(\dot{y}, y)$, for when it is possible to find $\Psi(c(\dot{y}, y), y)$, $\psi(r, y)$ and $h(r, y)$ such that it exists a state-space realization (8.25), see [12]. However, this is beyond the scope of this thesis.

8.3 A Comparison Between State-Space Realization and Estimation of Derivatives

This section contains a comparison, for linear system, between the methods to estimate derivatives, described in Section 8.1, and the method to realize the consistency relation in state-space form, described in Section 8.2.

Example 8.3

Consider the linear consistency relation $c(\dot{y}, y) = \dot{y} + y$. Assume that the derivative is estimated with a differentiating filter

$$D(p) = pH(p) = \frac{p}{(pT_d + 1)^2}. \tag{8.27}$$

This yields a residual generator

$$R = \frac{p}{(pT_d + 1)^2} y + y. \tag{8.28}$$

If y in eqution (8.28) is filtered with $H(p)$, then residual generator (8.28) have the same form as if the residual generator were realized in state-space form with the same dynamic as the filter $H(p)$. That is, there is no difference in the linear case between a residual generator realized in state-space form and a residual generator where derivatives are estimated with an approximately differentiating filter and all other signals are filtered.

In Example 8.3 the differences between a state-space realization and a realization

	Fault	Fault free
f_u	Bias-error in actuator	0
f_{y1}	Bias-error in sensor 1	0
f_{gy1}	Gain-error in sensor 1	1
f_{y2}	Bias-error in sensor 2	0

Table 8.3. Table describing the fault implemented in system (8.29) and their values in the fault free case.

when derivatives are estimated is presented. The connection between the polynomial $d(p)$ and the differentiating filter's design parameter is also illustrated. In other words, when dynamics are added to the residual generator, all zeros to the polynomial, $d(p)$, can be chosen in the same way as the design parameters for the approximately differentiating filter.

Consider the linear system described in Example 6.2 with faults implemented as

$$e_1 : \dot{x}_1 = -x_1 - 5x_2 + u + f_u \tag{8.29a}$$
$$e_2 : \dot{x}_2 = -x_2 + u + f_u \tag{8.29b}$$
$$e_3 : y_1 = f_{gy1}(x_1 + x_2) + f_{y1} \tag{8.29c}$$
$$e_4 : y_2 = x_2 + f_{y2}, \tag{8.29d}$$

where the faults f_i are described in Table 8.3. Measurement noise were added to the two sensors y_1 and y_2 during the simulation. The residual generators from Example 6.2 are given by

$$R_1 = \dot{y}_2 + y_2 - u \tag{8.30a}$$
$$R_2 = \dot{y}_1 - \dot{y}_2 + y_1 + 4y_2 - u. \tag{8.30b}$$

Residual generator (8.30b) was realized according to the method presented in Section 8.2.1. This was simulated in Matlab/Simulink along with the same residual generator where derivatives were estimated and all other signals in the residual generator were filtered with the same characteristic as the differentiating filter. The results from the simulation are presented in Figures 8.4 and 8.5 as power functions of the test, $T(z)$. The test $T(z)$ is based on the residual generator R_2 with gain-error and bias-error on sensor y_1. The threshold, J, for the test was chosen such that the probability of $|T(z)| < J$ in the fault free case is 95%. It can be seen in Figures 8.4 and 8.5 that the different power functions have approximately the same characteristic, as it was shown in Example 8.3. For non-linear consistency relations there are several conditions for when a state-space realization exists and therefore this approach is not considered as useful for the purpose of this thesis and hence not further investigated.

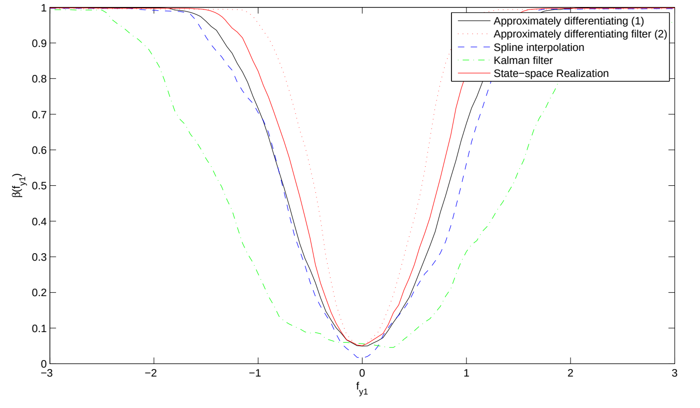

Figure 8.4. Power functions for residual generator R_2 with various bias-errors on sensor y_1.

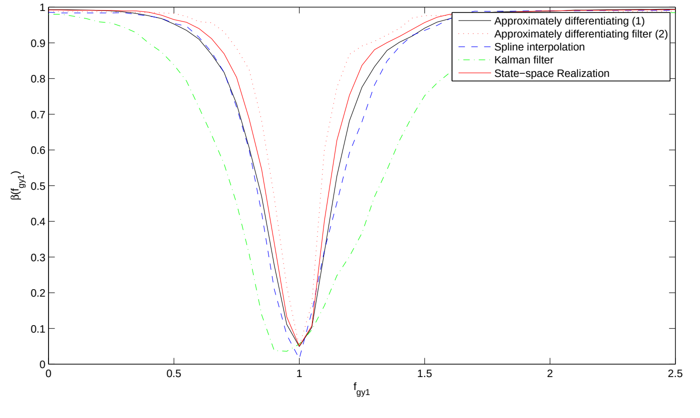

Figure 8.5. Power functions for residual generator R_2 with various gain-errors on sensor y_1.

