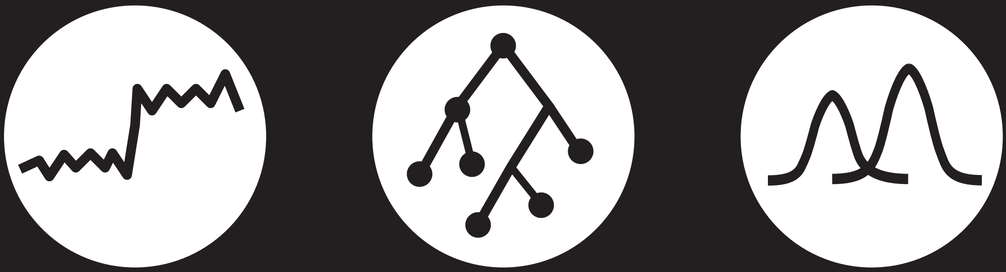
Linköping Studies in Science and Technology. Dissertations, No 1448

Methods for Automated Design of Fault Detection and Isolation Systems

with Automotive Applications

Carl Svärd



Linköping University

Linköping Studies in Science and Technology
Dissertations, No 1448

Methods for Automated Design of Fault Detection and Isolation Systems
with Automotive Applications

Carl Svärd

Linköping University
INSTITUTE OF TECHNOLOGY

Department of Electrical Engineering
Linköping 2012

Linköping Studies in Science and Technology
Dissertations, No 1448

Carl Svärd
carl@isy.liu.se
www.vehicular.isy.liu.se
Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping, Sweden

Svärd, Carl
Methods for Automated Design of Fault Detection and Isolation Systems
with Automotive Applications
ISBN 978-91-7519-894-1
ISSN 0345-7524

Typeset with $\LaTeX 2_\varepsilon$
Printed by LiU-Tryck, Linköping, Sweden 2012

To Emma

Abstract

Fault detection and isolation (FDI) is essential for dependability of complex technical systems. One important application area is automotive systems, where precise and robust FDI is necessary in order to maintain low exhaust emissions, high vehicle up-time, high vehicle safety, and efficient repair. To achieve good performance, and at the same time minimize the need for expensive redundant hardware, model-based FDI is necessary. A model-based FDI-system typically comprises fault detection by means of residual generation and residual evaluation, and finally fault isolation.

The overall objective of this thesis is to develop generic and theoretically sound methods for design of model-based FDI-systems. The developed methods are aimed at supporting an automated design methodology. To this end, the methods require a minimum of human interaction. By means of an automated design methodology the overall design process becomes more efficient and systematic, which also contributes to higher quality. These aspects are of particular importance in an industrial context.

Design of a model-based FDI-system for a complex real-world system is an intricate task that poses several difficulties and challenges that must be handled by the involved design methods. For instance, modeling of these systems often result in large-scale, non-linear, differential-algebraic models. Furthermore, despite substantial modeling work, models are typically not able to capture the behaviors of systems in all operating modes. This results in model-errors of time-varying nature and magnitude. This thesis develops a set of methods able to handle these issues in a systematic manner.

Two methods for model-based residual generation are developed. The two methods handle different stages of the design of residual generators. The first method considers the actual residual generator realization by means of sequential residual generation with mixed causality. The second method considers the problem of how to select an optimal set of residual generators from all possible residual generators that can be created with the first method. Together the two methods enable systematic design of a set of residual generators that fulfills a stated fault isolation requirement. Moreover, the methods are applicable to complex, large-scale, and non-linear differential-algebraic models.

Furthermore, a data-driven method for statistical residual evaluation is developed. The method relies on a comparison of the probability distributions of residuals and exploits no-fault data from the system in order to learn the behavior of no-fault residuals. The method can be used to design residual evaluators capable of handling residuals subject to stochastic uncertainties and disturbances caused by for instance time-varying model errors.

The developed methods, as well as the potential of an automated design methodology, are evaluated through extensive application studies. To verify their generality, the methods are applied to different automotive systems, as well as a wind turbine system. The performances of the obtained FDI-systems are good in relation to the required engineering effort. Particularly, no specific adaption or no tuning of the methods, or the design methodology, were made.

Populärvetenskaplig Sammanfattning

Syftet med denna avhandling är att utveckla metoder för automatiserad design av diagnossystem för att upptäcka och isolera fel i stora komplexa tekniska system. Att upptäcka och isolera fel är viktigt för att garantera ett systems pålitlighet och driftsäkerhet. Ett exempel är tunga lastbilar där förmågan att upptäcka och isolera fel är avgörande för att uppnå och bibehålla exempelvis låga avgasemissioner, hög nyttjandegrad, hög fordonssäkerhet och effektiva reparationer.

Ett sätt att upptäcka fel i ett system är att använda så kallade *modellbaserade residualer*. En modellbaserad residual kan skapas genom att bilda skillnaden mellan en observation från systemet och dess virtuella motsvarighet som skapas genom att simulera systemets felfria beteende med hjälp av en matematisk modell. En residual skild från noll indikerar att det kan finnas något fel i systemet. Genom att använda residualer baserade på observationer från olika delar av systemet så kan ett upptäckt fel dessutom isoleras till en specifik komponent i systemet. Detta är framförallt viktigt för effektiva reparationer.

Design av ett komplett diagnossystem för ett stort komplext system är en utmanande uppgift som kräver en ansenlig mängd utvecklingsarbete. För att erhålla en optimal lösning fodras väldefinierade krav med avseende på exempelvis robusthet och de fel som skall upptäckas och isoleras. Dessutom behövs detaljerad kunskap om systemets beteende, dels för det felfria fallet, men framförallt för alla tänkbara felfall. Denna typ av information är dock sällan tillgänglig åtminstone inte i början av en utvecklingsprocess. Med en automatiserad designmetodik så kan kontinuerliga förbättringar hos diagnossystemet göras snabbt och effektivt då nya krav och mer kunskap tillkommer. Detta innebär en systematisering och effektivisering av utvecklingsprocessen vilket i förlängningen också borgar för högre kvalité.

I avhandlingen utvecklas ett antal generella och teoretiskt välgrundade metoder för att upptäcka och isolera fel i komplexa tekniska system med hjälp av modellbaserade residualer. För att stödja en automatiserad designmetodik är metoderna utvecklade för att kräva minimal användarinteraktion. Stora komplexa system ställer höga krav på metodernas beskaffenheter. Exempelvis så beskrivs dessa system ofta utav stora dynamiska och olinjära modeller vilka måste kunna hanteras. Vidare så leder dessa systems mångfacetterade egenskaper och komplexitet till att modellerna inte alltid är kapabla att beskriva systemens beteende i alla situationer. Metoderna är utvecklade för att hantera dessa svårigheter på ett systematiskt sätt.

De utvecklade metoderna, såväl som potentialen hos en automatiserad designmetodik, utvärderas genom omfattande applikationsstudier. Metoderna appliceras med god framgång för att utveckla kompletta diagnossystem för såväl en dieselmotor i en tung lastbil som en vindkraftturbin. Slutsatsen är att metoderna kan användas för att designa ett diagnossystem med bra prestanda till en mycket liten arbetsinsats.

Acknowledgments

With this thesis I have accomplished one of my goals in life, namely to write a book. It has been five years filled with hard but foremost inspiring and rewarding work. Neither the writing nor the work would have been possible without a number of individual persons.

First of all, I would like to express my sincere gratitude to my supervisor Mattias Nyberg for his guidance, devotion, and ability to inspire. His effort and capability to continuously push things a little bit further have been invaluable. Mattias may be more of a perfectionist than me, and I did not think that was possible.

This work has been performed as a part of a collaborative industrial research project between Scania CV AB in Södertälje and the division of Vehicular Systems, Department of Electrical Engineering, Linköping University.

I would like to thank my assistant supervisors Erik Frisk and Mattias Krysander for giving discussions, and valuable comments and input. Special thanks goes to Erik for his support and for helping me structuring this thesis, and to Mattias for his alert and astute comments. I would also like to thank Lars Nielsen for letting me join his research group Vehicular Systems.

Many thanks also goes to all my colleagues at Scania and Vehicular Systems for contributing to a nice working atmosphere. Special thanks goes to Erik Höckerdal for help with LaTeX issues. Henrik Flemmer is thanked for being a supportive manager. I also thank my managers Niklas Karpe and Peter Vansölin for letting me be a part of this project and do research work. My former managers Mats Jennische and Peter Madsen also deserve acknowledgments. The steering group, with chairman Nils-Gunnar Vågstedt, are also thanked.

The work has been jointly financed by Scania CV AB and Vinnova, Swedish Governmental Agency for Innovation Systems, who are also acknowledged.

Finally, I thank my family and friends for their support. Special and sincere thanks goes to my parents, Åsa and Kjell, and sister Anna, for their understanding and encouragement. Last but not least, I would like to express my utmost gratitude and love to Emma for her great support, patience, and love.

Carl Svärd
Stockholm, April 2012

Contents

Chapter 1

Introduction

1.1 Background and Motivation

The ability to detect and isolate faults in complex technical systems is important in order to fulfill dependability requirements. One important example is automotive systems, where fault detection and isolation (FDI) is necessary in order to obtain and maintain for instance high vehicle uptime, low exhaust emissions, high vehicle safety, efficient repair, and good fuel economy. Uptime, repair, and fuel economy, are important factors in order to minimize the overall life-cycle cost of an automotive vehicle, which is of great importance for vehicle operators. Exhaust emissions are important in order to fulfill strict legislative requirements but are also, together with vehicle safety, important for conscious vehicle operators.

Complex technical systems aimed at commercial use are often designed for low cost and high functionality, and not primarily to facilitate FDI. In particular, this means that there are few sensors and foremost a limited amount of hardware redundancy in the form of multiple sensors measuring the same quantity. To achieve good performance, and at the same time minimize the need for expensive redundant hardware, model-based FDI is often adopted. A model-based FDI-system typically comprises fault detection by means of the two essential steps; residual generation and residual evaluation. In the first step, a model of the system is used together with measurements to generate residuals, i.e., signals that indicate whether there is a fault in the system or not. In the second step, the residuals are evaluated with the aim to reliably detect changes in the residual behavior and make a decision whether the change is caused by faults in the system.

The inherent properties of complex real-world systems in general, and automotive systems in particular, pose several difficulties and challenges when it comes to design of model-based FDI-system. First of all, these systems are typically described by models in the form of large-scale, non-linear, and coupled differential-algebraic equations. Consequently, this kind of models must be handled in the design of a model-based FDI-system, in particular by the method used for design of residual generators. Furthermore, complex systems often contain many physical interconnections which implies that the effect of a fault may propagate in the system and that the effect will be visible in many of the sensor measurements. This, in combination with the small number of sensors, makes fault isolation in these systems a non-trivial problem. For instance, the problem of fault decoupling in residual generators must be handled which in addition is further complicated by the properties of the involved models.

Furthermore, the complexity of the systems in combination with their often many operating modes, imply that models typically not are able to fully describe the behaviors of systems in all operating modes. Regardless of a substantial modeling work, this results in model-errors of time-varying nature and magnitude. In order to be able to detect small faults in a robust way, model errors and additional uncertainties must be handled. Specifically, this issue must be handled by the method used for design of residual evaluators.

1.2 Objective

In an industrial context, and with the challenges and difficulties discussed above in mind, it is clear that design of a complete model-based FDI-system for a complex real-world system is an intricate task that demands a substantial engineering effort. To obtain an optimal design, it is required to have well-defined requirements regarding for example robustness and the faults to detect and isolate. In addition, it is required to have detailed knowledge of the behavior of the supervised system. Both in the no-fault case, but in particular also in all fault cases. This kind of information is however seldom available for real-world systems, at least not during early stages in the design process. To conform to this situation, an iterative design process is adopted in this thesis. In this way, continuous improvements of the FDI-system can be made as more knowledge is obtained and additional requirements arise along the design process.

The overall objective of the thesis is to develop generic, systematic, and theoretically sound methods for design of model-based FDI-systems for complex real-world systems. In addition, in order to facilitate the adopted iterative design process, the methods are aimed at supporting an automated design methodology and require a minimum amount of human interaction. By means of an automated design methodology, the FDI-system can be rapidly redesigned and reconfigured which makes the iterative design process more efficient and systematic, and also contributes to higher quality. All these issues are essential in an industrial context.

1.3 Outline

The thesis is divided into two parts. The first part aims at providing the information necessary for placing the contributions of the second part in a scientific and industrial context. The first part consists of Chapters 2, 3, and 4. Chapter 2 discusses FDI in automotive systems with the aim to provide an application oriented background and motivation to the work carried out in the thesis. Chapter 3 considers design of FDI-systems, both in a general and theoretical context, and in an industrial context. Finally, Chapter 4 summarizes the main contributions of the thesis.

The second part consists of five papers enclosed as Papers A - E. Papers A and B consider residual generation, and Paper C residual evaluation. Papers D and E contain application studies in the form of an automotive diesel engine system and wind turbine system, respectively. These papers demonstrate and evaluate the applicability of the methods developed in Papers A, B, and C, in particular, and the potential of an automated design methodology in general.

Chapter 2

Fault Detection and Isolation in Automotive Systems

This chapter discusses fault detection and isolation (FDI) in the context of automotive systems. The overall aim is to provide an application oriented background and motivation to the work carried out in this thesis. The chapter is structured as follows. Section 2.1 presents some automotive systems where FDI is important, and discusses some of their characterizing properties of significance in this context. Section 2.2 elaborates on the importance of FDI as a mean to fulfill a set of requirements on automotive systems. Different activities involving FDI aimed at guarantee fulfillment of these requirements are also discussed. Finally, Section 2.3 presents a set of requirements for FDI in automotive systems. This is done from an industrial perspective, taking the properties of automotive systems in Section 2.1, as well as the properties of the different activities in Section 2.2, into account.

2.1 Automotive Systems

The intention with this section is to give examples of some automotive systems where FDI is important, and also of typical faults that may occur in these systems. Finally, some characteristic properties of automotive systems of particular significance in the context of FDI are highlighted.

2.1.1 Examples

A modern automotive vehicle is a complex *cyber-physical system* that contains electrical, mechanical, chemical, and thermo-dynamical, sub-systems. Of particular interest for heavy-duty vehicles is the diesel engine, which is frequently used as an application example in this thesis. In order to meet requirements in terms of fuel economy, emissions,

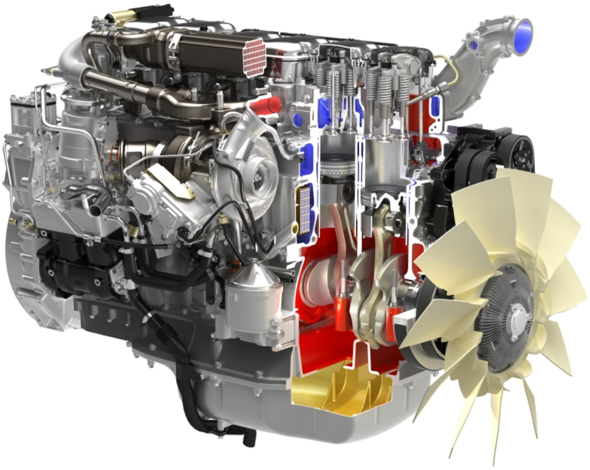
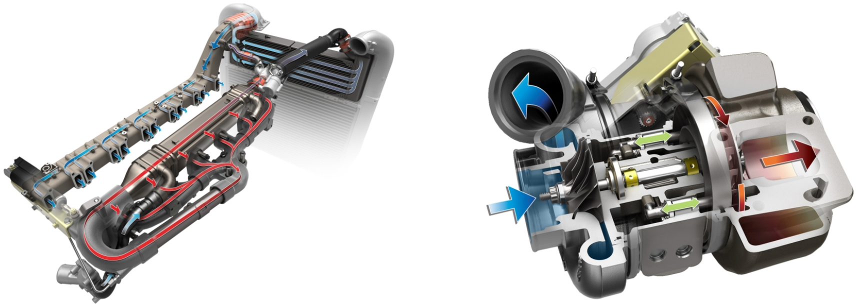


Figure 2.1: A Scania 13-liter, 6-cylinder diesel engine equipped with EGR and VGT. (Courtesy of Scania CV AB. Illustration by Semcon Informatic Graphic Solutions.)

and driveability, a modern diesel engine is equipped with for example Exhaust Gas Recirculation (EGR), Variable Geometry Turbocharger (VGT), and intake manifold throttle, see Figures 2.1, 2.2, and 2.3a. To purify exhausts, diesel engines interact with, and are dependent on, one or several advanced after-treatment systems such as a Diesel Particulate Filter (DPF), and a Selective Catalytic Reduction (SCR) system, see Figure 2.3b. In addition, to further increase driveability and meet safety requirements, they interact with other complex systems in the power train like an automatic gearbox and an auxiliary hydraulic braking system, see Figure 2.4.

2.1.2 Faults

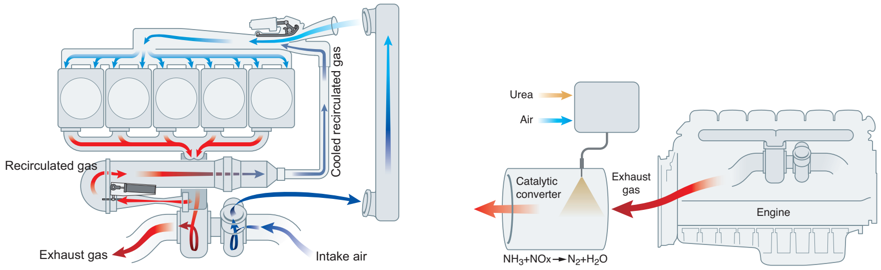
All of the above mentioned systems are, due to their function and complexity, vulnerable to faults. To investigate which faults to detect and isolate, Failure Mode Effect Analysis (FMEA) (Stamatis, 1995) and Fault Tree Analysis (FTA) (Haasl et al., 1981) may be carried out. For the specific case of automotive engines, emission critical faults are of special interest. Much effort is therefore spent on testing the engines in test-beds where faults can be injected and emissions measured. Typical emission critical faults are faults affecting the fuel-injection system, the cooling system, and the gas-flow system, faults in all sensors and actuators, and faults affecting after-treatment systems like the SCR-system and the DPF. Specific examples are gas-leakages in the VGT- or EGR-system, bad UREA quality in the SCR-system, broken or missing filter substrate in the DPF, or a bias- or gain fault in a sensor. Sensors and actuators are in themselves complex cyber-physical systems, and are particularly sensitive to faults, in comparison with for example purely mechanical systems. It is therefore important that especially faults in sensors and actuators in automotive systems can be detected and isolated.



(a) Exhaust Gas Recirculation (EGR).

(b) Variable Geometry Turbocharger (VGT).

Figure 2.2: To meet requirements in terms of fuel economy, emissions, and driveability, a modern diesel engine is equipped with EGR and VGT. (Courtesy of Scania CV AB. Illustration by Semcon Informatic Graphic Solutions.)



(a) Schematic of EGR-system.

(b) Schematic of SCR-system.

Figure 2.3: Usage of EGR and/or SCR in diesel engines reduces the generation of NOx. (Courtesy of Scania CV AB. Illustrations by Semcon Informatic Graphic Solutions.)

2.1.3 Characterizing Properties

Some characterizing properties of automotive systems, and many large real-world systems in general, of particular significance in the context of FDI, are highlighted below.

Few Sensors Automotive systems are typically designed for low cost and high functionality, and not primarily to facilitate FDI. Foremost, this means that there are few sensors in general, and in particular that there is limited, or no, hardware redundancy in the form of multiple sensors measuring the same physical quantity.

Many Operating Modes Automotive system are typically designed to operate in a number of different operating modes and normal operation usually involves several of these. For the example of a diesel engine, operating modes are typically determined by engine torque and engine speed. One operating mode is characterized by low

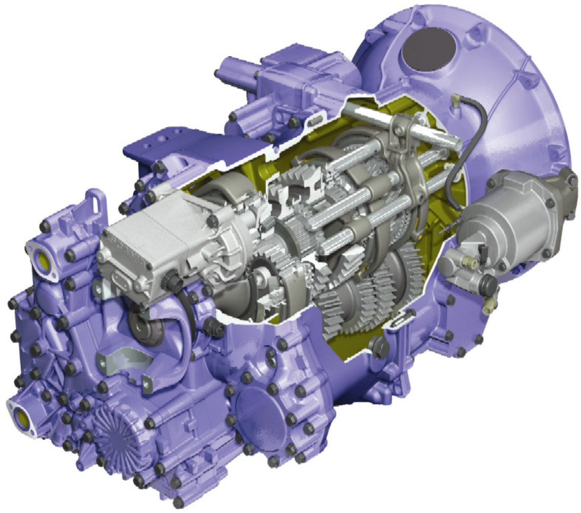


Figure 2.4: Scania GR875R 8-speed gearbox with a retarder. The retarder is a hydraulic braking system used on heavy duty trucks for long continuous braking, for example to maintain constant speed down a slope. (Courtesy of Scania CV AB. Illustration by Semcon Informatic Graphic Solutions.)

engine speed and high engine torque, and another mode by high engine speed, but low engine torque.

Highly Interconnected Automotive systems often contain many physical interconnections. For an example, the exhaust and intake parts of the diesel engine depicted in Figure 2.1 are coupled by means of the shaft connecting the turbine and the compressor. This implies that the effect of a fault may propagate in the system and effects will be visible in many of the measurements.

Complex Models Typically, physical modeling based on first principles of physics is utilized for modeling of automotive systems. As a consequence of the inherent complexity of automotive systems, as well as their multi-domain features, modeling typically results in large-scale, highly non-linear, differential-algebraic equations. In addition, due to the many interconnections in the systems, models are often highly coupled.

2.2 Importance of Fault Detection and Isolation

Automotive vehicles are designed in order to fulfill requirements in terms of:

- high vehicle uptime,
- low exhaust emissions,

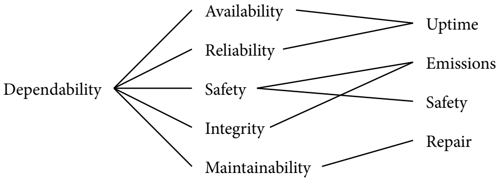


Figure 2.5: High vehicle uptime, low exhaust emissions, high vehicle safety, as well as efficient repair, are important for the dependability of an automotive vehicle.

- high vehicle safety,
- efficient repair,
- good fuel economy,
- high driveability.

High *vehicle uptime* together with efficient *repair*, in the sense that the time at the workshop is minimized, maximizes the possible revenue for a vehicle operator. Good *fuel economy* and efficient repair, in the sense that no unnecessary parts are changed, minimizes the vehicle cost. Vehicle uptime, repair, and fuel economy, are thus all important factors in order to minimize the overall life-cycle cost of an automotive vehicle. This, in combination with high *safety* and high *driveability*, is of great importance for vehicle operators. Requirements on low *exhaust emissions* are mainly driven by legislations.

The properties high vehicle uptime, low exhaust emissions, high safety, as well as efficient repair, are all examples of the more general dependability (Laprie, 1992; Storey, 1996) attributes *availability*, *reliability*, *safety*, *integrity*, and *maintainability*, see Figure 2.5.

A fault in the vehicle or any of its sub-systems may lead to a failure in the form of an impairment of any of the required properties listed above, for instance in the form of a standstill vehicle, increased exhaust emissions, or a non-functional braking system. Such consequences may be prevented, or at least reduced, if the fault can be detected, isolated, and accommodated. Thus, FDI is a mean in order to achieve the properties above.

To ensure achievement of the required properties, FDI is performed by means of the three activities:

- legislative on-board diagnosis,
- off-board diagnosis,
- on-board fault accommodation.

For an illustration, see Figure 2.6. These activities may be performed independently, but typically there are dependencies. For instance, results from legislative on-board diagnosis may be exploited for off-board diagnosis at the workshop. Nevertheless, the ability to be able to detect and isolate faults, to some extent, is important for all three activities. Next, the different activities will be discussed.

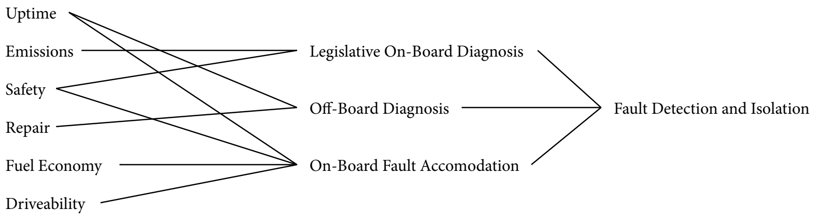

Uptime
Emissions
Safety
Repair
Fuel Economy
Driveability
Legislative On-Board Diagnosis
Off-Board Diagnosis
On-Board Fault Accomodation
Fault Detection and Isolation

Figure 2.6: Legislative on-board diagnosis, off-board diagnosis, and on-board fault accommodation, are important activities in order to achieve properties such as high vehicle uptime, low exhaust emissions, high safety, efficient repair, good fuel economy, and high driveability. All these activities involve fault detection and isolation.

2.2.1 Legislative On-Board Diagnosis

The on-board diagnosis (OBD) legislations (United Nations, 2008; European Parliament, 2009; California EPA, 2010; United States EPA, 2009) state that all manufactured automotive vehicles must be equipped with a high precision OBD-system capable of detecting faults in all components that, if broken, lead to emissions over pre-defined OBD-thresholds during a specific driving cycle. In addition, it is required that emission critical faults can be isolated. In the OBD-legislations, faults are classified according to their emission criticality and different classes requires different actions. A sufficient action for most faults is activation of a malfunction indicator light (MIL), but severe faults require engine torque limitation, or even engine shutdown. OBD is performed in electronic control units (ECUs), as the vehicle operates on the road. For heavy-duty trucks, emissions of especially nitrogen oxides (NOx) and particulate matter (PM) are crucial. Upcoming legislations in the European Union, Euro VI, require substantially lowered emissions, see Table 2.1.

The upcoming functional safety standard ISO 26262 may result in legislative requirements for faults that may lead to an impairment of the vehicle safety. This will require additional FDI and substantially increase the amount of legislative on-board diagnosis.

2.2.2 Off-Board Diagnosis

Off-board diagnosis refers to activities performed off-board the vehicle, typically in the workshop by a mechanic and with additional external computer support. In this setting, FDI can be combined with decision-theoretic troubleshooting, see, e.g., Heckerman et al. (1995); Langseth and Jensen (2002); Warnquist (2011), in order to not only locate but also replace faulty components. The overall aim of off-board fault diagnosis is to guarantee efficient repair of the vehicle, which in turn contributes to high vehicle uptime.

Due to hardware limitations on-board the vehicle and the ability to actively excite systems when the vehicle is at the workshop, off-board detection and isolation of faults potentially give better and more precise results for repair purposes. In addition, it is possible to exploit more knowledge and information from, and regarding, the vehicle in an off-board setting, and to use more powerful fault isolation methods, e.g., Bayesian fault

Table 2.1: EU Emission Standards for HD Diesel Engines, g/kWh (smoke in m^{-1})

Tier	Date	Test	CO	HC	NOx	PM	Smoke
Euro I	1992, < 85 kW	ECE R-49	4.5	1.1	8.0	0.612	
	1992, > 85 kW		4.5	1.1	8.0	0.36	
Euro II	1996-10		4.0	1.1	7.0	0.25	
	1998-10		4.0	1.1	7.0	0.15	
Euro III	*1999-10, EEVs only*	*ESC & ELR*	*1.5*	*0.25*	*2.0*	*0.02*	*0.15*
	2000-10	ESC & ELR	2.1	0.66	5.0	0.1	0.8
						0.13[1]	
Euro IV	2005-10		1.5	0.46	3.5	0.02	0.5
Euro V	2008-10		1.5	0.46	2.0	0.02	0.5
Euro VI	2013-01		1.5	0.13	0.4	0.01	

[1] for engines of less than 0.75 dm^3 swept volume per cylinder and a rated power speed of more than 3000 min^{-1}

isolation (Jensen and Nielsen, 2007; Schwall and Gerdes, 2002; Pernestål and Warnquist, 2012). Examples of additional knowledge and information may be measurements and on-board diagnosis results from all ECUs in the vehicle, and history from previous workshop visits, etc. These issues greatly contribute to better and more precise FDI results. Nevertheless, despite the quite different prerequisites, FDI is of great importance also in the context of off-board diagnosis.

2.2.3 On-Board Fault Accommodation

On-board fault accommodation, or fault management, is performed in ECUs on-board the vehicle during operation on the road. The aim of on-board fault accommodation is to prevent detected and isolated faults from developing into critical failures by taking appropriate actions, and thereby guarantee high vehicle uptime, high safety, high driveability, and also good fuel economy. With upcoming requirements such as the functional safety standard ISO 26262, it is likely that the amount of safety related fault accommodation will increase.

Typically, different faults require different actions. A common action is reconfiguration of the control system by means of fault tolerant control (FTC), see, e.g., Blanke et al. (2006); Yang et al. (2010). For instance, a fault in a sensor used in closed-loop control is accommodated by switching to open-loop control or by instead using a virtual alternative, e.g., a modeled value, to the faulty sensor and maintain closed-loop control. Some critical faults may however require more intricate actions such as system shutdown. In order to conduct the best possible action at any time, it is important to know which fault that has occurred and thus fault isolation is important also in the context of on-board fault accommodation.

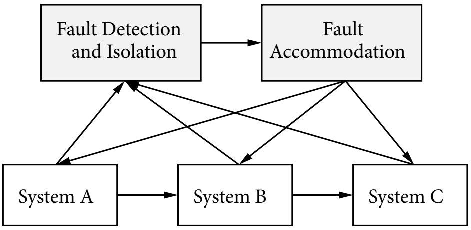


Figure 2.7: Centralized fault accommodation.

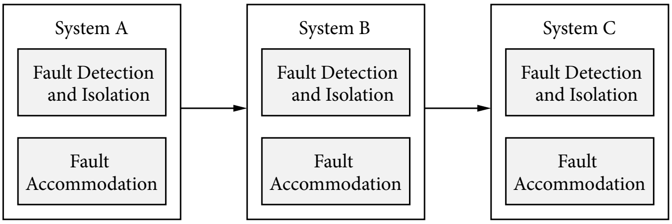

System A
Fault Detection and Isolation
Fault Accommodation
System B
Fault Detection and Isolation
Fault Accommodation
System C
Fault Detection and Isolation
Fault Accommodation

Figure 2.8: Decentralized fault accommodation.

Centralized and Decentralized Fault Accommodation

Traditionally in the literature, *centralized fault accommodation* is adopted, where a centralized FDI unit is used together with a centralized fault accommodation manager, see, e.g., Blanke et al. (2006), and Figure 2.7. However, this creates extra dependencies which increase the complexity and thus this approach is non-modular and scales badly with the size of the system.

Therefore, for large scale automotive systems with functionality distributed over several ECUs, *decentralized fault accommodation* may be more appropriate in order to handle the inherent complexity and making the fault accommodation problem more tractable, see Nyberg and Svärd (2010a,b). Using this approach, the FDI, as well as the fault accommodation, is performed locally in a distributed manner, see Figure 2.8. Independent of which fault accommodation approach that is adopted, FDI is nevertheless needed.

2.3 Requirements on FDI in Automotive Systems

The properties of automotive systems discussed in Section 2.1.3, in combination with the attributes of the different activities discussed in Section 2.2, impose certain requirements on how FDI is performed from and industrial perspective. The most important of these, in the context of this thesis, are listed below.

Existing Hardware Due to cost reasons and space limitations, it is not a desired option to mount additional hardware in the form of for instance multiple sensors, in order to detect and isolate faults. Thus, FDI in automotive systems should be performed by using existing hardware only.

Small Faults As said, the OBD-legislations require detection of all faults that may lead to increased exhaust emissions. Typically, this require detection of small faults in particularly sensor and actuators. For instance, many emission related automotive systems, e.g., the SCR-system, are dependent on correct sensor values for control and, as said in Section 2.1.2, sensors are particularly prone to faults. Even such a small fault as a deviation of a sensor value by 10 % may lead to incorrect control of these systems, which in turn may lead to increased emissions.

On-Board Implementation Apart from the particular case of off-board diagnosis, FDI is to be performed in an on-board environment subject to constraints on computational power and memory, and in some cases also on strict computational deadlines, i.e., real-time. Thus, it is desirable that the FDI can be performed in this environment.

Robustness The many operating modes of automotive systems, as discussed in Section 2.1.3, in combination with the urge to be able to handle different vehicle configurations and vehicle individuals, pose strict requirements on the robustness of the FDI.

Systematic Design In order to obtain an *FDI-system* of high quality, and at the same time enable reconfiguration, redesign, and an efficient overall design process, it is desirable that the methodology used to design the system is systematic.

These requirements will be further considered in the next chapter, in which design of FDI-systems is considered.

Chapter 3

Design of Fault Detection and Isolation Systems

While Chapter 2 aimed at providing an application oriented motivation and background to the work in this thesis, the overall purpose of this chapter is to place the contributions in a scientific and industrial context. To this end, this chapter considers design of fault detection and isolation (FDI) systems, first from a general point of view, and then in the context of automotive systems and Chapter 2. The chapter is structured as follows. In Sections 3.1 and 3.2 some theoretical concepts from the field of model-based diagnosis in general, and FDI in particular, are briefly introduced. For further details, refer to for instance Blanke et al. (2006); Chen and Patton (1999); Hamscher et al. (1992). Section 3.3 discusses some difficulties and challenges that are encountered and must be handled when designing FDI-systems for automotive systems under the prerequisites discussed in Chapter 2. In Section 3.4, design of FDI-systems in an industrial context is discussed and the automated design methodology adopted in this thesis is presented.

3.1 Fault Detection and Isolation Systems

A typical FDI-system consists of a set of *fault detection tests* and a *fault isolation scheme*, see Figure 3.1. The input to the FDI-system is a set of *observations*, i.e., measurements, from the supervised system, and the output is a *diagnosis statement*. The diagnosis statement contains a collection of faults that can be used to explain the observations.

Given a set of observations, $\mathbf{y}$, the outcome of a detection test τ_i is a binary fault *detection result*, d_i, equal to for instance 1 if the test has alarmed, or equal to 0, otherwise. To enable fault isolation, different detection tests typically monitors different faults, and thus different parts of the system. Each fault detection test typically utilizes a subset of the observations in order to determine if any fault is present in its monitored part of the system.

Common traditional approaches for construction of fault detection tests are for example *limit checking*, i.e., to check if a sensor is within its normal operating range, or

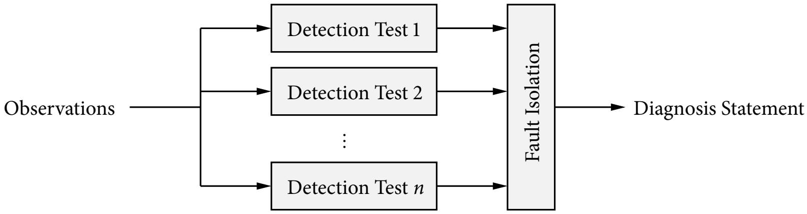


Figure 3.1: A typical FDI-system consists of a set of fault detection tests and a fault isolation scheme.

to employ *hardware redundancy*. For instance, if two sensors are used to measure the same physical quantity, it is possible to test if one of the sensors is faulty by comparing the values of the sensors. Another approach, providing potentially increased diagnosis performance and in which the need of additional, redundant, hardware is avoided, is to use detection tests based on *residuals*. Detection tests based on residuals will be further discussed in Section 3.2.