8.4 Discussion

The parameters for the different estimations methods are easier to choose if there exists good knowledge about the signal that shall be differentiated.

Conclusions about how good an estimation method is, are very hard to decide. It depends on what properties that define a good estimation. The different estimation methods are evaluated in a noise sensitive sense and the time-delay is also taken in to consideration in this chapter. Since estimated derivatives are used in residual generators in this thesis, the behavior of residual generators is an important property to investigate.

To get information on how good the estimation methods are to estimate derivatives from real measurement data, different residual generators where different signals are differentiated are needed. In the next chapter, a Scania diesel engine model is used to extract residual generators and one residual generator based on differentiation is presented. Further investigations about the estimation methods can be made from the behavior of the residual generator.

Chapter 9

Evaluation of the Residual Generation Methods

The methods to generate residual generators, see Part II, were compared and evaluated and the results are presented in this chapter. From which MSO sets it was possible to construct residual generators with the different methods was analyzed. But also how residual generators with estimated derivatives work, with measured signals from a real system, in this thesis a Scania engine.

9.1 Comparison of the Methods on a Satellite System

The different methods to generate residual generators were compared on a small satellite system. The intention with this comparison was to see from which MSO sets a residual generator were created and design a diagnosis system were two faults could be detected and isolated from each other. Consider the point-mass satellite model

$$e1: \dot{r} = v \tag{9.1a}$$

$$e2: \dot{v} = r\omega - \theta_1 \frac{1}{r^2} + (\theta_2 + f_{u_1})u_1 \tag{9.1b}$$

$$e3: \dot{\phi} = \omega \tag{9.1c}$$

$$e4: \dot{\omega} = -\frac{2v\omega}{r} + \theta_2 \frac{u_2}{r} \tag{9.1d}$$

$$e5: y_1 = r + f_{y_1} \tag{9.1e}$$

$$e6: y_2 = \phi \tag{9.1f}$$

$$e7: y_3 = \omega, \tag{9.1g}$$

where (r, ϕ) is the position of the satellite in polar coordinates in the plane, v radial velocity, ω angular velocity, u_1, u_2 radial and tangential thrust respectively and θ_1, θ_2 known constants, see [16]. The structural matrix for the satellite system is seen in Table 9.1. Equation e_4 is considered non-invertible for r and ω. The variable r can not be decided due to $\dot{\omega}$ may become zero, which would lead to a division by zero. The variable ω can not be decided because the radial velocity v may become zero. Even in this case it would lead to a division by zero. The two faults that the diagnosis system was designed to detect and isolate were an actuator fault, f_{u_1}, and a sensor fault, f_{y_1}. There were ten different MSO sets found in this system. All MSO sets and which equations the MSO sets consist of are seen in Table 9.2.

eq	r^d	v^d	ϕ^d	ω^d	r	v	ϕ	ω
e_1	×					×		
e_2		×			×			×
e_3			×					×
e_4				×	Δ	×		Δ
e_5					×			
e_6							×	
e_7								×
d_1	×				×			
d_2		×				×		
d_3			×				×	
d_4				×				×

Table 9.1. Structural matrix for system (9.1).

MSO	eq
1	e_3, e_6, e_7, d_2
2	$e_1, e_2, e_4, e_5, d_1, d_2, d_4$
3	$e_1, e_2, e_4, e_7, d_1, d_2, d_4$
4	$e_1, e_4, e_5, e_7, d_1, d_4$
5	$e_2, e_4, e_5, e_7, d_2, d_4$
6	$e_1, e_2, e_5, e_7, d_1, d_2$
7	$e_2, e_3, e_4, e_5, e_6, d_2, d_3, d_4$
8	$e_1, e_3, e_4, e_5, e_6, d_1, d_3, d_4$
9	$e_1, e_2, e_3, e_4, e_6, d_1, d_2, d_3, d_4$
10	$e_1, e_2, e_3, e_5, e_6, d_1, d_2, d_3$

Table 9.2. MSO sets found in system (9.1).

To achieve best possible fault isolation for a system, a residual generator from each MSO set is needed, see Section 4.4. In Table 9.3 it is shown for which MSO sets it was possible to design residual generators with the different methods.

Residual generators from seven out of the ten MSO sets were found with integral causality. Hence, better fault detectability and isolation could be achieved if

MSO	IC	DC	MC
1	Yes	Yes	Yes
2	Yes	Yes	Yes
3	Yes	-	Yes
4	-	Yes	Yes
5	Yes	Yes	Yes
6	Yes	Yes	Yes
7	Yes	Yes	Yes
8	-	Yes	Yes
9	Yes	-	Yes
10	-	Yes	Yes

Table 9.3. Found residual generators with different methods, integral causality (IC), derivative causality (DC) and mixed causality (MC), for each MSO set.

residual generators from the remaining MSO sets were found. Neither with derivative causality were residual generators from all MSO sets found. Only from eight out of the ten MSO sets were residual generators derived. However, with mixed causality residual generators from all MSO sets were found.

The fault detectability for each found residual generator is seen in Table 9.4. With all residual generators that were constructed with integral causality, the fault f_{y_1} could not be isolated from the fault f_{u_1}. With derivative causality, the fault f_{u_1} could not be isolated from the fault f_{y_1}. However, with mixed causality were a residual generator from each MSO set constructed, which yielded that the fault f_{y_1} could be isolated from the fault f_{u_1} and vice versa.