3.1.1 Fault Isolation

There are several approaches for fault isolation, most originating from the field of Artificial Intelligence (AI), see, e.g., de Kleer and Williams (1987); Reiter (1987); Greiner et al. (1989). Another approach is Bayesian fault isolation, see, e.g.,Jensen and Nielsen (2007). Here, in order to briefly illustrate the concept of fault isolation a method referred to as *structured residuals* (Gertler, 1991), or *structured hypothesis tests* (Nyberg, 2002) will be considered.

For an example, consider a set of detection tests $\{\tau_1, \tau_2, \tau_3\}$ constructed to detect and isolate three faults, $\{f_1, f_2, f_3\}$. The following *fault signature matrix*,

$$\begin{array}{c|ccc} & f_1 & f_2 & f_3 \\ \hline \tau_1 & & 1 & 1 \\ \tau_2 & 1 & & 1 \\ \tau_3 & 1 & 1 & \end{array} \tag{3.1}$$

shows which tests that are *sensitive* to which faults, i.e., test τ_1 is sensitive to faults f_2 and f_3, and so on. Now assume a situation where tests τ_1 and τ_2, but not τ_3, have alarmed. The outcome from the detection tests are thus $d_1 = 1$, $d_2 = 1$, and $d_3 = 0$, which combined with the fault signature matrix (3.1) results in the *sub-diagnosis statements* $D_1 = \{f_2, f_3\}$, $D_2 = \{f_1, f_3\}$, and $D_3 = \{f_1, f_2, f_3\}$. The latter is due to a common convention, saying that nothing can be deduced regarding the status of the system if a test has not alarmed. The diagnosis statement $\mathcal{D}$ then becomes

$$\mathcal{D} = D_1 \cap D_2 \cap D_3 = \{f_2, f_3\} \cap \{f_1, f_3\} \cap \{f_1, f_2, f_3\} = \{f_3\},$$

and it can be concluded that fault f_3 is present. In general, considering an FDI-system containing the detection tests $\{\tau_1, \tau_2, \ldots, \tau_n\}$, where the outcome of the test τ_i is a detection result d_i with a corresponding sub-diagnosis statement D_i. Under a single fault assumption, the diagnosis statement $\mathcal{D}$ can be obtained as

$$\mathcal{D} = \bigcap_{i=1}^{n} D_i,$$

for multiple faults, see, e.g., de Kleer and Williams (1987).

3.2 Detection Tests Based on Residuals

A residual is a signal ideally zero in the no-fault case and non-zero otherwise. A *residual generator*, R_i, takes measurements, $\mathbf{y}$, from the supervised system as input, and produces a residual, r_i, as output, i.e., $r_i = R_i(\mathbf{y})$. A common way to construct a fault detection test based on a residual is to evaluate its behavior in order to conclude whether or not a fault is present in its monitored part of the system. This is done by means of a *residual evaluator*, T_i, taking a residual r_i as input and producing a detection test result d_i as output, i.e., $d_i = T_i(r_i)$. Typically, residual evaluation is performed by forming a *test quantity* from the residual and then threshold the test quantity. In this case, a detection test τ_i based on the residual $r_i = R_i(\mathbf{y})$, by means of a residual evaluator $d_i = T_i(r_i)$, has the form

$$d_i = \tau_i(\mathbf{y}) = T_i(R_i(\mathbf{y})) = \begin{cases} 1 & \text{if} \quad \lambda_i(r_i) > J_i \\ 0 & \text{if} \quad \lambda_i(r_i) \leq J_i, \end{cases} \tag{3.2}$$

where λ_i is a test quantity, and J_i is a detection threshold. Methods for residual generation and residual evaluation will be discussed in Sections 3.2.2 and 3.2.3, respectively.

In Figure 3.2, a residual r and test quantity λ created for fault detection in an automotive diesel engine are shown. A fault occurs at $t = 700$ s. First of all, it is noted that the behavior of the residual r is non-ideal, in the sense that the residual is non-zero both in the no-fault and fault cases. Moreover, it can be seen that the response of the residual to the fault is subtle. Nevertheless, as indicated by the behavior of the test statistic λ, the fault can be detected by an appropriate residual evaluation.

3.2.1 Structure of FDI-Systems based on Residuals

An FDI-system with fault detection tests based on residuals typically have the structure shown in Figure 3.3. Observations $\mathbf{y}$ in the form of measurements from the supervised system are used as input to a residual generation block, which contains a set of residual generators, $R_1, R_2, \ldots, R_n$. The output from the residual generation block is a set of residuals $r_1, r_2, \ldots, r_n$, with $r_i = R_i(\mathbf{y})$. The residuals $r_1, r_2, \ldots, r_n$ are used as input to the residual evaluation block, which contains a set of residual evaluators, $T_1, T_2, \ldots, T_n$. The output from the residual evaluation block is a set of fault detection results, $d_1, d_2, \ldots, d_n$, with $d_i = T_i(r_i)$. These are used as input to the fault isolation block, where the detected fault(s) are isolated.

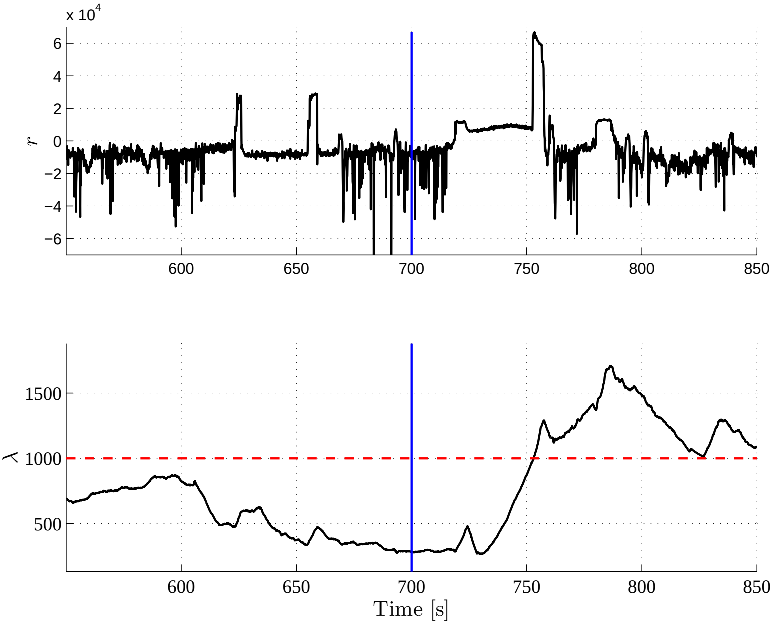


Figure 3.2: A residual r (top) and test quantity λ (bottom) created for fault detection in an automotive diesel engine. The red dashed line is the detection threshold J. A fault occurs at $t = 700$ s. Note the non-ideal behavior of the residual and its subtle response to the fault. By an appropriate residual evaluation by means of the test quantity λ, the fault can nevertheless be detected.

3.2.2 Residual Generation

Typically, residual generators are constructed by using a mathematical model of the system. For instance, a residual can be obtained as the comparison between a value estimated by a model and the corresponding measured quantity. The residual generator consists in this case of the model used for the estimation and the equation describing the comparison, referred to as the *residual equation*.

One approach to residual generation that is of particular interest in this thesis is *sequential residual generation*, see, e.g., Staroswiecki and Declerck (1989); Cassar and Staroswiecki (1997); Staroswiecki (2002); Pulido and Alonso-González (2004); Ploix et al. (2005); Travé-Massuyès et al. (2006); Blanke et al. (2006). This approach has shown to be successful for real applications (Dustegor et al., 2006, 2004; Izadi-Zamanabadi, 2002; Cocquempot et al., 1998), and in addition has the potential to be automated to a high extent.

Additional approaches include for instance *observer-based residual generation*, see, e.g., Massoumnia et al. (1989); Hammouri et al. (2001); De Persis and Isidori (2001); Li and Kadirkamanathan (2001); Martínez-Guerra et al. (2005); Kaboré et al. (2000); Hou (2000); Patton and Hou (1998); Gao and Ding (2007); Vemuri et al. (2001); Shields (1997),

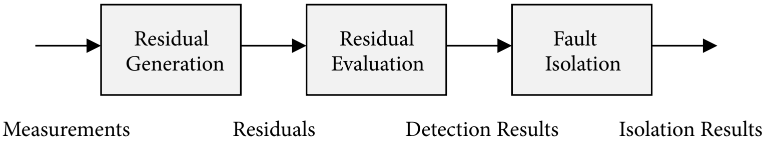


Figure 3.3: An FDI-system with fault detection tests based on residuals by means of residual generation and residual evaluation.

parity-space methods, e.g., Chow and Willsky (1984); Nyberg and Frisk (2006); Varga (2003), and *frequency domain methods*, e.g., Frank and Ding (1994).

Fault Decoupling

To achieve a specific fault signature matrix, for example one similar to (3.1), *decoupling* of faults in residuals is needed. The faults that are decoupled are referred to as *non-monitored faults*, whereas the faults not decoupled are called *monitored faults*. In the example of Section 3.1.1, fault f_1 is decoupled in τ_1, which means that for τ_1, fault f_1 is a non-monitored fault and f_2 and f_3 are monitored faults. Decoupling of faults in a set of tests based on residuals, means that the residuals must be sensitive to different subsets of faults.

In the context of fault isolation, *fault decoupling* is a fundamental problem in residual generation. In most of the observer-based residual generation methods mentioned above, decoupling of faults is obtained by transforming the original model into a sub-model where only the faults of interest are present. In sequential residual generation methods, the original model is often divided into sub-models with specific properties and residual generators are then designed for each sub-model. Since a residual generator only is sensitive to those faults affecting its corresponding sub-model, all other faults are decoupled.

3.2.3 Residual Evaluation

As said, the aim of residual evaluation is to detect changes in the residual behavior caused by faults in the system. Typical components of a residual evaluator are a test quantity λ_i and detection threshold J_i, see (3.2). There are, in essence, two main approaches (Ding et al., 2007) for design of the test quantity and threshold; *statistical residual evaluation* (Willsky and Jones, 1976; Gertler, 1998; Basseville and Nikiforov, 1993; Peng et al., 1997; Al-Salami et al., 2006; Blas and Blanke, 2011; Wei et al., 2011), and *norm-based residual evaluation* (Emami-Naeini et al., 1988; Frank, 1995; Frank and Ding, 1997; Sneider and Frank, 1996; Chen and Patton, 1999; Zhang et al., 2002; Zhong et al., 2007; Ingimundarson et al., 2008; Al-Salami et al., 2010; Li et al., 2011; Abid et al., 2011). In the statistical approach, the framework of statistical hypothesis testing is exploited for design of the test quantity, or test statistic, which typically is based on a likelihood ratio (Gustafsson, 2000). In norm-based approaches, the test quantity is instead based on some norm of the residual, e.g., the mean-power.

Uncertainties

Typically, and as was illustrated in Figure 3.2, residuals are not perfectly zero in the no-fault case due to uncertainties in the form of for example model errors and measurement noise. This may decrease the ability to detect faults and also lead to false detections. The approach used to design the test quantity and threshold in (3.2) are thus important means in order to handle uncertainties and thus guarantee good fault detection. For both statistical and norm-based residual evaluation, *adaptive thresholds* (Clark, 1989; Frank, 1994; Sneider and Frank, 1996) is a traditional approach to handle uncertainties.

The non-ideal behavior of the residual r in Figure 3.2 is a direct consequence of uncertainties in the form of model errors. As illustrated by the fact that the fault nevertheless can be detected by means of the test statistic λ, these uncertainties are handled by proper residual evaluation.

3.3 Design Challenges for Automotive Systems

In Section 2.1.3, it was concluded that automotive systems typically are equipped with few sensors, have many operating modes, contain many physical interconnections, and are described by complex models. Further, it was in Section 2.3 required that FDI in automotive systems should be done in order to, as far as possible, only use existing hardware, be able to detect small faults, be implementable in an on-board environment, and also be robust against uncertainties. In addition, it was concluded that all these desired properties should be achieved by means of a systematic and efficient design methodology.

The prerequisites in terms of the properties of automotive systems, in combination with the requirements on the FDI for these systems, pose several challenges and difficulties that must be handled by the methods used for design of the FDI-system.

Fault Decoupling

As said earlier, fault decoupling is essential in order to obtain fault isolation. The fact that automotive systems typically not are equipped with multiple sensors from start, in combination with the requirement to only use existing hardware for FDI, implies that it is necessary to employ analytical redundancy and model-based FDI in order to obtain good performance. This typically leads to an FDI-system with detection tests based on model-based residuals, as was considered in Section 3.2.

In addition, the many physical interconnections in an automotive system implies that the effect of a fault may propagate in the system and that the effects will be visible in many of the measurements. This fact, in combination with the small number of sensors, makes decoupling of faults a non-trivial problem. Thus, it is of great importance that the methods used to design an automotive FDI-system, in particular the residual generation method, are able to handle this issue. Regarding the requirement concerning systematic design, it is important that the residual generation method facilitates fault decoupling in a systematic manner.

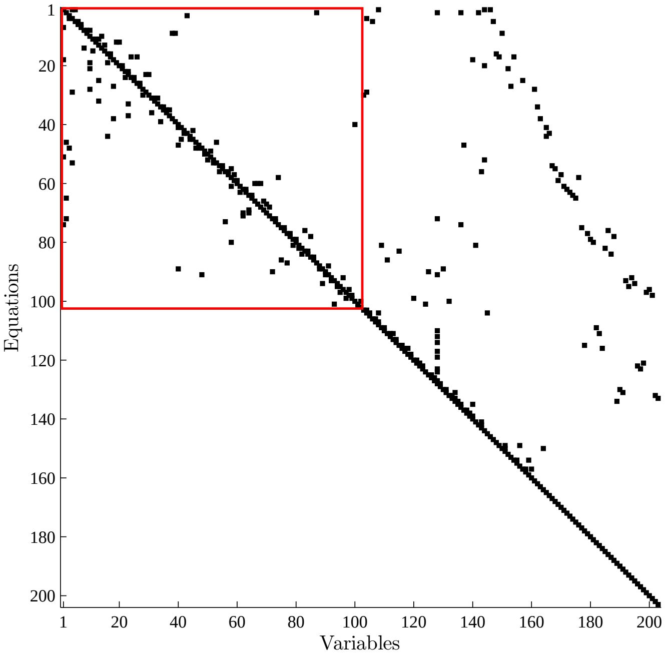


Figure 3.4: The structure of a part of a model of an automotive diesel engine where the rows correspond to model equations and columns to variables in the model. A black square in position (i, j) indicates that equation i contains variable j. The red square illustrates a coupled part of the model corresponding to a differential-algebraic loop. It may be noted the loop involves almost 50% of the equations. A fault affecting any of the equations in the coupled part of the model will influence all other equations in that part.

Model Complexity

As said, automotive systems in general, and automotive diesel engines in particular, yield models in the form of large-scale, non-linear, and coupled differential-algebraic equations. The methods used in the design of the FDI-system, in particular the residual generation method, must thus be able to handle such models in a systematic manner. Moreover, regarding the requirement concerning on-board implementability of automotive FDI-systems, it is important that the output of the residual generation method, i.e., the set of residual generators, is suitable for implementation in an on-board environment despite the complexity of the model used as input.

As said, models of automotive systems are often coupled due to the many interconnections in these systems. In particular, this results in algebraic and differential loops or cycles (Blanke et al., 2006; Katsillis and Chantler, 1997) comprised of sets of equations that contains the same set of unknown variables. This is illustrated in Figure 3.4 which shows the *structure*, i.e., which equations that contain which unknown variables, of a part of a model of an automotive diesel engine. It may be noted that the loop shown in

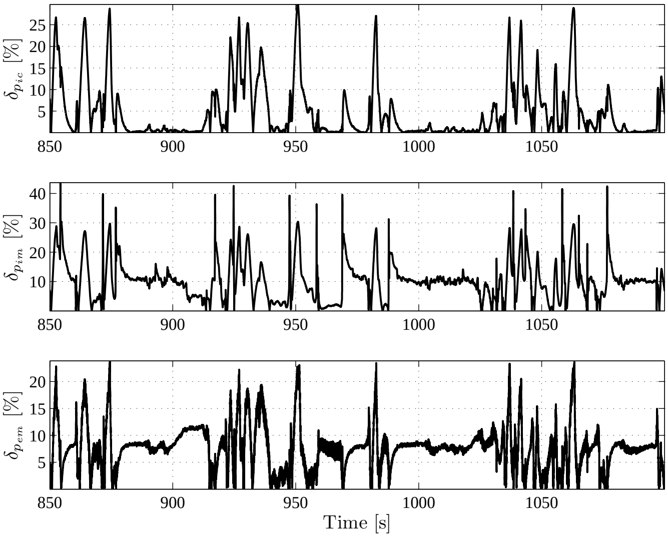


Figure 3.5: Relative model errors for the intercooler manifold pressure p_{im}, intake manifold pressure p_{im}, and exhaust manifold pressure p_{em}, for a model of an automotive diesel engine during a part of the World Harmonized Transient Cycle (WHTC). Note that the magnitude of the model errors vary with time.

Figure 3.4 involves almost 50 % of the equations in the model.

Uncertainties

Due to the inherent complexity of automotive systems, in combination with their many operating modes, models are typically not capable of capturing the behaviors of systems in all different operating modes. This results in uncertainties in the form of model errors, in particular stationary errors (Höckerdal et al., 2011a,b), regardless of substantial modeling work. In addition, due to the typically unfriendly environment in terms of for example high temperatures in or around automotive systems, there are also uncertainties in the form of measurement errors and noise in sensors.

Typically, the magnitudes and nature of these uncertainties are different for different operating modes. For example, the model may be more accurate in one operating mode than another, and a sensor may be more or less sensitive to noise in different operating modes. Since the operating mode of the system varies with time, so does the magnitudes and nature of the uncertainties. This is illustrated in Figure 3.5, which shows relative model errors for three state-variables in a model of an automotive diesel engine during a part of the World Harmonized Transient Cycle (WHTC). Clearly, the magnitude of the model errors vary with time. To meet the posed requirements regarding small faults and robustness, this issue must be handled by the FDI-system. In particular, uncertainties may lead to residuals with the non-ideal behavior illustrated in Figure 3.2 and in order to be able to detect small faults, it is important that uncertainties are handled in the residual evaluation.

3.4 Automated Design of FDI-Systems

Taking the challenges discussed in Section 3.3 into account, it is clear that design of a complete FDI-system for an automotive system, and large-scale real world systems in general, is an intricate and complex task that demands a substantial engineering effort. To obtain an optimal design, it is required to have well-defined requirements regarding for example robustness and the faults to detect and isolate, as well as detailed knowledge of the behavior of the supervised system both in the no-fault case, but in particular also in all fault cases. However, this kind of information is seldom available for real systems, at least not during early stages in the design process.

Conforming to this situation, an iterative design methodology is adopted in this thesis. In this way, continuous improvements of the FDI-system can be made as more knowledge is obtained and additional requirements arise along the design process. To support rapid redesign and reconfiguration, and in this sense make the overall design process more efficient, it is desirable to automate as many steps as possible of the design methodology. In addition, an automated methodology makes the design process more systematic which also contributes to higher quality.

3.4.1 Design Methodology

The considered design methodology is conceptually illustrated in Figure 3.6. The methodology supports design of the residual generation and residual evaluation blocks in an FDI-system with a structure in accordance with Figure 3.3.

The methodology is comprised of three main design stages. Firstly, residual generators are designed given a model of the supervised system and requirements regarding which faults to detect and isolate, robustness, computational power and memory. Design of residual generators is in this work, as in Nyberg (1999); Krysander (2006); Nyberg and Krysander (2008), considered to be a two-step approach, see Figure 3.7. In the first step, given the model, a large number of *candidate residual generators* is found, and in the second step a set of residual generators fulfilling the given requirements is selected and *realized*, i.e., put in a form suitable for implementation.

In the second stage, given the set of residual generators from the first stage and data in the form of measurements from the supervised system, residual evaluators are designed. The third and final stage is to evaluate the complete FDI-system with respect to the given requirements. In particular, it is necessary to investigate the sensitivity of the detection tests, comprised of the residual generators and residual evaluators, to the required set of faults in the presence of uncertainties and disturbances. For this, data in the form of measurements from the supervised system in a set of representative fault-cases, is needed. The results of the evaluation are then analyzed and the process is, if necessary, repeated with revised requirements.

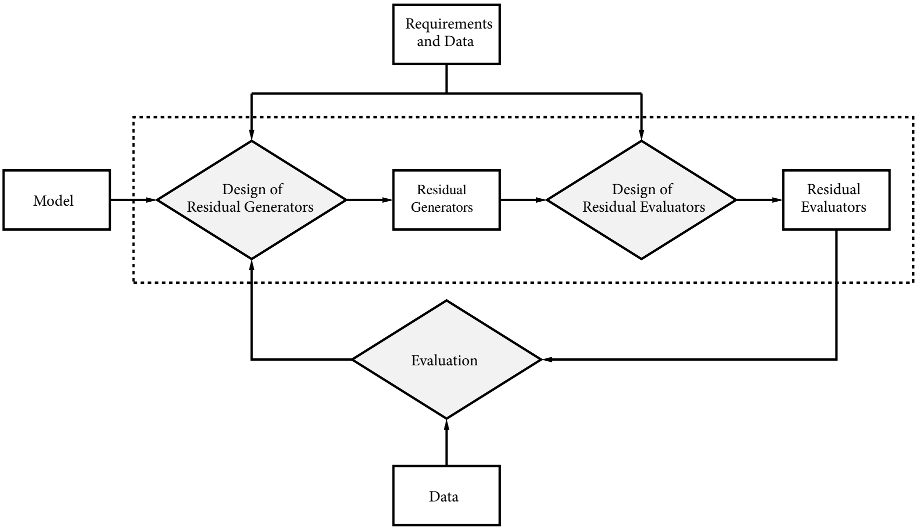


Figure 3.6: The considered methodology for design of FDI-systems.

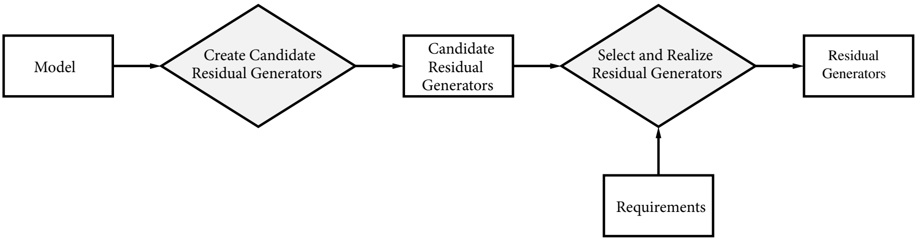

Model
Create Candidate Residual Generators
Candidate Residual Generators
Select and Realize Residual Generators
Residual Generators
Requirements

Figure 3.7: The considered two-step approach for design of residual generators.

It is noted that the available amount of fault data typically is substantially lower than the available amount of no-fault data for a number of reasons. First of all, this is due to the fact that faults are rare. To create fault data, one alternative is to inject faults in the real system. This is however considered to be expensive, both in terms of time and money, since it typically require hardware modifications and active usage of the system. Another alternative is to create fault data by simulation. To give realistic results, this on the other hand requires models capable of describing the faulty system, which in turn require detailed knowledge regarding the behavior of the faulty system and possibly also its environment. This kind of information is seldom available for real applications. Consequently, it may not be possible to exploit fault data in all stages of the design methodology, even though this is highly desirable.

Chapter 4

Summary of Main Contributions

The overall contribution of this thesis is a set of generic and theoretically sound methods for design of FDI-systems, aimed at supporting an automated design methodology. Specifically, this thesis contributes to the part of the design methodology enclosed in the dashed area of Figure 3.6. The developed methods, as well as the overall design methodology, are evaluated through extensive application studies.

In particular, theoretical and methodological contributions are made in the areas of model-based residual generation and statistical residual evaluation in form of three papers enclosed as Paper A, Paper B, and Paper C. Technological contributions, by means of state-of-practice illustrations and proof-of-concept demonstrations, to the field of model-based FDI are made in the form of application studies in two papers enclosed as Paper D and Paper E. In addition, the application studies performed in these two papers together serve as evaluations of the methods developed in Papers A, B, and C.

In the context of the design challenges discussed in Section 3.3, model complexity and fault decoupling are considered in Papers A and B, and uncertainties in Paper C.

4.1 Summaries

Brief summaries of the main contributions of Papers A - E are given below.

Paper A - Residual Generation

The main contribution of Paper A is a sequential residual generation method that enables simultaneous use of integral and derivative causality, i.e., mixed causality. In addition, the method is able to handle equation sets corresponding to algebraic and differential loops in a systematic manner, and is in this sense applicable to complex, large-scale, and coupled models of automotive systems. The method relies on a formal framework for computing unknown variables according to a computation sequence. In this framework, mixed causality is utilized and the analytical properties of the equations in the model, as well as the available tools for algebraic equation solving, are taken into account.

In the context of the two-step approach for design of residual generators, see Figure 3.7, additional contributions are made. Firstly, it is proven that the set of residual generators that can be realized, i.e., created, with the method by necessity is a subset of the set of candidate residual generators based on all Minimal Structurally Over-determined (MSO) sets of equations (Krysander et al., 2008; Gelso et al., 2008; Pulido and Alonso-González, 2004; Travé-Massuyès et al., 2006) in the given model. Secondly, it is empirically shown that the combination of the ability to handle mixed causality and loops substantially increase the amount of realizable candidate residual generators. This is done by means of application of the method to models of two different automotive systems, a diesel engine and a hydraulic braking system.

Paper A relies partly on work presented in Svärd and Nyberg (2008a); Svärd and Nyberg (2008).

Paper B - Selection of Residual Generators

Paper B elaborates further on the two-step approach of Figure 3.7 and in particular the second step. Two different requirements on the sought set of residual generators are considered. Firstly, it is required that the set of residual generators fulfills an isolability requirement, stating which fault that should be isolated from each other. Secondly, motivated by implementation aspects, it is required that the set of residual generators is of minimal cardinality.

Two algorithms for solving the residual generator selection problem are presented in Paper B. Both algorithms exploit a formulation of the selection problem which enables an efficient reduction of the search-space by taking the realizability properties of candidate residual generators, with respect to the considered method for residual generation, into account. The first algorithm provides an exact solution fulfilling both requirements and is suitable for small problems. The second algorithm, which constitutes the main contribution, is suitable for large problems and provides an approximate solution by means of a greedy heuristic by relaxing the minimal cardinality requirement.

Soundness and completeness for both algorithms are shown. In this context, this means that the algorithms provide a set of realizable residual generators fulfilling the stated isolability requirement if, and only if, the requirement can be met with the considered residual generation method. Both algorithms are general in the sense that they are aimed at supporting any computerized residual generation method, not only the method developed in Paper A. The algorithms are applied and evaluated on an automotive diesel engine system.

A preliminary version of Paper B was presented in Svärd et al. (2011a).

Paper C - Residual Evaluation

The main contribution of Paper C is an adaptive and data-driven statistical residual evaluation method. The key property of the method is its ability to handle residuals that are subject to time-varying uncertainties and disturbances, caused for instance by model errors and noise. The test quantity used in the method is based on an explicit comparison of the probability distribution of the residual, estimated online using current data, with a no-fault residual distribution. The no-fault distribution is based on a set of a-priori known no-fault residual distributions, and is continuously adapted to the current situation.

The comparison is done in the framework of statistical hypothesis testing, by means of the Generalized Likelihood Ratio (GLR). To be suitable for on-line implementation in an on-board environment, a computational efficient version of the test quantity is derived by considering a properly chosen approximation to one of the likelihood maximization problems in the GLR. As a second contribution, an algorithm is proposed for learning the required set of no-fault residual distributions off-line from no-fault training data. This algorithm is based on a formulation of the learning problem as a K-means clustering problem. The residual evaluation method is demonstrated and extensively evaluated by application to a residual designed for fault detection in an automotive diesel engine.

A preliminary version of Paper C was presented in Svärd et al. (2011c).

Papers D and E - Application Studies

In Paper D, the methods for residual generation, residual generator selection, and residual evaluation, from Papers A, B, and C, respectively, are combined into an automated design methodology and applied for design of an FDI-system for an automotive diesel engine. In Paper E, the methods for residual generation and residual generator selection are combined with a preliminary version of the residual evaluation method, and applied for design of an FDI-system for the Wind Turbine Benchmark (Fogh Odgaard et al., 2009).

Papers D and E contain minor theoretical contributions. Technological contributions are however made in the sense that both works illustrate how a set of generic methods may be combined into a complete methodology in order to solve a realistic industrial FDI problem. In this sense, these works serve as an illustration of the state-of-practice in model-based fault detection and isolation. Moreover, the papers evaluate and verify the applicability of an automated design methodology in general, and the methods developed in Papers A, B, and C, in particular.

A preliminary version of Paper E was presented in Svärd and Nyberg (2011).

4.2 Publications

The research work leading to this thesis is presented in the following publications.

Journal Papers

- C. Svärd and M. Nyberg. Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(6):1310–1328, 2010 **(Paper A)**

- C. Svärd and M. Nyberg. Automated design of an FDI-system for the wind turbine benchmark. *Journal of Control Science and Engineering*, vol. 2012, 2012. Article ID 989873, 13 pages **(Paper E)**

Submitted

- C. Svärd, M. Nyberg, and E. Frisk. Realizability constrained selection of residual generators for fault diagnosis with an automotive engine application. Submitted to *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2011b **(Paper B)**
- C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Data-driven and adaptive statistical residual evaluation for fault detection with an automotive application. Submitted to *Mechanical Systems and Signal Processing*, 2012b **(Paper C)**
- C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Automotive engine FDI by application of an automated model-based and data-driven design methodology. Submitted to *Control Engineering Practice*, 2012a **(Paper D)**

Conference Papers

- C. Svärd, M. Nyberg, and E. Frisk. A greedy approach for selection of residual generators. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis (DX-11)*, Murnau, Germany, 2011a
- C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Residual evaluation for fault diagnosis by data-driven analysis of non-stationary probability distributions. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011)*, 2011c
- C. Svärd and M. Nyberg. Automated design of an FDI-system for the wind turbine benchmark. In *Proceedings of 18th IFAC World Congress*, Milano, Italy, 2011
- M. Nyberg and C. Svärd. A service based approach to decentralized diagnosis and fault tolerant control. In *Proceedings of 1st Conference on Control and Fault-Tolerant Systems (SysTol'10)*, Nice, France, 2010b
- M. Nyberg and C. Svärd. A decentralized service based architecture for design and modeling of fault tolerant control systems. In *Proceedings of 21st International Workshop on Principles of Diagnosis (DX-10)*, Portland, Oregon, USA, 2010a
- C. Svärd and M. Nyberg. A mixed causality approach to residual generation utilizing equation system solvers and differential-algebraic equation theory. In *Proceedings of 19th International Workshop on Principles of Diagnosis (DX-08)*, Blue Mountains, Australia, 2008a
- C. Svärd and M. Nyberg. Observer-based residual generation for linear differential-algebraic equation systems. In *Proceedings of 17th IFAC World Congress*, Seoul, Korea, 2008b

References

M. Abid, W. Chen, S. X. Ding, and A. Q. Khan. Optimal residual evaluation for nonlinear systems using post-filter and threshold. *International Journal of Control*, 84(3):526 – 39, 2011.

I. M. Al-Salami, S. X. Ding, and P. Zhang. Statistical based residual evaluation for fault detection in networked control systems. In *Proceedings of Workshop on Advances Control and Diagnosis*, Nancy, France, November 2006. Nancy Université Henri Poincaré de Nancy.

I. M. Al-Salami, K. Chabir, D. Sauter, and C. Aubrun. Adaptive thresholding for fault detection in networked control systems. In *Proceedings of the IEEE International Conference on Control Applications*, pages 446 – 451, Yokohama, Japan, 2010.

M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, 1993.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, second edition, 2006.

M. R. Blas and M. Blanke. Stereo vision with texture learning for fault-tolerant automatic baling. *Computers and Electronics in Agriculture*, 75(1):159 – 68, 2011.

California EPA. Sections 1971.1, 1968.2, and 1971.5 of title 13, california code of regulations: HD OBD and OBD II regulations. http://www.arb.ca.gov/msprog/obdprog/hdobdreg.htm, 2010. California Environmental Protection Agency, Air Resources Board.

J. P. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *Proceedings of IFAC Control Ind. Syst.*, pages 841–846, Belfort, France, 1997.

J. Chen and R. J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7):603–613, July 1984.

R. N. Clark. State estimation schemes for instrument fault detection. In R. J. Patton, P. M. Frank, and R. N. Clark, editors, *Fault Diagnosis in Dynamic Systems: Theory and Application*, chapter 2, pages 21–45. Prentice Hall, 1989.

V. Cocquempot, R. Izadi-Zamanabadi, M. Staroswiecki, and M. Blanke. Residual generation for the ship benchmark using structural approach. In *Proceedings of the UKACC International Conference on Control '98*, pages 1480–1485, September 1998.

J. de Kleer and B. C Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1): 97–130, 1987.

C. De Persis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46:853–865, 2001.

S. X. Ding, P. Zhang, and E. L. Ding. Fault detection system design for a class of stochastically uncertain systems. In Hong-Yue Zhang, editor, *Fault Detection, Supervision and Safety of Technical Processes 2006*, pages 705 – 710. Elsevier Science Ltd, 2007.

D. Dustegor, V. Cocquempot, and M. Staroswiecki. Structural analysis for residual generation: Towards implementation. In *Proceedings of the 2004 IEEE Inter. Conf. on Control App.*, pages 1217–1222, 2004.

D. Dustegor, E. Frisk, V. Cocquempot, M. Krysander, and M. Staroswiecki. Structural analysis of fault isolability in the damadics benchmark. *Control Engineering Practice*, 14 (6):597 – 608, 2006.

A. Emami-Naeini, M. M. Akhter, and S. M. Rock. Effect of model uncertainty on failure detection: the threshold selector. *IEEE Transactions on Automatic Control*, 33(12):1106 –1115, 1988.

European Parliament. Regulation No 595/2009 of the european parliament and of the council of 18 june 2009 on type-approval of motor vehicles and engines with respect to emissions from heavy duty vehicles (Euro VI) and on access to vehicle repair and maintenance information and amending Regulation (EC) No 715/2007 and Directive 2007/46/EC and repealing Directives 80/1269/EEC, 2005/55/EC and 2005/78/EC, 2009. European Parliament and the Council of the European Union.

P. Fogh Odgaard, J. Stoustrup, and M. Kinnaert. Fault tolerant control of wind turbines - a benchmark model. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 155–160, Barcelona, Spain, 2009.

P. M. Frank. Enhancement of robustness in observer-based fault-detection. *International Journal of Control*, 59(4):955–981, 1994.