(a)

	f_{u_1}	f_{y_1}
R_1		
R_2	X	X
R_3	X	
R_5	X	X
R_6	X	X
R_7	X	X
R_9	X	

(b)

	f_{u_1}	f_{y_1}
R_1		
R_2	X	X
R_4		X
R_5	X	X
R_6	X	X
R_7	X	X
R_8		X
R_{10}	X	X

(c)

	f_{u_1}	f_{y_1}
R_1		
R_2	X	X
R_3	X	
R_4		X
R_5	X	X
R_6	X	X
R_7	X	X
R_8		X
R_9	X	
R_{10}	X	X

Table 9.4. (a) Fault detectability with IC. (b) Fault detectability with DC. (c) Fault detectability with MC.

This comparison shows that mixed causality is better than derivative and integral causality separately to find residual generators. This is because the union of all residual generators found with integral and derivative causality is also found with mixed causality.

9.2 Evaluation of the Residual Generation Methods on a Scania Diesel Engine

To compare the methods on a more complex system than the satellite system, a comparison was made on a Scania diesel engine. One residual generator, where derivatives are used, was evaluated to see if the residual generator could detect faults even when derivatives were estimated.

9.2.1 Engine Model

The model of the Scania diesel engine used in this thesis is developed at the Division of Vehicular Systems, Linköpings University, see [15].

To get the model compatibly with the methods used to find residual generators, some modifications have been made to the original model. The modifications include two new states, T_1 and x_r, which are added to get the model in state-space form. The modified model is on the form

$$\dot{x} = f(x, u_c, u_m) \tag{9.2a}$$

$$y = h(x), \tag{9.2b}$$

where u_c is the control input vector

$$u_c = (u_\delta, u_{egr}, u_{vgt})^T \tag{9.2c}$$

and u_m an input vector with measured variables

$$u_m = (n_e, T_{im}, T_{amb}, p_{amb})^T. \tag{9.2d}$$

The state vector x is

$$x = (p_{im}, p_{em}, X_{Oim}, X_{Oem}, \omega_t, \tilde{u}_{vgt}, T_1, x_r)^T, \tag{9.2e}$$

and the output vector y

$$y = (w_{cmp}, p_{im}, p_{em}, n_{trb})^T. \tag{9.2f}$$

All nomenclature used above to describe the model are explained in Table 9.5. One of the input signals used in u_c are not the actual control signal, instead of the EGR control signal the measured position of the EGR valve is used. However, u_δ and u_{vgt} are the control signals.

In Figure 9.1 a Matlab/Simulink implementation of the model is shown.

Validation of the Modified Model

To verify that the modifications made in the model (9.2) are correct, a validation of the modified model was performed. Same input signals, u_c and u_m, were used

Variable	Description	Unit
u_δ	Mass of injected fuel	$mg/cycle$
u_{egr}	EGR control signal	$\%$
u_{vgt}	VGT control signal	$\%$
n_e	Engine rotational speed	rpm
T_{im}	Intake manifold temperature	K
T_{amb}	Ambient temperature	K
p_{amb}	Ambient pressure	Pa
p_{im}	Intake manifold pressure	Pa
p_{em}	Exhaust manifold pressure	Pa
X_{Oim}	Intake manifold oxygen concentration	$-$
X_{Oem}	Exhaust manifold oxygen concentration	$-$
ω_t	Turbine rotational speed	rad/s
$\tilde{u}_{vgt}$	VGT valve position	$-$
T_1	Cylinder temperature after intake stroke	K
x_r	Residual gas fraction	$-$
w_{cmp}	Compressor mass flow	kg/s
n_{trb}	Turbine rotational speed	rpm

Table 9.5. Nomenclature for the Scania engine model (9.2).

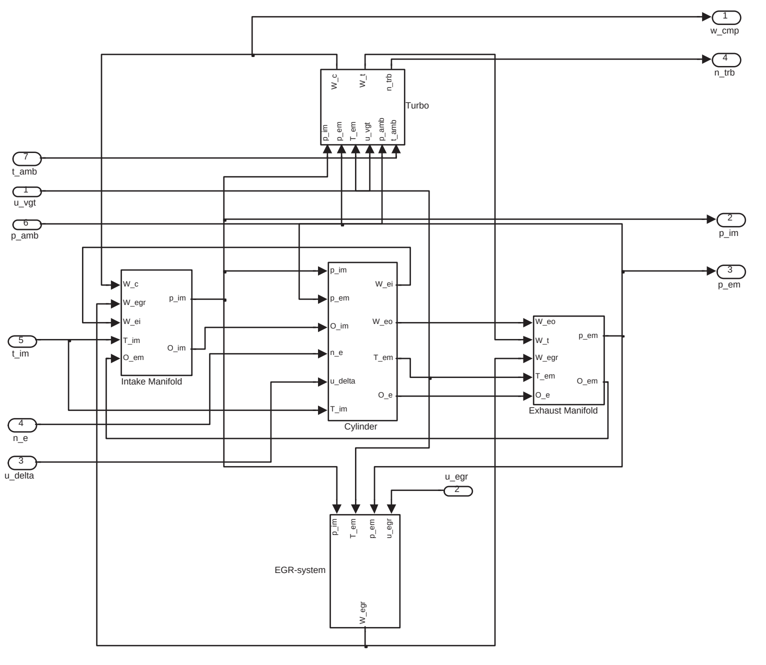

Figure 9.1. A MATLAB/SIMULINK implementation of the model (9.2).

to both the original model and the modified model and the outputs, y, from both models were compared. The relative errors between these outputs were used to validate the modified model and the relative error is calculated as

$$E(i) = \frac{y_{original}(i) - y_{modified}(i)}{\frac{1}{N}\sum_{i=1}^{N} y_{original}(i)}. \tag{9.3}$$

How the relative errors vary over time is seen in Figure 9.2. The mean relative error and max relative error is also shown for every output. These were calculated without the first 20 seconds of the simulation. This is because the modified model is incorrect in the beginning of a simulation due to removed saturations.