P. M. Frank. Residual evaluation for fault diagnosis based on adaptive fuzzy thresholds. In *Qualitative and Quantitative Modelling Methods for Fault Diagnosis, IEE Colloquium on*, pages 4/1 –411, April 1995. doi:10.1049/ic:19950512.

P. M. Frank and X. Ding. Frequency domain approach to optimally robust residual generation and evalutaion for model-based fault diagnosis. *Automatica*, 30(4):789–804, 1994.

P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6):403 – 424, 1997.

Z. Gao and S. X. Ding. Actuator fault robust estimation and fault-tolerant control for a class of nonlinear descriptor systems. *Automatica*, 43(5):912 – 920, 2007.

E. R. Gelso, S. M. Castillo, and J. Armengol. An algorithm based on structural analysis for model-based fault diagnosis. *Artificial Intelligence Research and Development*, 184: 138–147, 2008.

J. Gertler. Analytical redundancy methods in fault detection and isolation; survey and analysis. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 9–21, Baden-Baden, Germany, 1991.

J. J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, 1998.

R. Greiner, B. A. Smith, and R. W. Wilkerson. A correction to the algorithm in reiter's theory of diagnosis. *Artificial Intelligence*, 41:79–88, 1989.

F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, 2000.

D. E. Haasl, N. H. Roberts, W. E. Vesely, and F. F. Goldberg. *Fault Tree Handbook*. U.S. Nuclear Regulatory Commission, 1981.

H. Hammouri, P. Kabore, and M. Kinnaert. A geometric approach to fault detection and isolation for bilinear systems. *IEEE Transactions on Automatic Control*, 46(9): 1451–1455, September 2001.

W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan Kaufmann Publishers, 1992.

D. Heckerman, J. S. Breese, and K. Rommelse. Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57, 1995.

E. Höckerdal, E. Frisk, and L. Eriksson. EKF-based adaptation of look-up tables with an air mass-flow sensor application. *Control Engineering Practice*, 19(5):442–453, 2011a.

E. Höckerdal, E. Frisk, and L. Eriksson. Bias reduction in DAE estimators by model augmentation: Observability analysis and experimental evaluation. In *50th IEEE Conference on Decision and Control*, Orlando, Florida, USA, 2011b.

M. Hou. *Fault detection and isolation for descriptor systems*, chapter 5. Issues of Fault Diagnosis for Dynamic Systems. Springer-Verlag, 2000.

A. Ingimundarson, A. G. Stefanopoulou, and D. A. McKay. Model-based detection of hydrogen leaks in a fuel cell stack. *IEEE Transactions on Control Systems Technology*, 16 (5):1004 –1012, 2008.

R. Izadi-Zamanabadi. Structural analysis approach to fault fiagnosis with application to fixed-wing aircraft motion. In *Proceedings of the 2002 American Control Conference*, volume 5, pages 3949–3954, 2002.

F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer, 2007.

P. Kaboré, S. Othman, T. F. McKenna, and H. Hammouri. Observer-based fault diagnosis for a class of non-linear systems - application to a free radical copolymerization reaction. *International Journal of Control*, 73(9):787–803, 2000.

G. Katsillis and M. Chantler. Can dependency-based diagnosis cope with simultaneous equations? In *Proceedings of the 8th Inter. Workshop on Princ. of Diagnosis, DX'97*, pages 51–59, Le Mont-Saint-Michel, France, 1997.

M. Krysander. *Design and Analysis of Diagnosis Systems Using Structural Methods*. PhD thesis, Linköpings universitet, June 2006.

M. Krysander, J. Åslund, and M. Nyberg. 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*, 38(1):197–206, 2008.

H. Langseth and F. V. Jensen. Decision theoretic troubleshooting of coherent systems. *Reliability Engineering and System Safety*, 80(1):19–62, 2002.

J. C. Laprie. *Dependability: Basic Concepts and Terminology*. Springer-Verlag, 1992.

P. Li and V. Kadirkamanathan. Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 31(3):337–343, 2001.

W. Li, Z. Zhu, and S. X. Ding. Fault detection design of networked control systems. *IET Control Theory and Applications*, 5(12):1439 – 49, 2011.

R. Martínez-Guerra, R. Garrido, and A. Osorio-Miron. The fault detection problem in nonlinear systems using residual generators. *IMA Journal of Mathematical Control and Information*, 22(2):119–136, 2005.

M. A. Massoumnia, G. C. Verghese, and A.S. Willsky. Failure detection and isolation. *IEEE Transactions on Automatic Control*, 34(3):316–321, March 1989.

M. Nyberg. Automatic design of diagnosis systems with application to an automotive engine. *Control Engineering Practice*, 87(8):993–1005, August 1999.

M. Nyberg. Model-based diagnosis of an automotive engine using several types of fault models. *IEEE Transaction on Control Systems Technology*, 10(5):679–689, 2002.

M. Nyberg and E. Frisk. Residual generation for fault diagnosis of systems described by linear differential-algebraic equations. *IEEE Transactions on Automatic Control*, 51 (12):1995–2000, 2006.

M. Nyberg and M. Krysander. Statistical properties and design criterions for AI-based fault isolation. In *Proceedings of the 17th IFAC World Congress*, pages 7356–7362, Seoul, Korea, 2008.

M. Nyberg and C. Svärd. A decentralized service based architecture for design and modeling of fault tolerant control systems. In *Proceedings of 21st International Workshop on Principles of Diagnosis (DX-10)*, Portland, Oregon, USA, 2010a.

M. Nyberg and C. Svärd. A service based approach to decentralized diagnosis and fault tolerant control. In *Proceedings of 1st Conference on Control and Fault-Tolerant Systems (SysTol'10)*, Nice, France, 2010b.

R. J. Patton and M. Hou. Design of fault detection and isolation observers: A matrix pencil approach. *Automatica*, 34(9):1135–1140, 1998.

Y. Peng, A. Youssouf, P. Arte, and M. Kinnaert. A complete procedure for residual generation and evaluation with application to a heat exchanger. *IEEE Transactions on Control Systems Technology*, 5(6):542 – 555, 1997.

M. Pernestål, A. Nyberg and H. Warnquist. Modeling and troubleshooting with interventions applied to an auxiliary truck braking system. *IFAC Engineering Applications of Artificial Intelligence*, 25:705–719, 2012.

S. Ploix, M. Desinde, and S. Touaf. Automatic design of detection tests in complex dynamic systems. In *Proceedings of 16th IFAC World Congress*, Prague, Czech Republic, 2005.

B. Pulido and C. Alonso-González. Possible conflicts: a compilation technique for consistency-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics, Special Issue on Diagnosis of Complex Systems*, 34(5):2192–2206, 2004.

R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

M. Schwall and C. Gerdes. A probabilistic approach to residual processing for vehicle fault detection. In *In Proceedings of the 2002 ACC*, pages 2552–2557, 2002.

D. N. Shields. Observer design and detection for nonlinear descriptor systems. *International Journal of Control*, 67(2):153–168, 1997.

H. Sneider and P. M. Frank. Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation. *IEEE Transactions on Control Systems Technology*, 4(3):274 –282, 1996.

D. H. Stamatis. *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. ASQ Quality Press, 1995.

M. Staroswiecki. *Fault Diagnosis and Fault Tolerant Control*, chapter Structural Analysis for Fault Detection and Isolation and for Fault Tolerant Control. Encyclopedia of Life Support Systems, Eolss Publishers, Oxford, UK, 2002.

M. Staroswiecki and P. Declerck. Analytical redundancy in non-linear interconnected systems by means of structural analysis. In *Proceedings of IFAC AIPAC'89*, pages 51–55, Nancy, France, 1989.

N. Storey. *Safety-Critical Computer Systems*. Addison Wesley Longman, 1996.

C. Svärd and M. Nyberg. A mixed causality approach to residual generation utilizing equation system solvers and differential-algebraic equation theory. In *Proceedings of 19th International Workshop on Principles of Diagnosis (DX-08)*, Blue Mountains, Australia, 2008a.

C. Svärd and M. Nyberg. Observer-based residual generation for linear differential-algebraic equation systems. In *Proceedings of 17th IFAC World Congress*, Seoul, Korea, 2008b.

C. Svärd and M. Nyberg. A mixed causality approach to residual generation utilizing equation system solvers and differential-algebraic equation theory. Technical Report LiTH-ISY-R-2854, Department of Electrical Engineering, Linköpings Universitet, Sweden, 2008.

C. Svärd and M. Nyberg. Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(6):1310–1328, 2010.

C. Svärd and M. Nyberg. Automated design of an FDI-system for the wind turbine benchmark. In *Proceedings of 18th IFAC World Congress*, Milano, Italy, 2011.

C. Svärd and M. Nyberg. Automated design of an FDI-system for the wind turbine benchmark. *Journal of Control Science and Engineering*, vol. 2012, 2012. Article ID 989873, 13 pages.

C. Svärd, M. Nyberg, and E. Frisk. A greedy approach for selection of residual generators. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis (DX-11)*, Murnau, Germany, 2011a.

C. Svärd, M. Nyberg, and E. Frisk. Realizability constrained selection of residual generators for fault diagnosis with an automotive engine application. Submitted to *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2011b.

C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Residual evaluation for fault diagnosis by data-driven analysis of non-stationary probability distributions. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011)*, 2011c.

C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Automotive engine FDI by application of an automated model-based and data-driven design methodology. Submitted to *Control Engineering Practice*, 2012a.

C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Data-driven and adaptive statistical residual evaluation for fault detection with an automotive application. Submitted to *Mechanical Systems and Signal Processing*, 2012b.

L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component-supported analytical redundancy. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 36(6):1146–1160, November 2006.

United Nations. Regulation no. 49: Uniform provisions concerning the measures to be taken against the emission of gaseous and particulate pollutants from compressionignition engines for use in vehicles, and the emission of gaseous pollutants from positive-ignition engines fuelled with natural gas or liquefied petroleum gas for use in vehicles, 2008. ECE-R49.

United States EPA. 40 CFR Part 86, 89, et al: Control of air pollution from new motor vehicles and new motor vehicle engines; final rule. http://www.epa.gov/obd/regtech/heavy.htm, 2009. United States Environmental Protection Agency.

A. Varga. On computing least order fault detectors using rational nullspace bases. In *Proc. Safeprocess 2003*, pages 229–234, Washington DC, 2003.

A. T. Vemuri, M. M. Polycarpou, and A. R. Ciric. Fault diagnosis of differential-algebraic systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 31(2):143–152, March 2001.

H. Warnquist. Computer-assisted troubleshooting for efficient off-board diagnosis. Technical report, Linköping University, Department of Computer and Information Science, 2011. LiU-TEK-LIC-2011:29, Linköping Studies in Science and Technology. Thesis No. 1490.

X. Wei, H. Liu, and Y. Qin. Fault diagnosis of rail vehicle suspension systems by using glrt. In *Control and Decision Conference (CCDC), 2011 Chinese*, pages 1932 –1936, may 2011.

A. Willsky and H. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control*, 21(1): 108 – 112, feb 1976.

H. Yang, B. Jiang, and V. Cocquempot. *Fault tolerant control design for hybrid systems*. Springer Verlag, 2010.

X. Zhang, M. M. Polycarpou, and T. Parisini. A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE Transactions on Automatic Control*, 47(4):576 –593, 2002.

M. Zhong, H. Ye, S. X. Ding, and G. Wang. Observer-based fast rate fault detection for a class of multirate sampled-data systems. *IEEE Transactions on Automatic Control*, 52 (3):520 – 525, 2007.

Publications

Paper A

Residual Generators for Fault Diagnosis using Computation Sequences with Mixed Causality Applied to Automotive Systems☆

☆Published in *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(6):1310-1328, 2010.

Residual Generators for Fault Diagnosis using Computation Sequences with Mixed Causality Applied to Automotive Systems

Carl Svärd and Mattias Nyberg

*Vehicular Systems, Department of Electrical Engineering,
Linköping University, SE-581 83 Linköping, Sweden.*

Abstract

An essential step in the design of a model-based diagnosis system is to find a set of residual generators fulfilling stated fault detection and isolation requirements. To be able to find a good set, it is desirable that the method used for residual generation gives as many candidate residual generators as possible, given a model. This paper presents a novel residual generation method that enables simultaneous use of integral and derivative causality, i.e., mixed causality, and also handles equation sets corresponding to algebraic and differential loops in a systematic manner. The method relies on a formal framework for computing unknown variables according to a computation sequence. In this framework, mixed causality is utilized and the analytical properties of the equations in the model, as well as the available tools for algebraic equation solving, are taken into account. The proposed method is applied to two models of automotive systems, a Scania diesel engine and a hydraulic braking system. Significantly more residual generators are found with the proposed method in comparison with methods using solely integral or derivative causality.

1 Introduction

Fault diagnosis of technical systems has become increasingly important with the rising demand for reliability and safety, driven by environmental and economical incentives. One example is automotive engines that are by regulations required to have high precision on-board diagnosis of failures that are harmful to the environment (United Nations, 2008).

To obtain good detection and isolation of faults, model-based fault diagnosis is necessary. In the Fault Detection and Isolation (FDI) approach to model-based fault diagnosis, *residuals* are used to detect and isolate faults present in the system, see, e.g., Blanke et al. (2006). Residuals are signals that are ideally zero in the non-faulty case and non-zero else, and are typically generated by utilizing a mathematical model of the system and measurements.

In this paper, we have the view that design of diagnosis systems is a two-step approach, as elaborated in Nyberg and Krysander (2008); Nyberg (1999). In the first step, a large number of candidate residual generators are found, and in the second step the residual generators most suitable to be included in the final diagnosis system are picked out. Since different residual generators have different properties regarding fault and noise sensitivities, it is for the second step important that there is a large selection of different residual generator candidates to choose between. Thus, the initial set of candidate residual generators should be as large as possible.

A residual generator design approach (Staroswiecki and Declerck, 1989) which has shown to be successful in real applications (Dustegor et al., 2004; Izadi-Zamanabadi, 2002; Cocquempot et al., 1998; Svärd and Wassén, 2006; Hansen and Molin, 2006) is to compute unknown variables in the model by solving equation sets one at a time in a sequence, i.e., according to a computation sequence, and then evaluate a redundant equation to obtain a residual. To determine from which equations and in which order the unknown variables should be computed, *structural analysis* is utilized. In addition to (Staroswiecki and Declerck, 1989), similar approaches have been described and exploited in, e.g., Cassar and Staroswiecki (1997); Staroswiecki (2002); Blanke et al. (2006); Pulido and Alonso-González (2004); Ploix et al. (2005); Travé-Massuyès et al. (2006).

In the works mentioned above, the approach is to apply either integral or derivative causality (Blanke et al., 2006) for differential equations. However, as will be illustrated in this paper through application studies, it is advantageous to allow simultaneous use of integral and derivative causality, i.e., mixed causality. Furthermore, real-world applications involve complex models that give rise to algebraic and differential loops or cycles (Blanke et al., 2006; Katsillis and Chantler, 1997), corresponding to sets of equations that have to be treated simultaneously. Thus, it is desirable that a method for residual generation is able to handle mixed causality and equation sets corresponding to algebraic and differential loops. The intention with the following simple example is to illustrate these issues. Consider the set of differential-algebraic equations

$$\begin{aligned} e_1 :&\quad \dot{x}_1 - x_2 = 0 \\ e_2 :&\quad \dot{x}_3 - x_4 = 0 \\ e_3 :&\quad \dot{x}_4 x_1 + 2x_2 x_4 - y_1 = 0 \\ e_4 :&\quad x_3 - y_3 = 0 \\ e_5 :&\quad x_2 - y_2 = 0, \end{aligned} \tag{1}$$

which is a subsystem of a model describing the planar motion of a point-mass satellite (Brockett, 1970; De Persis and Isidori, 2001), and where x_1, x_2, x_3, x_4 are unknown variables and y_1, y_2, y_3 known variables. Assume that we want to use equation e_5 as residual. This implies that the unknown variables x_1, x_2, x_3, x_4 must be computed from the equations e_1, e_2, e_3, e_4. A *structure*, i.e., which unknown variables are contained in which equations, of the equation set $\{e_1, e_2, e_3, e_4\}$ with respect to $\{x_1, x_2, x_3, x_4\}$, in permuted form, is depicted below.

$$\begin{array}{c|cccc} & x_3 & x_4 & x_2 & x_1 \\ \hline e_4 & \mathbf{1} & & & \\ e_2 & 1 & \mathbf{1} & & \\ e_3 & & 1 & \mathbf{1} & \mathbf{1} \\ e_1 & & & \mathbf{1} & \mathbf{1} \end{array} \tag{2}$$

This structure reveals the order and from which equations, marked with bold, the unknown variables should be computed. It is clear that computation of the variables will involve handling of the differential loop arising in the equation set $\{e_1, e_3\}$, since to compute x_2 the value of x_1 is needed and vice versa. Furthermore, computation of the variables according to (2) will require use of mixed causality: derivative causality when solving for x_4 in e_2, and integral causality when solving for x_1 in e_1.

The main contribution of this paper is a novel method for residual generation that enables simultaneous use of integral and derivative causality, and is able to handle equation sets corresponding to algebraic and differential loops in a systematic manner. In this sense, the proposed method also generalizes previous methods for residual generation, e.g., Staroswiecki and Declerck (1989); Dustegor et al. (2004); Izadi-Zamanabadi (2002); Cocquempot et al. (1998); Cassar and Staroswiecki (1997); Staroswiecki (2002); Blanke et al. (2006); Pulido and Alonso-González (2004); Ploix et al. (2005); Travé-Massuyès et al. (2006). To achieve this, a formal framework for sequential computation of variables is presented. In this framework, tools for equation solving and approximate differentiation, as well as analytical and structural properties of the equations in the model, are essential.