The mean relative error for all outputs was between 0.2% and 0.36% and the maximum relative error for all outputs was between 0.9% and 2.4%. The relative errors were around a tenth of the relative errors that the original model has compared to the measured outputs from the engine, see [15], which justifies the modifications.

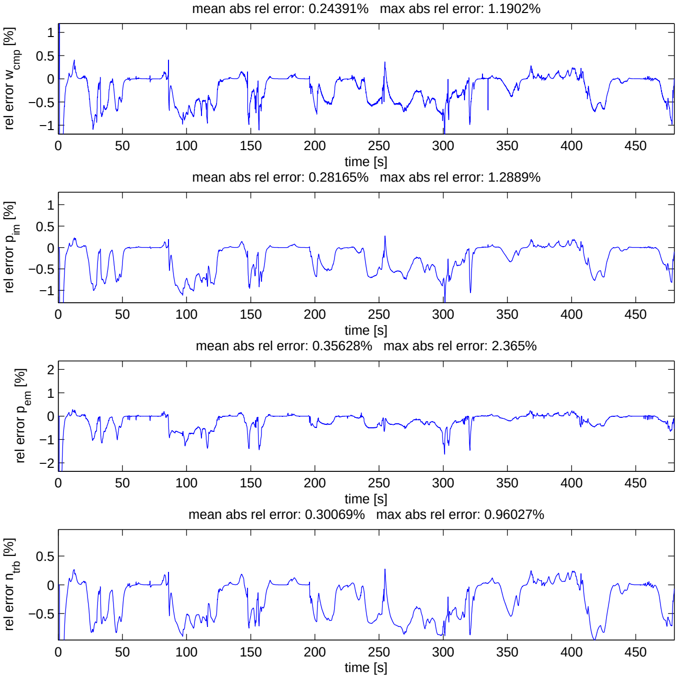

Figure 9.2. Validation of modified engine model. Relative error for w_{cmp} (top), p_{im} (second from top), p_{em} (second from bottom) and n_{trb} (bottom).

9.2.2 Comparison of the Different Methods to Generate Residual Generators

To evaluate the methods to generate residual generators from the engine model a MATLAB toolbox developed at Scania was used. The toolbox automatically generates residual generators. The toolbox is built on the work that has been done in [6], [5] and [3]. The toolbox handled integral causality and has therefore been upgraded to handle derivative causality and partly mixed causality.

To evaluate the methods, the number of residual generators from different MSO sets was compared. From the engine model the toolbox extracted 90 different MSO sets. With integral causality residual generators from four different MSO sets were found and with derivative causality it was only possible to find a residual generator from one MSO set. It was possible to find residual generators from 35 of the total 90 MSO sets with mixed causality. The MSO sets where residual generators were found, with the different methods, are seen in Table 9.6. An important thing to notice is that the union of residual generators found with integral and derivative causality is not equal to the set of residual generators found with mixed causality.

Residual Generators with Integral Causality

It was possible to generate residual generators from four different MSO sets with integral causality. The small number of residual generators was mainly due to many non-invertible equations. The constructed residual generators were

$$R_{IC_1} = f_1(n_{trb}, p_{amb}, p_{im}, T_{amb}, w_{cmp}) \tag{9.4a}$$

$$R_{IC_{43}} = f_2(u_\delta, n_e, n_{trb}, p_{amb}, p_{em}, p_{im}, T_{amb}, T_{im}, u_{vgt}) \tag{9.4b}$$

$$R_{IC_{76}} = f_3(u_\delta, n_e, p_{amb}, p_{em}, p_{im}, T_{amb}, T_{im}, u_{vgt}, w_{cmp}) \tag{9.4c}$$

$$R_{IC_{88}} = f_4(u_\delta, n_e, n_{trb}, p_{amb}, p_{em}, p_{im}, T_{amb}, T_{im}, u_{vgt}, w_{cmp}). \tag{9.4d}$$

Only the signals that the residual generators need is written since the residual generators contain between five and 40 equations. Residual generator (9.4a) is static.

Residual Generators with Derivative Causality

It was only possible to find one residual generator with derivative causality and that was the static residual generator (9.4a), which was also found with integral causality.

There are two main reasons that explain why no new residual generators were found with derivative causality.

The first reason is because there are many non-invertible equations in the model. These equations are of the type $x_1 = min(x_2, x_3)$, $x_1 = max(x_2, x_3)$ and $x_1 = saturate(x_2)$. If these equations are removed from the model, there are in total 598 MSO sets found. From these MSO sets it was possible to design 13 residual generators with derivative causality. In only two of these MSO sets were residual generators found with integral causality as well. Hence, derivative

MSO	IC	DC	MC
1	Yes	Yes	Yes
2	-	-	Yes
4	-	-	Yes
5	-	-	Yes
7	-	-	Yes
8	-	-	Yes
10	-	-	Yes
11	-	-	Yes
12	-	-	Yes
20	-	-	Yes
21	-	-	Yes
23	-	-	Yes
24	-	-	Yes
25	-	-	Yes
39	-	-	Yes
40	-	-	Yes
41	-	-	Yes
43	Yes	-	Yes
44	-	-	Yes
45	-	-	Yes
46	-	-	Yes
51	-	-	Yes
53	-	-	Yes
57	-	-	Yes
58	-	-	Yes
60	-	-	Yes
61	-	-	Yes
62	-	-	Yes
63	-	-	Yes
74	-	-	Yes
76	Yes	-	Yes
85	-	-	Yes
86	-	-	Yes
88	Yes	-	Yes
90	-	-	Yes

Table 9.6. Found residual generators with different methods, integral causality (IC), derivative causality (DC) and mixed causality (MC), for each MSO set.

causality would give a contribution to the overall fault detectability if all *min*, *max* and *saturate* equations were removed. However, modifications of that kind results in an incorrect model.

The second reason why there were no new residual generators found with derivative causality is because differential loops, see Section 6.4. Differential loops often occur in systems where there are cross dependencies between differential variables. Since no methods to handle differential loops with derivative causality are presented in this thesis, all MSO sets where differential loops occurred were discarded.

Residual Generators with Mixed Causality

In this section one residual generator that was found with mixed causality is shown and evaluated in a fault detectability and isolation sense.

The MSO set, from which the residual generator was derived, was chosen in a way that no residual generators with either integral or derivative causality could be found in that set. Another criterion for the MSO set was that it did not contain the signal t_{amb} but did contain the signal t_{im}. This yielded that it would be possible to isolate a fault affecting sensor t_{im} from a fault affecting sensor t_{amb}.

The structural matrix for the MSO set from which the residual generator was constructed is seen in Figure 9.3. The strongly connected components marked in the figure induce a differential loop which was solved with integral causality. Equation ($e37$) was used as residual equation.

The residual generator was

$$R_{MC_4} = f(u_\delta, n_e, n_{trb}, T_{im}, p_{amb}, p_{im}, p_{em}, w_{cmp}). \tag{9.5}$$

Based on the residual generator (9.5) a test was constructed. The length of the sliding test window was $N = 2000$ samples. The residual generator was simulated with measured sensor signals from a truck and with different faults implemented. The results of the simulations are seen in three different figures. In Figure 9.4, the result from the fault-free simulation is illustrated. In Figure 9.5, the result from the simulation where a bias fault in sensor T_{im} has occurred after $220s$ is presented. In Figure 9.6, the result from the simulation where a bias fault in sensor w_{cmp} has occurred after $200s$ is presented. The threshold was chosen such that the probability of $|T| < J$ in the fault free case was 99%.

The residual generator did only react for a limited time for the bias fault in T_{im}. There are two possible explanations for this. One explanation is that the residual generator only reacted for changes in T_{im} and the second is that the control loop compensated for the fault.

Derivatives from three signals, p_{im}, p_{em} and n_{trb}, was used in residual generator (9.5). Smoothing spline was used to estimate the derivatives and from the simulation results in Figures 9.4, 9.5 and 9.6 it is seen that the derivative estimation are satisfactorily correct since the tests reacted when the two different faults occurred and did not react in the fault-free case.

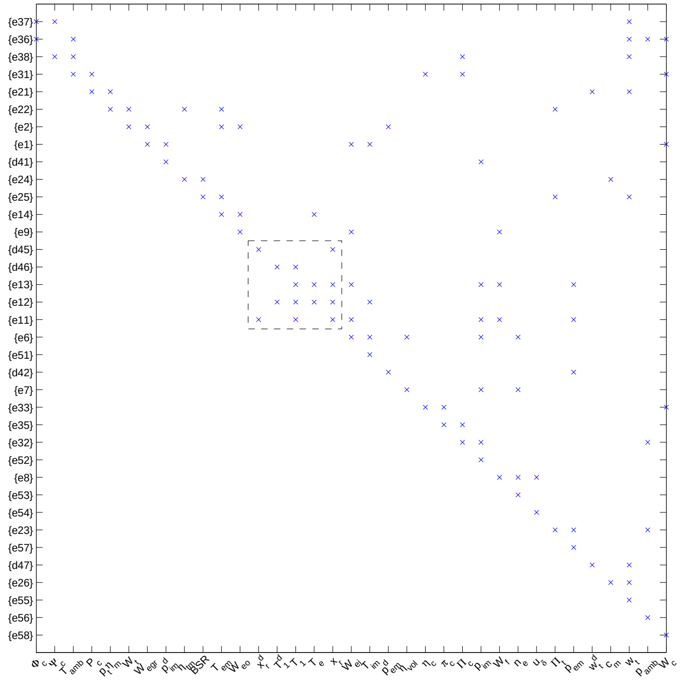

Figure 9.3. Structural matrix for MSO set 4 where residual generator (9.5) is found.

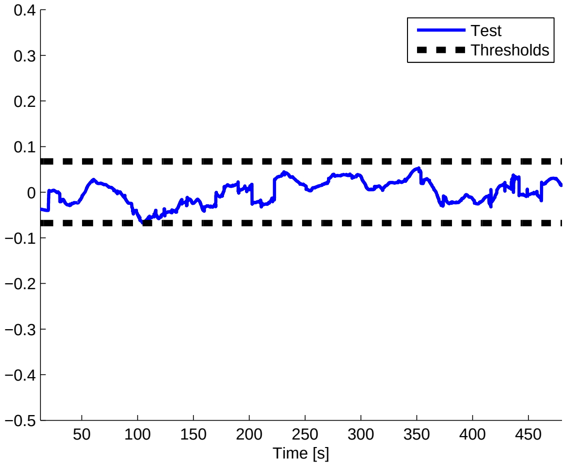

Figure 9.4. Fault-free simulation of residual generator (9.5).