In Section 2 some preliminaries, basic theories and references regarding structural analysis and differential-algebraic equation systems are given. Section 3 presents the framework for sequential computation of variables, in which the concepts *Block-Lower Triangular semi-explicit Differential-Algebraic Equation form* (BLT semi-explicit DAE form), *tools*, and *computation sequence* are important. Tools, or more precisely algebraic equation solving tools, are crucial for the ability to handle loops. In Section 4, it is shown how a computation sequence is utilized for residual generation. The resulting residual generator is referred to as a *sequential residual generator*. Motivated by implementation aspects, the concept of a *proper sequential residual generator* is introduced as a sequential residual generator in which no unnecessary variables are computed and in which computations are performed from as small equation sets as possible. A necessary condition for the existence of a proper sequential residual generator is derived, connecting proper sequential residual generators with *Minimal Structurally Over-determined* (MSO) equation sets (Krysander et al., 2008). An algorithm able to find proper sequential residual generators, given a model and a set of tools, is outlined. A key step in the algorithm is to find minimal and irreducible computation sequences, which is considered in Section 5. In Section 6, the proposed method for residual generation is applied to models of an automotive diesel engine and an auxiliary hydraulic braking system. The application studies clearly show the benefits of using a mixed causality approach and handling algebraic and differential loops. Finally, Section 7 concludes the paper. For readability, proofs to all lemmas and theorems are collected in Appendix A.

2 Preliminaries and Background Theory

Consider a *model*, $M(E, X, Y)$, or M for short, consisting of a set of equations $E = \{e_1, e_2, \ldots, e_m\}$ relating a set of unknown variables $X = \{x_1, x_2, \ldots, x_n\}$, and a set of known, i.e., measured, variables $Y = \{y_1, y_2, \ldots, y_r\}$. Introduce a third variable set $D = \{\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n\}$, containing the (time) derivatives of the variables in X. Without loss of generality, it is assumed that the equations in E are in the form

$$e_i : \quad f_i\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}\right) = 0, \quad i = 1, 2, \ldots, m \tag{3}$$

where $\dot{\mathbf{x}} = (\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n)$ is a vector of the variables in D, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ a vector of the variables in X, and $\mathbf{y} = (y_1, y_2, \ldots, y_r)$ a vector of the variables in Y. Also without loss of generality, it is assumed that each equation $e_i \in E$ contains, at most, one differentiated variable $\dot{x}_j \in D$ and that $\dot{x}_j$ is contained only in one equation. This assumption can be made without loss of generality, since an equation containing more than one differentiated variable always can be written as an equation with only one differentiated variable by introducing new algebraic variables and add trivial differential equations. For an example, consider the equation $\dot{x}_1 + \dot{x}_2 + x_1 = 0$ containing two differentiated variables. By introducing the algebraic variable x_3 and substitute $\dot{x}_2$ with x_3, and then add the equation $x_3 = \dot{x}_2$, the equation can be written as $\dot{x}_1 + x_3 + x_1 = 0$. This equation now contains only one differentiated variable.

Define the set of trajectories of the variables in Y that are *consistent* with the model $M(E, X, Y)$ as

$$\mathcal{O}\left(M\right) = \left\{\mathbf{y} : \exists \mathbf{x}; f_i\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}\right) = 0, i = 1, 2, \ldots, m\right\}. \tag{4}$$

The set $\mathcal{O}\left(M\right)$ is the *observation set* of the model M. We formally define a residual generator as follows.

Definition 1 (Residual Generator). *A system with input* $\mathbf{y}$ *and output* r *is a* residual generator *for the model* $M(E, X, Y)$ *and* r *is a residual if* $\mathbf{y} \in \mathcal{O}\left(M\right) \Rightarrow \lim_{t \to \infty} r \to 0$

2.1 Integral and Derivative Causality

In the context of the methods for residual generation mentioned in Section 1, there are two approaches for handling differential equations, referred to as *integral* and *derivative* causality, see, e.g., Blanke et al. (2006). When adopting integral causality, the differentiated variables, or states, of a differential equation can be computed. The use of integral causality hence relies on the assumption that ordinary differential equations can be solved, i.e., integrated, which in general requires that initial conditions of the states are known. Integral causality is used in for example Pulido et al. (2008) and Pulido and Alonso-González (2004).

If instead derivative causality is applied, a differential equation is interpreted as an algebraic equation and only undifferentiated, i.e., algebraic, variables can be computed. Usage of derivative causality thus relies on the assumption that values of the differentiated variables in a differential equation are available. This requires in general that derivatives of known, or previously computed, variables can be computed or estimated. Derivative causality is used in Staroswiecki (2002), and also adopted in, e.g., Dustegor et al. (2004). The difference between integral and derivative causality is discussed in Pulido et al. (2007) and from a simulation point of view in Cellier and Elmqvist (1993). Causality also plays a central role when using a bond-graph modeling framework, see, e.g., Narasimhan and Biswas (2007).

The chosen causality approach naturally influences which variables that can be computed from an equation set. For instance, consider the differential equation e_1 : $\dot{x}_1 - x_2 = 0$ from (1), where both x_1 and x_2 are unknown variables. If integral causality is used, x_1 can be computed from e_1 but if instead derivative causality is used, x_2 can be computed from e_1.

2.2 Structure of Equation Sets

To study which unknown variables are contained in a set of equations, a structural representation of the equation set will be used. Let $E' \subseteq E$ and introduce the notations

$$\begin{aligned}\mathrm{var}_X(E') &= \left\{x_j \in X : \exists e_i \in E', \frac{\partial f_i}{\partial x_j} \not\equiv 0 \vee \frac{\partial f_i}{\partial \dot{x}_j} \not\equiv 0\right\},\\ \mathrm{var}_D(E') &= \left\{\dot{x}_j \in D : \exists e_i \in E', \frac{\partial f_i}{\partial \dot{x}_j} \not\equiv 0\right\}.\end{aligned}$$

Consider the model (1) and let $X = \{x_1, x_2, x_3, x_4\}$ and $D = \{\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4\}$. For instance, it holds that

$$\mathrm{var}_X(\{e_3\}) = \{x_1, x_2, x_4\}. \tag{5}$$

Let $G = (E, X, A)$ be a bipartite graph where E and X are the (disjoint) sets of vertices, and

$$A = \left\{(e_i, x_j) \in E \times X : x_j \in \mathrm{var}_X(\{e_i\})\right\},$$

the set of arcs. We will call the bipartite graph G the *structure* of the equation set E with respect to X. Note that with this representation, there is no structural difference between the variable x_j and the differentiated variable $\dot{x}_j$. An equivalent representation of G is the $m \times n$ biadjacency matrix B defined as

$$B_{ij} = \begin{cases} 1 & \text{if } (e_i, x_j) \in A \\ 0 & \text{otherwise} \end{cases}$$

Return to the model (1). The structure of the equation set $\{e_1, e_2, e_3, e_3\}$ with respect to $\{x_1, x_2, x_3, x_4\}$ is given by the biadjacency matrix (2). The result in (5) corresponds to the third row of (2).

We will also consider the structure of E with respect to D which refers to the bipartite graph $\bar{G} = (E, D, \bar{A})$, where

$$\bar{A} = \left\{ (e_i, \dot{x}_j) \in E \times D : \dot{x}_j \in \text{var}_D(\{e_i\}) \right\}.$$

2.3 Structural Decomposition

A *matching* on the bipartite graph $G = (E, X, A)$ is a subset of A such that no two arcs have common vertices. A matching with maximum cardinality is a *maximum matching*. A matching is a *complete matching* with respect to E (or X), if the matching covers every vertex in E (or X). By directing the arcs contained in a matching on the bipartite graph G in one direction, and the remaining arcs in the opposite direction, a *directed graph* can be obtained from G, see for example Asratian et al. (1998). A directed graph is said to be *strongly connected* if for every pair of vertices x_i and x_j there is a directed path from x_i to x_j. The maximal strongly-connected subgraphs of a directed graph are called its *strongly-connected components* (SCC).

There exists a unique structural decomposition of the bipartite graph $G = (E, X, A)$, referred to as the Dulmage-Mendelsohn (DM) decomposition, see Dulmage and Mendelsohn (1958); Murota (1987). It decomposes G into irreducible bipartite subgraphs $G^+ = (E^+, X^+, A^+)$, $G_i^0 = (E_i^0, X_i^0, A_i^0)$, $i = 1, 2, \ldots, s$, and $G^- = (E^-, X^-, A^-)$, called DM-components, see Figure 1. The component G^+ is the over-determined part of G, $G^0 = \bigcup_{i=1}^{s} G_i^0$ the just-determined part, and G^- the under-determined part. The DM-components $G_i^0 = (E_i^0, X_i^0, A_i^0)$ correspond to the SCCs of the directed graph induced by any complete matching on the bipartite graph G^0, (Murota, 1987). The equation set $E^0 = \bigcup_{i=1}^{s} E_i^0$ is said to be a just-determined equation set with respect to the variables $X^0 = \bigcup_{i=1}^{s} X_i^0$. For an application of the DM-decomposition see for example Krysander and Frisk (2008).

Algebraic and Differential Loops

If the structure of an equation set, with respect to a set of unknown variables, contains SCCs of larger size than one, the equation set contains *loops* or *cycles*, see, e.g., Blanke et al. (2006); Katsillis and Chantler (1997); Pulido et al. (2007). If the equation set contains cyclic dependencies including unknown differentiated variables, the loop is said to be *differential*, else *algebraic*.

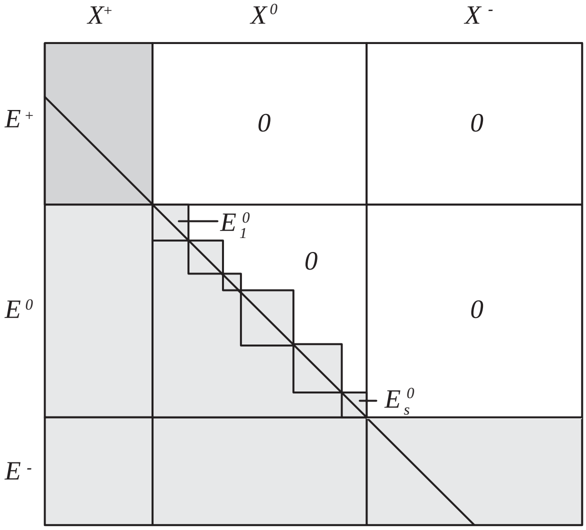


Figure 1: DM-decomposition of the bipartite graph $G = (E, X, A)$. The DM-components $G_i^0 = (E_i^0, X_i^0, A_i^0)$ correspond to the SCCs of the structure of E^0 with respect to X^0.

In the example outlined in Section 1, the structure (2), which in fact is the result of a DM-decomposition, revealed three SCCs which are bold-marked. The SCCs are $(\{e_4\}, \{x_3\})$,$(\{e_2\}, \{x_4\})$, and $(\{e_1, e_3\}, \{x_1, x_2\})$ of size 1, 1, and 2 respectively. The latter corresponds to a differential loop.

2.4 Differential-Algebraic Equation Systems

Due to its general form, it is assumed that the model (3) contains both differential and algebraic equations, i.e., it is a Differential-Algebraic Equation (DAE) system, or descriptor system (Kunkel and Mehrmann, 2006; Brenan et al., 1989; Ascher and Petzold, 1998). The most general form of a DAE is $\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}) = 0$, where $\mathbf{f}$ is some vector-valued function, cf. (3). DAEs appear in large classes of technical systems like mechanical-, electrical-, and chemical systems. Further, DAEs are also the result when using physically based object-oriented modeling tools, e.g., Modelica (Mattson et al., 1998).

Differential Index

A common approach when analyzing and solving general DAE-systems, is to seek a reformulation of the original DAE into a simpler and well-structured description with the same set of solutions (Kunkel and Mehrmann, 2006; Brenan et al., 1989). To classify how difficult such a reformulation is, the concept of index has been introduced. There are different index concepts depending on the kind of reformulation that is sought. In this paper we will use the *differential index*, which is defined as the minimum number of times that all or parts of the DAE must be differentiated with respect to time in order to write the DAE as an *explicit Ordinary Differential Equation* (ODE), $\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{y})$, see for example Brenan et al. (1989).

Semi-Explicit DAEs

An important class of DAEs are *semi-explicit* DAEs

$$\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z}, \mathbf{w}, \mathbf{y}) \tag{6a}$$

$$0 = \mathbf{h}(\mathbf{z}, \mathbf{w}, \mathbf{y}), \tag{6b}$$

where $\mathbf{z}$ and $\mathbf{w}$ are vectors of unknown variables, and $\mathbf{y}$ a vector of known variables. A semi-explicit DAE is of index one if and only if (6b) can be (locally) solved for $\mathbf{w}$ so that $\mathbf{w} = \tilde{\mathbf{h}}(\mathbf{z}, \mathbf{y})$, see, e.g., Brenan et al. (1989). An explicit ODE can easily be obtained from a semi-explicit DAE of index one by substituting $\mathbf{w} = \tilde{\mathbf{h}}(\mathbf{z}, \mathbf{y})$ into (6a).

3 Sequential Computation of Variables

In this section a framework for sequential computation of variables is presented. The framework is built upon the concepts BLT semi-explicit DAE form, tools, and computation sequence. The small model (1) introduced in Section 1, will be used as a running example to illustrate and exemplify the theory.

Large sets of equations often have a sparse structure, i.e., only a few unknown variables in each equation. This makes it possible to partition the set of equations into subsets that can be solved, in a sequence, for only a subset of the unknowns. The main argument for computing variables in this way is efficiency and in some cases this may be the only feasible way to compute the unknowns. This approach has been used in the context of equation solving, see Steward (1962); Kron (1963); Steward (1965), and is also utilized in methods for non-causal simulation (Fritzon, 2004).

3.1 BLT Semi-Explicit DAE Form

One property that the partitioning must fulfill, is that computation of variables from a certain subset of equations must only use variables that are known, that is, measured or have been computed from another subset in a previous step of the sequence.

Furthermore, with the efficiency argument in mind, it is most desirable to partition the set of equations into as small blocks, i.e., subsets, as possible. However, even if the equation set has a sparse structure, there could be algebraic or differential loops, that makes it impossible to consider subsets of solely one equation.

In addition, it is desirable that the equations are partitioned into blocks or subsets from which variables can be computed in a straightforward manner. Since the considered set of equations (3) contains both differential and algebraic equations, subsets will correspond to DAEs. Computation of variables from semi-explicit DAEs of index one, referred to as simulation of the DAE, is a well studied problem and several methods exist, see, e.g., Hairer and Wanner (2002); Ascher and Petzold (1998). Furthermore, as said in Section 2.4, a semi-explicit DAE of index one can trivially be transformed to an explicit ODE. Explicit ODEs are suitable for real-time simulation in embedded systems, for example Engine Control Units (ECUs), because real-time simulation often require use of an explicit integration method, e.g., forward Euler (Ascher and Petzold, 1998), which assumes an explicit ODE. For a detailed discussion regarding real-time simulation, see Cellier and Kofman (2006).

Motivated by these arguments, we consider a partitioning of the equation set so that a block-lower triangular form is achieved, where each block corresponds to a semi-explicit DAE of index one.

Definition 2 (BLT Semi-Explicit DAE Form)**.** *The system*

$$\begin{aligned}\dot{\mathbf{z}}_1 &= \mathbf{g}_1\left(\mathbf{z}_1, \mathbf{w}_1, \mathbf{y}\right)\\ \dot{\mathbf{z}}_2 &= \mathbf{g}_2\left(\mathbf{z}_1, \mathbf{z}_2, \mathbf{w}_1, \mathbf{w}_2, \mathbf{y}\right)\\ &\vdots\\ \dot{\mathbf{z}}_s &= \mathbf{g}_s\left(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_s, \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_s, \mathbf{y}\right)\end{aligned} \tag{7}$$

where $\mathbf{w}_i = \left(\mathbf{w}_i^1, \mathbf{w}_i^2, \dots, \mathbf{w}_i^{p_i}\right)$ *and*

$$\mathbf{w}_i^1 = \mathbf{h}_i^1\left(\Psi_i, \mathbf{y}\right) \tag{8}$$

$$\mathbf{w}_i^2 = \mathbf{h}_i^2\left(\Psi_i, \mathbf{w}_i^1, \mathbf{y}\right) \tag{9}$$

$$\vdots \tag{10}$$

$$\mathbf{w}_i^{p_i} = \mathbf{h}_i^{p_i}\left(\Psi_i, \mathbf{w}_i^2, \dots, \mathbf{w}_i^{p_i-1}, \mathbf{y}\right), \tag{11}$$

where

$$\Psi_i = \left(\dot{\mathbf{w}}_1, \dot{\mathbf{w}}_2, \dots, \dot{\mathbf{w}}_{i-1}, \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_i, \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{i-1}\right),$$

for $i = 1, 2, \dots, s$*, and where* $\mathbf{z}_i$ *and* $\mathbf{w}_i$ *are vectors of unknown variables, all pairwise disjoint, and* $\mathbf{y}$ *a vector of known variables, is in* Block-Lower Triangular semi-explicit Differential-Algebraic Equation form *(BLT semi-explicit DAE form).*

Note that it is not necessary that both $\mathbf{z}_i$ and $\mathbf{w}_i$ are present in (7) for every $i = 1, 2, \dots, s$. In particular, the system

$$\begin{aligned}\mathbf{w}_1 &= \mathbf{h}_1\left(\mathbf{y}\right)\\ \mathbf{w}_2 &= \mathbf{h}_2\left(\mathbf{w}_1, \mathbf{y}\right)\\ &\vdots\\ \mathbf{w}_s &= \mathbf{h}_s\left(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{s-1}, \mathbf{y}\right),\end{aligned}$$

containing no differentiated variables at all, also is in BLT semi-explicit DAE form.