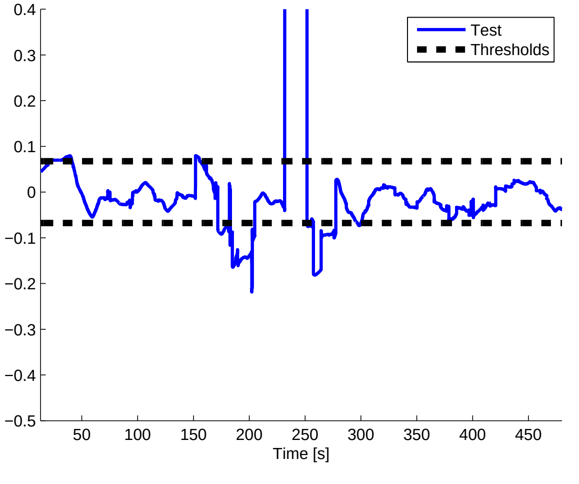

Figure 9.5. Simulation of residual generator (9.5) with a bias fault in sensor T_{im}.

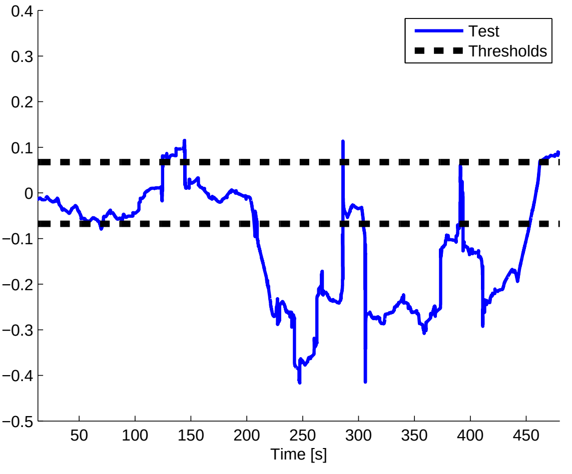

Figure 9.6. Simulation of residual generator (9.5) with a bias fault in sensor w_{cmp}.

The fact that the residual generator (9.5) reacted to the bias fault in T_{im} shows that it is useful. With this residual generator, a bias fault in T_{im} could be isolated from a fault in T_{amb}.

Fault Isolation Comparison

To see how much the fault isolation would improve if all found residual generators were realized, a fault isolation comparison was made. Two assumptions were made, the first was that all found residual generators were realizable. The second assumption was that only actuator and sensor faults were considered. That is, if a residual generator needs signal T_{im}, the residual generator is sensitive for faults in sensor T_{im}.

The fault detectability is seen in Table 9.7 for all residual generators constructed with integral causality and in Table 9.8 with mixed causality. From the fault detectability tables, it was possible to construct fault isolation matrices. The faults that actually occur are seen on the vertical axis and all possible faults that could explain the behavior are seen on the horizontal axis.

The fault isolation is seen in Table 9.9 for the diagnosis system designed with integral causality and in Table 9.10 with mixed causality. Faults in sensor u_{egr} have been left out because no residual generator is sensitive for faults in sensor u_{egr}.

The fault isolation with mixed causality has improved compared to the fault isolation with integral causality. For example, a fault in sensor p_{im} was previously a possible explanation for all possible faults but with mixed causality it is not. This shows that mixed causality is better in a fault isolation sense than integral causality and would give a contribution if it is used when designing a model-based diagnosis system.

	u_δ	n_{eng}	n_{trb}	p_{amb}	p_{em}	p_{im}	t_{amb}	t_{im}	u_{vgt}	w_{cmp}	u_{egr}
R_1			×	×		×	×			×	
R_{43}	×	×	×	×	×	×	×	×	×		
R_{76}	×	×		×	×	×	×	×	×	×	
R_{88}	×	×	×	×	×	×	×	×	×	×	

Table 9.7. Fault detectability with integral causality.

9.3 Discussion

The evaluation has shown that derivative causality does not work well when trying to find residual generators when the model consists of many non-invertible equations. However, with integral causality residual generators were found and when using these two methods separately, integral causality is considered better. By using mixed causality it was possible to generate residual generators from many more MSO sets than it was with integral and derivative causality separately. Even though many more residual generators were found with mixed causality, there are still only residual generators from 35 of 90 MSO sets found. This can partly be explained by the many non-invertible equations. Hence, if non-invertible equations are avoided as far as possible when models are constructed it would be easier to find more residual generators.

The realized residual generator with mixed causality has three estimated derivatives of measured signals as inputs. The results of the simulations of the residual generator show that the method to estimate derivatives is good enough in this application. Only the derivatives of three measurement signals are estimated, and in order to evaluate derivatives of other signals more residual generators, which need derivatives of other signals as input, are needed to be realized and analyzed.

	u_δ	n_{eng}	n_{trb}	p_{amb}	p_{em}	p_{im}	t_{amb}	t_{im}	u_{vgt}	w_{cmp}	u_{egr}
R_1			×	×		×	×			×	
R_2	×	×	×	×	×	×	×	×			
R_4	×	×	×	×	×	×		×	×		
R_5	×	×	×	×	×	×		×		×	
R_7	×	×	×	×	×		×	×	×		
R_8	×	×	×	×	×		×	×		×	
R_{10}	×	×	×	×	×			×	×	×	
R_{11}	×	×	×	×		×	×	×	×		
R_{12}	×	×	×	×		×		×	×	×	
R_{20}	×	×		×	×	×	×	×	×		
R_{21}	×	×		×	×	×	×	×		×	
R_{23}	×	×		×	×	×		×	×	×	
R_{24}	×	×		×	×		×	×	×	×	
R_{25}	×	×		×		×	×	×	×	×	
R_{39}	×	×	×	×	×	×	×	×	×		
R_{40}	×	×	×	×	×	×	×	×	×		
R_{41}	×	×	×	×	×	×	×	×	×		
R_{43}	×	×	×	×	×	×	×	×	×		
R_{44}	×	×	×	×	×	×	×	×	×		
R_{45}	×	×	×	×	×	×	×	×		×	
R_{46}	×	×	×	×	×	×	×	×		×	
R_{51}	×	×	×	×	×	×		×	×	×	
R_{53}	×	×	×	×	×	×		×	×	×	
R_{57}	×	×	×	×	×		×	×	×	×	
R_{58}	×	×	×	×	×		×	×	×	×	
R_{60}	×	×	×	×	×		×	×	×	×	
R_{61}	×	×	×	×	×		×	×	×	×	
R_{62}	×	×	×	×		×	×	×	×	×	
R_{63}	×	×	×	×		×	×	×	×	×	
R_{74}	×	×		×	×	×	×	×	×	×	
R_{76}	×	×		×	×	×	×	×	×	×	
R_{85}	×	×	×	×	×	×	×	×	×	×	
R_{86}	×	×	×	×	×	×	×	×	×	×	
R_{88}	×	×	×	×	×	×	×	×	×	×	
R_{90}	×	×	×	×	×	×	×	×	×	×	

Table 9.8. Fault detectability with mixed causality.

	u_δ	n_{eng}	n_{trb}	p_{amb}	p_{em}	p_{im}	t_{amb}	t_{im}	u_{vgt}	w_{cmp}
u_δ	×	×		×	×	×	×	×	×	
n_{eng}	×	×		×	×	×	×	×	×	
n_{trb}			×	×		×	×			
p_{amb}				×		×	×			
p_{em}	×	×		×	×	×	×	×	×	
p_{im}				×		×	×			
t_{amb}				×		×	×			
t_{im}	×	×		×	×	×	×	×	×	
u_{vgt}	×	×		×	×	×	×	×	×	
w_{cmp}				×		×	×			×

Table 9.9. Fault isolation with integral causality.

	u_δ	n_{eng}	n_{trb}	p_{amb}	p_{em}	p_{im}	t_{amb}	t_{im}	u_{vgt}	w_{cmp}
u_δ	×	×		×				×		
n_{eng}	×	×		×				×		
n_{trb}			×	×						
p_{amb}				×						
p_{em}	×	×		×	×			×		
p_{im}				×		×				
t_{amb}				×			×			
t_{im}	×	×		×				×		
u_{vgt}	×	×		×				×	×	
w_{cmp}				×						×

Table 10.10. Fault isolation with mixed causality.

Chapter 10

Conclusions and Further Work

10.1 Conclusions

The conclusions that have been made in this thesis can be summarized as:

- Three different methods to estimate derivatives were presented and from those methods it has been shown that smoothing spline approximation has the best properties for the purpose in this thesis. That is, to use estimated derivatives of known signals as inputs in residual generators.
- Using mixed causality when searching for residual generators gives a contribution to the number of MSO sets where residual generators can be realized, which improves the fault isolation compared to the existing method at Scania using integral causality.
- Using an extra equation to describe the connection between a variable and its derivative in structural models is necessary when mixed causality is used to generate residual generators.
- Purely differential loops can always be solved with integral causality. Loops with both differential and static equations can be solved with integral causality under certain conditions.

10.2 Further Work

There are some areas that may need further work in order to improve the methods to find realizable residual generators. These are:

- Design an engine model with as few as possible non-invertible equations would most likely increase the number of possible residual generators.

- Test the methods to estimate derivatives in a real engine for more residual generators to see how they behave in a real-time application.

Bibliography

[1] Blanke M. & Kinnaert M. &Lunze J. & Staroswiecki M. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag Berlin Heidelberg. 2003. ISBN 3-540-01056-4.

[2] Asratian A. & Denley T. & Häggkvist R. *Bipartite Graphs and Their Applications*. Cambridge University Press. 1998. ISBN 0 521 59345.

[3] Einarsson H. & Arrhenius G. *utomatic design of diagnosis systems using consistency based residuals - optimizing isolation and computational load*. Master's thesis, Uppsala University, 2005. UPTEC F 1401-5757; 05032.

[4] Svärd C. & Wassén H. *Development of Methods for Automatic Design of Residual Generators*. Master's thesis, Dept. of Electrical Engineering, Linköping 2006. Reg nr: LITH-ISY-EX–06/3888–SE.

[5] Eriksson L. *Structural Algorithms for Diagnostic System Design Using Simulink Models*. Master's thesis, Dept. of Electrical Engineering, Linköping 2005. Reg nr: LiTH-ISY-EX-3601-2004.

[6] Krigsman K. & Nilsson J. *Code Generation for Efficient Real-Time Execution of Diagnostic Residual Generators*. Master's thesis, Dept. of Microelectronics and Information Technology, KTH, 2005. IMIT/LECS-2004-66.