Some Properties of the BLT Semi-Explicit DAE Form

Consider the system

$$
\begin{aligned}
\dot{\mathbf{z}}_1 &= \mathbf{g}_1\left(\mathbf{z}_1, \mathbf{w}_1, \mathbf{y}\right) && \text{(12a)}\\
\mathbf{w}_1^1 &= Bh_1^1\left(\mathbf{z}_1, \mathbf{y}\right) && \text{(12b)}\\
\mathbf{w}_1^2 &= \mathbf{h}_1^2\left(\mathbf{z}_1, \mathbf{w}_1^1, \mathbf{y}\right) && \text{(12c)}\\
\dot{\mathbf{z}}_2 &= \mathbf{g}_2\left(\mathbf{z}_1, \mathbf{z}_2, \mathbf{w}_1, \mathbf{w}_2, \mathbf{y}\right) && \text{(12d)}\\
\mathbf{w}_2^1 &= \mathbf{h}_2^1\left(\dot{\mathbf{w}}_1, \mathbf{z}_1, \mathbf{z}_2, \mathbf{w}_1, \mathbf{y}\right) && \text{(12e)}\\
\mathbf{w}_2^2 &= \mathbf{h}_2^2\left(\dot{\mathbf{w}}_1, \mathbf{z}_1, \mathbf{z}_2, \mathbf{w}_1, \mathbf{w}_2^1, \mathbf{y}\right), && \text{(12f)}
\end{aligned}
$$

where $\mathbf{w}_1 = \left(\mathbf{w}_1^1, \mathbf{w}_1^2\right)$ and $\mathbf{w}_2 = \left(\mathbf{w}_2^1, \mathbf{w}_2^2\right)$, which is in BLT semi-explicit DAE form with $s = 2$ and $p_1 = p_2 = 2$. By studying the system (12), we can deduce some properties of the BLT semi-explicit DAE form;

Mixed Causality The form generalizes the use of integral and derivative causality, since for example integral causality is used in (12a) and derivative causality in (12e).

Blocks are DAEs of Index One or Zero Each block, e.g. (12a)-(12c), corresponds to a semi-explicit DAE of, at most, index one with respect to the unknown variables in each block, i.e., $\mathbf{z}_1$ and $\mathbf{w}_1$ in the first block and $\mathbf{z}_2$ and $\mathbf{w}_2$ in the second block. Note that in accordance with the note above, vectors $\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{w}_1$, and $\mathbf{w}_2$ must not all be present in (12). If, for instance, $\mathbf{w}_1$ is missing and hence also (12b) and (12c), the first block is an explicit ODE, i.e., a DAE of index zero. If both $\mathbf{z}_1$ and $\mathbf{w}_1$ are present, the first block corresponds to a semi-explicit DAE of index one.

Transformation to ODE Due to the previous property, a system in BLT semi-explicit DAE form can trivially be transformed to a variant of an explicit ODE. In (12), we may substitute (12b) into (12c) and then substitute the result along with (12b) into (12a) so that we obtain

$$
\begin{aligned}
\dot{\mathbf{z}}_1 &= \mathbf{g}_1\left(\mathbf{z}_1, \mathbf{w}_1, \mathbf{y}\right)\\
&= \mathbf{g}_1\left(\mathbf{z}_1, \left[\mathbf{h}_1^1\left(\mathbf{z}_1, \mathbf{y}\right), \mathbf{h}_1^2\left(\mathbf{z}_1, \mathbf{h}_1^1\left(\mathbf{z}_1, \mathbf{y}\right)\right)\right], \mathbf{y}\right)\\
&= \tilde{\mathbf{g}}_1\left(\mathbf{z}_1, \mathbf{y}\right),
\end{aligned}
$$

and then repeat the procedure for the second block to obtain

$$
\begin{aligned}
\dot{\mathbf{z}}_1 &= \tilde{\mathbf{g}}_1\left(\mathbf{z}_1, \mathbf{y}\right)\\
\dot{\mathbf{z}}_2 &= \tilde{\mathbf{g}}_2\left(\mathbf{z}_1, \mathbf{z}_2, \dot{\mathbf{y}}, \mathbf{y}\right).
\end{aligned}
$$

As said above, ODEs may be preferable in real-time applications.

Blocks are SCCs Each block in the BLT semi-explicit DAE form is a SCC of the structure of the corresponding equations with respect to the unknown variables in that block. This can be seen by studying the structure[1] of the equations in (12) with respect to the variables $\left\{\mathbf{z}_1, \mathbf{w}_1^1, \mathbf{w}_1^2, \mathbf{z}_2, \mathbf{w}_2^1, \mathbf{w}_2^2\right\}$, which is shown in (13). In this structure, the equation in (12a) has been named e_1, the equation in (12b) has been named e_2, and so forth.

$$\begin{array}{c|cccccc}
 & \mathbf{z}_1 & \mathbf{w}_1^1 & \mathbf{w}_1^2 & \mathbf{z}_2 & \mathbf{w}_2^1 & \mathbf{w}_2^2 \\
\hline
e_1 & \mathbf{1} & \mathbf{1} & \mathbf{1} & & & \\
e_2 & \mathbf{1} & \mathbf{1} & & & & \\
e_3 & \mathbf{1} & \mathbf{1} & \mathbf{1} & & & \\
e_4 & 1 & 1 & 1 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\
e_5 & 1 & 1 & 1 & \mathbf{1} & \mathbf{1} & \\
e_6 & 1 & 1 & 1 & \mathbf{1} & \mathbf{1} & \mathbf{1}
\end{array} \tag{13}$$

Efficiency Recall the discussion regarding efficiency in the beginning of Section 3.1. As a consequence of the previous property, the original set of equations is partitioned in as small blocks as possible, in the sense that there are no dependencies between blocks, i.e., no loops occur.

Sequential Computation of Variables The block-lower triangular structure makes it possible to compute variables sequentially by considering the blocks one at the time, starting from the first block. Since the structure guarantees that a certain block only contains unknown variables from the present and previous blocks.

3.2 Computational Tools

Whether a system in BLT semi-explicit DAE form can be obtained from a given set of equations and whether trajectories of the unknown variables can be computed from the resulting system, depends naturally on the properties of the equations in the model. Equally important is also the set of *tools* that are available for use.

Consider the BLT semi-explicit DAE form (12). To obtain for example the function $\mathbf{h}_1^1$ in (12b) from a subset of equations given in the model, some kind of tool for algebraic equation solving is needed. To compute a trajectory of the variable $\mathbf{z}_1$ from (12a), a differential equation must be solved and hence a tool for this is needed. Furthermore, to obtain the derivative $\dot{\mathbf{w}}_1$, present in (12e), from the trajectory of $\mathbf{w}_1$ computed in (12b) and (12c), a tool for differentiation is needed.

Motivated by this discussion, we consider three types of tools; algebraic equation solving tools, differential equation solving tools, and differentiation tools.

Algebraic Equation Solving Tools

A tool for algebraic equation solving is typically some software package for symbolic or numeric solving of linear or non-linear algebraic equations. Algebraic equation solving

[1] It is here assumed that $f(x)$ implies $\frac{\partial f}{\partial x} \not\equiv 0$.

tools are essential for handling models containing algebraic loops. If, for example, the available algebraic equation solving tool only can solve scalar equations, loops can not be handled.

More precisely, an algebraic equation solving tool (AE tool) is a function taking a set of variables $V_i \subseteq X \cup D$ and a set of equations $E_i \subseteq E$ as arguments, and returning a function $\mathbf{g}_i$, which can be a symbolic expression or numeric algorithm, taking variables from $\{X \cup D\} \setminus V_i$ and Y as arguments and returning a vector corresponding to the elements in V_i. Now assume that $\mathbf{g}_i$ is the function returned by an AE tool when V_i and E_i are used as arguments, and that the equation set $\bar{E}_i$ corresponds to $\mathbf{v}_i = \mathbf{g}_i(\mathbf{u}_i, \mathbf{y})$, where $\mathbf{v}_i$ is a vector of the elements in V_i, $\mathbf{u}_i$ a vector of the elements in $U_i \subseteq \{X \cup D\} \setminus V_i$, and $\mathbf{y}$ a vector of known variables. A natural assumption regarding an AE tool, whatever algorithm or method it corresponds to, is that the AE tool should not introduce new solutions. That is, a solution to $\bar{E}_i$ should also be a solution to the original equation set E_i. Moreover, an AE tool should neither remove solutions, i.e., solutions to E_i must also be solutions to $\bar{E}_i$. Furthermore, motivated by the idea of using sequential computation of variables for residual equation, we are interested in unique solutions. This discussion justifies the following assumption.

Assumption 1. *Given U_i and $\mathbf{y}$, the solution sets of $\bar{E}_i$, obtained from the AE tool, and E_i, with respect to V_i, are equal and unique.*

AE tools giving unique solutions generally assume that the given set of equations contains as many equations as unknown variables. One example is Newton iteration, which is a common numerical method for solving non-linear equations, see, e.g., Ortega and Rheinboldt (2000). In addition, under- and over-determined sets of equations for which an unique analytical solution exists are rare. This motivates the following assumption.

Assumption 2. *An AE tool requires that its arguments V_i and E_i correspond to a just-determined equation set.*

In this work, we assume that tools for algebraic equation solving are available through existing standard software packages like, e.g., Maple or Mathematica, and design and implementation of such tools will not be considered. For solving algebraic loops, also *tearing* (Steward, 1965; Kron, 1963) can be a successful approach. In the following, we also assume that AE tools fulfill the properties stated in Assumptions 1 and 2.

Differential Equation Solving Tools

A differential equation solving tool is typically a method or software for numerical integration of an (explicit or implicit) ODE, i.e., a DAE of index zero. Numerical integration is a well studied area and there are several efficient approaches and methods, see, e.g., Brenan et al. (1989); Ascher and Petzold (1998). Implementations are available in for example Matlab and Simulink.

Independent of which differential equation solving tool that is used, initial conditions for the state variables are in general required. The availability of initial conditions depends on the knowledge about the underlying system represented by the model. For complex physical systems, object-oriented modeling tools, e.g., Modelica (Mattson et al., 1998), are frequently used to build models. Often, this leads to models in which state variables correspond to physical quantities such as pressures and temperatures and then initial conditions may have clear physical interpretations. For example, in an engine model a variable corresponding to the intake manifold pressure should be equal to the ambient pressure when the engine starts.

If all equilibrium points of the considered ODE are (globally) asymptotically stable, or by using, e.g., state-feedback (Khalil, 2002) can be made so, the effect of the initial conditions is neglectable. However, the computed trajectory will in this case differ from the true trajectory for some time due to transients.

Recall from Section 3.1 that each block in a BLT semi-explicit DAE system can be transformed to an explicit ODE. In the following, we assume that differential equation solving tools are always available and that an explicit ODE can be solved, i.e., that trajectories of the state variables in the ODE can be computed, if the initial conditions of the state variables are known and consistent. Of course, this assumption is not always valid and numerical solving of ODEs involves difficulties and problems such as stability and stiffness, but this is not in the scope of this paper.

Differentiation Tools

A differentiation tool is for example an implementation of a method for approximate differentiating of known variables. There are several approaches, e.g., low-pass filtering or smoothing spline approximation (Wei and Li, 2006). An extensive survey of methods can be found in Barford et al. (1999). Methods for approximate differentiation is not in the scope of this paper, and will not be further considered.

In the following, we assume that differentiation of a set of known variables either is possible or not possible. That is, if a tool for approximate differentiation is available, we assume that the quality of the measurements of the involved variables are good enough to support the tool.

One alternative to differentiate variables directly, is to propagate unknown differentiated variables through a set of equations so that these can be expressed as derivatives of measured variables only. Assume for example that we want to compute the derivative $\dot{x}_1$ and we also have that $x_1 = y_1$. To compute $\dot{x}_1$, we use a differentiation tool to compute $\dot{y}_1$ and then use $\dot{x}_1 = \dot{y}_1$.

3.3 Computation Sequence

To describe the way and order in which a set of variables is computed from a set of equations, we will introduce the concept *computation sequence*. Before going into details, we need some additional notation. Let $V \subseteq X \cup D$ and define

$$\mathrm{Diff}\,(V) = \left\{\dot{x}_j \in D : \dot{x}_j \in V \vee x_j \in V\right\}, \tag{14}$$

$$\mathrm{unDiff}\,(V) = \left\{x_j \in X : x_j \in V \vee \dot{x}_j \in V\right\}. \tag{15}$$

For instance, we have that $\mathrm{Diff}\,(\{\dot{x}_1, x_2\}) = \{\dot{x}_1, \dot{x}_2\}$ and $\mathrm{unDiff}\,(\{\dot{x}_1, x_2\}) = \{x_1, x_2\}$.

Now consider the model $M(E, X, Y)$, where E is the set of equations specified in (3), X the set of unknown variables, and Y the set of known variables.

Definition 3 (Computation Sequence). *Given a set of variables $X' \subseteq X$, an AE tool $\mathcal{T}$, and an ordered set*

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \dots, (V_k, E_k)),$$

where $V_i \subseteq \text{var}_X(E_i) \cup \text{var}_D(E_i)$, and $\{E_i\}$ is pairwise disjoint. The ordered set $\mathcal{C}$ is a computation sequence for X' with $\mathcal{T}$, *if*

1. *$X' \subseteq \text{unDiff}(V_1 \cup V_2 \cup \dots \cup V_k)$, and*
2. *a system in BLT semi-explicit DAE form is obtained by sequentially calling the tool $\mathcal{T}$, with arguments V_i and E_i, for each element $(V_i, E_i) \in \mathcal{C}$.*

For an example, recall the model (1), where $E = \{e_1, e_2, e_3, e_4, e_5\}$, $X = \{x_1, x_2, x_3, x_4\}$ and $Y = \{y_1, y_2, y_3\}$. Assume that the given AE tool $\mathcal{T}$ is ideal, in the sense that it can solve all solvable linear and non-linear equations. Then the ordered set

$$\mathcal{C} = ((\{x_3\}, \{e_4\}), (\{x_4\}, \{e_2\}), (\{\dot{x}_1\}, \{e_1\}), (\{x_2\}, \{e_3\})) \tag{16}$$

is a computation sequence for $\{x_1, x_2, x_3, x_4\}$ with $\mathcal{T}$ according to Definition 3, since

$$\text{unDiff}(\{x_3\} \cup \{x_4\} \cup \{\dot{x}_1\} \cup \{x_2\}) = \{x_1, x_2, x_3, x_4\},$$

and the BLT semi-explicit DAE system

$$x_3 = y_3 \tag{17a}$$
$$x_4 = \dot{x}_3 \tag{17b}$$
$$\dot{x}_1 = x_2 \tag{17c}$$
$$x_2 = \frac{-\dot{x}_4 x_1 + y_1}{2x_4}, \tag{17d}$$

is obtained by sequentially calling $\mathcal{T}$ with elements from $\mathcal{C}$ as arguments.

Note that the obtained BLT semi-explicit DAE system (17) has three blocks; the first block corresponds to (17a), the second to (17b), and the third to (17c) and (17d). Also note that the equation set $\{e_1, e_3\}$, containing a differential loop, corresponds to a semi-explicit DAE of index one given by (17c) and (17d). Furthermore, derivative causality is used in (17b) and (17d), and integral causality in (17c).

4 Sequential Residual Generation

In this section it is shown how a computation sequence can be utilized for residual generation. A residual generator based on a computation sequence will be defined as a sequential residual generator. In a sequential residual generator, the generation of a residual will consist of finite sequence of variable computations ending with evaluation of an unused equation. The concepts of minimal and irreducible computation sequence, as well as proper sequential residual generator will then be introduced. A necessary condition for the existence of a proper sequential residual generator is given. The section ends with an algorithm able to find proper sequential residual generators, given a model and an AE tool.

An important property of a computation sequence is given by the following lemma.

Lemma 1. *Let the ordered set*

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \ldots, (V_k, E_k))$$

be a computation sequence for the variables $X' \subseteq X$ with the AE tool $\mathcal{T}$, and let $\bar{E}'$ be the set of equations in BLT semi-explicit DAE form obtained from $\mathcal{C}$ with the AE tool $\mathcal{T}$. Then the solution sets of $\bar{E}'$ and $E_1 \cup E_2 \cup \ldots \cup E_k$, with respect to $V_1 \cup V_2 \cup \ldots \cup V_k$, are equal and unique.

With this lemma, the following important result can be proved.

Theorem 1. *Let $M(E, X, Y)$ be a model, $\mathcal{T}$ an AE tool, and*

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \ldots, (V_k, E_k)),$$

a computation sequence for $X' \subseteq X$ with $\mathcal{T}$, where $E_i \subseteq E$. Also, let $e_i \in E \setminus E_1 \cup E_2 \cup \ldots \cup E_k$ where $\mathrm{var}_X(e_i) \subseteq X'$ and it is assumed that e_i is written as $f_i(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}) = 0$. Then the BLT semi-explicit DAE system obtained from $\mathcal{C}$ with $\mathcal{T}$ and $r = f_i(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y})$, is a residual generator for $M(E, X, Y)$ if

1. *consistent initial conditions of all states are available, and*
2. *all needed derivatives can be computed with the available differentiation tools.*

Motivated by this theorem, we define a sequential residual generator as follows.

Definition 4 (Sequential Residual Generator)**.** *A residual generator for $M(E, X, Y)$ obtained from a computation sequence $\mathcal{C}$ and an equation $e_i \in E$, in accordance with the description in Theorem 1, is a* sequential residual generator *for $M(E, X, Y)$, denoted $\mathcal{S} = (\mathcal{T}(\mathcal{C}), e_i)$, and e_i is a* residual equation.