[7] Gustafsson F. Ljung L. & Millnert M. *Signalbehandling*. Studentlitteratur. 2001. ISBN 91-44-01709-X.

[8] Nyberg M. & Frisk E. *Model Based Diagnosis of Technical Processes*. Linköping. 2006.

[9] Glad T. & Ljung L. *Reglerteori, Flervariabla och olinjära metoder*. Studentlitteratur. 2003. ISBN 91-44-03003-7.

[10] T. Wei &, M. Li *High order numerical derivatives for one-dimensional scattered noisy data*. Applied Mathematics and Computation. 175:1744-1759. 2006

[11] Gustafsson F. *Adaptive Filtering and Change Detection*. John Wiley & Sons Ltd. 2001. ISBN 0 471 49287 6.

[12] Frisk E. & Åslund J. *Lowering orders of derivatives in non-linear residual generation using realization theory*. Automatica 41:1799-1807. 2005.

[13] Wilson J. Rugh *Linear System Theory, Second Edition*. Prentice-Hall. 1996. ISBN 0-13-441205-2.

[14] Krysander M. *Design and Analysis of Diagnostic Systems Utilizing Structural Methods*. Department of Electrical Engineering, Linköping 2003. ISBN 91-7373-733-X.

[15] Wahlström J. *Control of EGR and VGT for emission control and pumping work minimization in diesel engines*. Lic thesis, Department of Electrical Engineering, Linköping 2006. LiU-TEK-LIC-2006:52, Thesis No. 1271.

[16] De Persis C. & Isidori A., *A Geometric Approach to Nonlinear Fault Detection and Isolation*. IEEE Transactions on Automatic Control, Vol. 46, No. 6. 2001.

[17] Petzold L. *Differential/Algebraic Equations are Not ODE's*. Applied mathematics Division, Sandia National Laboratories, Livermore, California 94550. SIAM J. SCI. STAT. COMPUT. Vol. 3, No. 3, September 1982.

[18] Scania AB (publ) *Årsredovisning 2006 Scania CV AB*.

[19] Krysander M. & Åslund J. & Nyberg M. *An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis*. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, Accepted for publication.

[20] Söderkvist S. *Tidskontenuerliga Signaler & System*. Tryckeriet Erik Larsson AB 2000, 3:e upplagan.

Appendix A

Residual Generators from Example 6.2

From the system

$$e_1 : \dot{x}_1 = -x_1 - 5x_2 + u \tag{A.1a}$$
$$e_2 : \dot{x}_2 = -x_2 + u \tag{A.1b}$$
$$e_3 : y_1 = x_1 + x_2 \tag{A.1c}$$
$$e_4 : y_2 = x_2 \tag{A.1d}$$

there are three different MSO sets found. These are seen in Table A.1.

MSO	Equations
1	$\{e_1, e_3, e_4, d_1\}$
2	$\{e_2, e_4, d_2\}$
3	$\{e_1, e_2, e_3, d_1, d_2\}$

Table A.1. Different MSO sets found from system (A.1).

All possible residual generators will then be

$$R_1 = \dot{y}_1 - \dot{y}_2 + y_1 + 4y_2 - u \tag{A.2a}$$

$$R_2 = \dot{y}_2 + y_2 - u \tag{A.3a}$$

$$x_1 = \frac{x_1^d}{4} + \frac{5}{4}y_1 - u \tag{A.4a}$$
$$x_1^d = \frac{d}{dt}x_1 \tag{A.4b}$$
$$R_3 = y_2 - y_1 - x_1 \tag{A.4c}$$

$$x_2 = \frac{x_2^d}{4} - \frac{1}{4}\frac{d}{dt}y_1 + \frac{1}{4}u \tag{A.5a}$$

$$x_2^d = \frac{d}{dt}x_2 \tag{A.5b}$$

$$R_4 = y_2 - x_2 \tag{A.5c}$$

$$x_2 = -x_2^d + u \tag{A.6a}$$

$$x_2^d = \frac{d}{dt}x_2 \tag{A.6b}$$

$$R_5 = y_2 - x_2 \tag{A.6c}$$

$$x_2 = -x_2^d + u \tag{A.7a}$$

$$x_1 = -x_1^d - 5x_2 + u \tag{A.7b}$$

$$x_1^d = \frac{d}{dt}x_1 \tag{A.7c}$$

$$x_2^d = \frac{d}{dt}x_2 \tag{A.7d}$$

$$R_6 = y_1 - x_1 - x_2 \tag{A.7e}$$

$$x_1 = \frac{x_1^d}{4} + \frac{5}{4}y_1 - u \tag{A.8a}$$

$$x_1^d = \frac{d}{dt}x_1 \tag{A.8b}$$

$$R_7 = \frac{d}{dt}y_1 + y_1 - \frac{d}{dt}x_1 - x_1 - u \tag{A.8c}$$

$$x_2 = \frac{x_2^d}{4} - \frac{1}{4}\frac{d}{dt}y_1 + \frac{1}{4}u \tag{A.9a}$$

$$x_2^d = \frac{d}{dt}x_2 \tag{A.9b}$$

$$R_8 = \frac{d}{dt}x_2 + x_2 - u \tag{A.9c}$$

$$x_2 = -x_2^d + u \tag{A.10a}$$

$$x_2^d = \frac{d}{dt}x_2 \tag{A.10b}$$

$$R_9 = \frac{d}{dt}y_1 - y_1 - \frac{d}{dt}x_2 - 4x_2 + u \tag{A.10c}$$