4.1 Proper Sequential Residual Generator

Regarding implementation aspects, e.g., complexity or numerical issues, smaller computation sequences are generally better. In particular, it is unnecessary to compute variables that are not contained in the residual equation, or not used to compute any of the variables contained in the residual equation. Motivated by this discussion, we make the following definition.

Definition 5 (Minimal Computation Sequence)**.** *Given a set of variables $X' \subseteq X$ and an AE tool $\mathcal{T}$, a computation sequence $\mathcal{C}$ for X' with $\mathcal{T}$ is* minimal*, if there is no other computation sequence $\mathcal{C}'$ for X' with $\mathcal{T}$ such that $\mathcal{C}' \subset \mathcal{C}$.*

Return to the model (1) in Section 1. Consider the last two equations in the model,

$$\begin{aligned} e_4 :&\quad x_3 - y_3 = 0 \\ e_5 :&\quad x_2 - y_2 = 0, \end{aligned}$$

and let $\mathcal{T}$ be an ideal AE tool. The computation sequence

$$\mathcal{C}_1 = ((\{x_3\}, \{e_4\}), (\{x_2\}, \{e_5\})) \tag{18}$$

for $\{x_2, x_3\}$ with $\mathcal{T}$ is minimal. The resulting BLT semi-explicit DAE form is given by

$$x_3 = y_3 \tag{19a}$$

$$x_2 = y_2. \tag{19b}$$

However, $\mathcal{C}_1$ is not minimal for $\{x_3\}$ since $\mathcal{C}_2 = (\{x_3\}, \{e_4\})$ is a (minimal) computation sequence for $\{x_3\}$ with $\mathcal{T}$, and $\mathcal{C}_2 \subset \mathcal{C}_1$.

Computation of variables according to a minimal computation sequence thus implies that no unnecessary variables are computed. However, with the complexity and numerical aspects in mind, it is also most desirable that computation of variables in each step is performed from as small equation sets as possible. This leads to the following definition.

Definition 6 (Irreducible Computation Sequence). *Given a set of variables $X' \subseteq X$ and an AE tool $\mathcal{T}$, a computation sequence*

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \ldots, (V_k, E_k)),$$

for X' with $\mathcal{T}$ is irreducible, *if no element $(V_i, E_i) \in \mathcal{C}$ can be partitioned as $V_i = V_{i1} \cup V_{i2}$ and $E_i = E_{i1} \cup E_{i2}$, such that*

$$\mathcal{C}' = ((V_1, E_1), \ldots, (V_{i1}, E_{i1}), (V_{i2}, E_{i2}), \ldots, (V_k, E_k))$$

is a computation sequence for X' with $\mathcal{T}$.

Return to the equation set $\{e_4, e_5\}$ considered above. Clearly, the ordered set $\mathcal{C}_3 = (\{x_2, x_3\}, \{e_4, e_5\})$ is a minimal computation sequence for $\{x_2, x_3\}$ with the ideal AE tool $\mathcal{T}$. The corresponding BLT semi-explicit DAE system is given by (19). However, $\mathcal{C}_3$ is not irreducible since $\mathcal{C}_1$ given by (18) is also a computation sequence for $\{x_2, x_3\}$.

From now on, we will only consider AE tools fulfilling the following, quite non-limiting, property.

Assumption 3. *Let $E_i = E_{i1} \cup E_{i2}$ and $V_i = V_{i1} \cup V_{i2}$, in accordance with Definition 6. If an AE tool can solve E_i for V_i, it can also solve E_{i1} for V_{i1} and E_{i2} for V_{i2}.*

Sequential residual generators based on minimal and irreducible computation sequences are of particular interest.

Definition 7 (Proper Sequential Residual Generator). *Given an equation $e_i \in E$, an AE tool $\mathcal{T}$, and a computation sequence $\mathcal{C}$ for* $\mathrm{var}_X(e_i)$ *with $\mathcal{T}$. A sequential residual generator $\mathcal{S} = (\mathcal{T}(\mathcal{C}), e_i)$ is* proper, *if $\mathcal{C}$ is a minimal and irreducible computation sequence for* $\mathrm{var}_X(e_i)$ *with $\mathcal{T}$.*

For construction of a sequential residual generator, a computation sequence and a residual equation is needed. Due to Assumption 2, the equation set contained in a computation sequence is a just-determined set of equations. Since the residual equation is redundant, see Theorem 1, it follows that the equations in a computation sequence and the residual equation constitute an over-determined equation set. Hence, an over-determined set of equations is needed to construct a sequential residual generator. For construction of a proper sequential residual generator, a *Minimal Structurally Over-determined* (MSO) set (Krysander et al., 2008), is needed.

Theorem 2. *Let $\mathcal{S} = (\mathcal{T}(\mathcal{C}), e_i)$ be a proper sequential residual generator, where*

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \ldots, (V_k, E_k)),$$

then the equation set $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$ is an MSO set with respect to $\text{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i)$

Note that Theorem 2 establishes a link between structural and analytical methods. This is done without the use of any assumptions of generic equations as in, e.g., Krysander et al. (2008), instead assumptions have been placed on the tools.

Recall again the model (1) and consider the computation sequence $\mathcal{C}$, given by (16), with the corresponding BLT semi-explicit DAE form (17). The computation sequence $\mathcal{C}$ together with the equation e_5 is a sequential residual generator for the model (1), if we assume that the initial condition of x_1 is known and consistent and the derivatives $\dot{x}_3$ and $\dot{x}_4$ can be computed with the available differentiation tools. As a matter of fact, the residual generator is a *proper* sequential residual generator since the computation sequence $\mathcal{C}$ for $\text{var}_X(e_5) = \{x_2\}$ with the ideal AE tool $\mathcal{T}$ is minimal and irreducible. Hence, we can by Theorem 2 conclude that the equation set $E = \{e_1, e_2, e_3, e_4, e_5\}$ is an MSO set.

4.2 Finding Proper Sequential Residual Generators

Theorem 2 states a necessary condition for the existence of a proper sequential residual generator. Hence, a first step when searching for all proper sequential residual generators may be to find all MSO sets. There are efficient algorithms for finding all MSO sets in large equation sets, see, e.g., Krysander et al. (2008).

Motivated by this, we propose the following algorithm for finding proper sequential residual generators, given a model $M(E, X, Y)$ and an AE tool $\mathcal{T}$. The function findAllMSOs is assumed to find all MSO sets in the equation set E. The function findComputationSequence, taking an equation set E', a variable set X' and an AE tool $\mathcal{T}$, is assumed to return a minimal and proper computation sequence for X' with $\mathcal{T}$.

The algorithm is justified by the following theorem.

Theorem 3. *Let $M(E, X, Y)$ be a model and $\mathcal{T}$ an AE tool. Also, let R be the set returned by findResidualGenerators when E, X, and $\mathcal{T}$ are used as input. Then all elements $(\mathcal{T}(\mathcal{C}), e_i) \in R$ are proper sequential residual generators for $M(E, X, Y)$ if, in accordance with Theorem 1, consistent initial conditions of all states are available, and all needed derivatives can be computed with the available differentiation tools.*

```
function findResidualGenerators($E, X, \mathcal{T}$)
    $R := \varnothing$
    MSOs := findAllMSOs($E, X$)
    for all $\bar{E} \in$ MSOs do
        $X' := \mathrm{var}_X(\bar{E})$
        for all $e_i \in \bar{E}$ do
            $E' := \bar{E} \setminus e_i$
            $\mathcal{C}$ := findComputationSequence($E', X', \mathcal{T}$)
            if $\mathcal{C} \neq \varnothing$ then
                $R = R \cup \{(\mathcal{T}(\mathcal{C}), e_i)\}$
            end if
        end for
    end for
    return $R$
end function
```

The most important step in findResidualGenerators is thus to find a minimal and irreducible computation sequence, i.e., the function findComputationSequence. This is the topic of next section.

5 Method for Finding a Computation Sequence

A proper sequential residual generator consists of a BLT semi-explicit DAE system, obtained from a minimal and irreducible computation sequence, and a residual equation. Essential for construction of a proper sequential residual generator is thus to find a minimal and irreducible computation sequence. The method that we propose for finding a computation sequence is presented in this section. First, the different steps of the method are illustrated by studying an example.

5.1 Illustrative Example

Consider the following set of equations,

$$\begin{aligned}
e_1 :&\quad \dot{x}_1 + x_1 x_6 - x_3 - x_5^2 x_7 = 0\\
e_2 :&\quad \dot{x}_2 + x_2 x_3 + y_1 = 0\\
e_3 :&\quad \dot{x}_3 + x_3 - x_2 x_4 + y_2 = 0\\
e_4 :&\quad \dot{x}_4 + x_2 - x_5 - y_3 = 0\\
e_5 :&\quad x_1 - x_2 x_3 - x_4 + x_6 - 2x_7 - y_4 = 0\\
e_6 :&\quad x_3^2 - x_6 - x_7 + y_5 = 0\\
e_7 :&\quad x_4 - y_6 = 0,
\end{aligned}$$

where $X = \{x_1, x_2, \ldots, x_7\}$ are unknown variables and $Y = \{y_1, y_2 \ldots, y_6\}$ known variables. Assume that we want to find a computation sequence for X with a given AE tool.

First identify the SCCs, recall Section 2.3, of the structure of $E = \{e_1, e_2, \ldots, e_7\}$ with respect to X, and order the corresponding partitions of the equation and variable sets accordingly

$$\begin{array}{c|ccccccc}
 & x_4 & x_3 & x_2 & x_5 & x_6 & x_7 & x_1 \\
\hline
e_7 & \mathbf{1} & & & & & & \\
e_3 & 1 & \mathbf{1} & \mathbf{1} & & & & \\
e_2 & & \mathbf{1} & \mathbf{1} & & & & \\
e_4 & 1 & & 1 & \mathbf{1} & & & \\
e_6 & & 1 & & & \mathbf{1} & \mathbf{1} & \\
e_1 & & 1 & & 1 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\
e_5 & 1 & 1 & 1 & & \mathbf{1} & \mathbf{1} & \mathbf{1}
\end{array} \tag{20}$$

The ordered partitions are

$$\mathcal{E} = (\{e_7\}, \{e_2, e_3\}, \{e_4\}, \{e_1, e_5, e_6\})$$

and

$$\mathcal{X} = (\{x_4\}, \{x_2, x_3\}, \{x_5\}, \{x_1, x_6, x_7\}),$$

where each element in $\mathcal{E}$ is a SCC with respect to the corresponding element in $\mathcal{X}$, e.g., $(\{e_2, e_3\}, \{x_2, x_3\})$. The SCCs are marked with bold in (20).

The first SCC, $(\{e_4\}, \{x_7\})$, contains one linear algebraic equation. Under assumption that our AE tool can handle such equations, e_7 is solved for x_4 and we obtain

$$x_4 = y_6. \tag{21}$$

Then consider the next SCC, $(\{e_2, e_3\}, \{x_2, x_3\})$ which contains two differential equations. The permuted structure of $\{e_2, e_3\}$ with respect to the differentiated variables $\{\dot{x}_2, \dot{x}_3\}$ is

$$\begin{array}{c|cc}
 & \dot{x}_3 & \dot{x}_2 \\
\hline
e_3 & 1 & \\
e_2 & & 1
\end{array} \tag{22}$$

As seen, the structure (22) contains two SCCs of size one, $(\{e_3\}, \{\dot{x}_3\})$ and $(\{e_2\}, \{\dot{x}_2\})$. Assuming our AE tool admits it, we then solve e_3 for $\dot{x}_3$ and e_2 for $\dot{x}_2$ and obtain

$$\begin{aligned}
\dot{x}_3 &= -x_3 + x_2 x_4 - y_2 \\
\dot{x}_2 &= -x_2 x_3 - y_1.
\end{aligned} \tag{23}$$

The next SCC, $(\{e_4\}, \{x_5\})$, contains a differential equation. However, since x_5 is the variable intended to compute from the equation, we can handle e_6 as an algebraic equation and solve it for x_5,

$$x_5 = x_2 + \dot{x}_4 - y_3. \tag{24}$$

The SCC $(\{e_1, e_5, e_6\}, \{x_1, x_6, x_7\})$ contains the differential equation e_1 and the two algebraic equations e_5 and e_6. By analyzing the equations we see that x_6 and x_7 are algebraic variables contained in both e_5 and e_6 and that x_1 is a differentiated variable present in e_1. We then solve e_1 for $\dot{x}_1$ and obtain

$$\dot{x}_1 = -x_1 x_6 + x_5^2 x_7 + x_3. \tag{25}$$

The structure of $\{e_5, e_6\}$ with respect to $\{x_6, x_7\}$ reveals a SCC of size two, see (26).

$$\begin{array}{c|cc} & x_6 & x_7 \\ \hline e_5 & 1 & 1 \\ e_6 & 1 & 1 \end{array} \tag{26}$$

Under the assumption that our AE tool can handle it, we solve the equation system $\{e_5, e_6\}$ for $\{x_6, x_7\}$ and obtain

$$\begin{aligned} x_6 &= 2x_3^2 + x_1 - x_2 x_3 - x_4 + 2y_5 - y_4 \\ x_7 &= x_1 - x_2 x_3 - x_4 + x_3^2 + y_5 - y_4. \end{aligned} \tag{27}$$

Collecting the equations (21), (23), (24), (25), and (27) gives

$$\begin{aligned} x_4 &= y_6 && \text{(28a)} \\ \dot{x}_3 &= -x_3 + x_2 x_4 - y_2 && \text{(28b)} \\ \dot{x}_2 &= -x_2 x_3 - y_1 && \text{(28c)} \\ x_5 &= x_2 + \dot{x}_4 - y_3 && \text{(28d)} \\ \dot{x}_1 &= -x_1 x_6 + x_5^2 x_7 + x_3 && \text{(28e)} \\ x_6 &= 2x_3^2 + x_1 - x_2 x_3 - x_4 + 2y_5 - y_4 && \text{(28f)} \\ x_7 &= x_1 - x_2 x_3 - x_4 + x_3^2 + y_5 - y_4, && \text{(28g)} \end{aligned}$$

which is a system in BLT semi-explicit DAE form with four blocks. The equation (28a) correspond to the first block, which only contains an algebraic equation. The second block is given by (28b) and (28c), and correspond to an explicit ODE with respect to the variables $\{x_2, x_3\}$. Hence, integral causality is used in this block. The third block contains (28d), which is a differential equation in which derivative causality is used. The equations (28e)–(28g) constitute the fourth and last block. This block corresponds to a semi-explicit DAE of index one, with respect to the variables $\{x_1, x_6, x_7\}$.

The resulting computation sequence for $\{x_1, x_2, \ldots, x_7\}$ with the given AE tool is,

$$\begin{aligned} \mathcal{C} = (&(\{x_4\}, \{e_7\}), (\{\dot{x}_3\}, \{e_3\}), (\{\dot{x}_2\}, \{e_2\}), (\{x_5\}, \{e_4\}), \\ &(\{\dot{x}_1\}, \{e_1\}), (\{x_6, x_7\}, \{e_6, e_5\})). \end{aligned}$$

5.2 Summary of the Method

Given an AE tool and a just-determined set of equations, the proposed method for finding a computation sequence can be outlined as follows:

1. Find the SCCs of the structure of the equation set with respect to the unknown variables. No distinction is made between a variable and its derivative.

2. For each SCC, split the equations into one set of differential equations and one set of algebraic equations, and the variables into one set of differentiated variables and one set of algebraic variables.

3. For the differential equations, find the SCCs of the structure of the differential equations with respect to the differentiated variables. For each SCC, try to solve the differential equations for the intended differentiated variables with the AE tool. Note that due to the assumption that each differential equation only contains one differentiated variable, all SCCs are of size one.

4. For the algebraic equations, find the SCCs of the structure of the algebraic equations with respect to the algebraic variables. For each SCC, try to solve the algebraic equations for the intended algebraic variables with the AE tool.

5.3 Algorithm

The method is formally described in the function findComputationSequence below. The function takes a just-determined equation set $E' \subseteq E$, a set of unknown variables $X' \subseteq X$, and an AE tool $\mathcal{T}$ as input, and returns an ordered set $\mathcal{C}$ as output. The function findAllSCCs is assumed to return an ordered set of equation and variable pairs, where each pair corresponds to a SCC of the structure of the equation set with respect to the variable set. The order of the SCCs returned by findAllSCCs is assumed to be the one depicted in Figure 1, for more information regarding ordering of SCCs please refer to Murota (1987). There are efficient algorithms for finding SCCs in directed graphs, see for example Tarjan (1972). The DM-decomposition (Dulmage and Mendelsohn, 1958) can also be utilized. In Matlab, the DM-decomposition is implemented in the function `dmperm`, from which also the order of the SCCs, according to Figure 1, easily can be obtained. Other functions used in findComputationSequence are:

- Diff and unDiff, takes a variable set as input and returns its differentiated and undifferentiated correspondence, see (14) and (15).

- isInitCondKnown determines if the initial conditions of the given variables are known and consistent, and the function isDifferentiable determines if the given variables can be differentiated with the available differentiation tool.

- isJustDetermined is used to determine if the structure of the given equation set, with respect to the given variable set, is just-determined. This is essential, since otherwise the computation of SCCs makes no sense.

- getDifferentialEquations takes a set of equations and a set of differentiated variables as input, and returns the differential equations in which the given differentiated variables are contained.

- isToolSolvable determines if the given AE tool can solve the given equations for the given set of variables.
- Append, takes an ordered set and an element as input and simply appends the element to the end of the set.
- The operator $|\cdot|$, taking a set as input, is assumed to return the number of elements in the set and the notion $A(i)$ is used to refer to the i:th element of the ordered set A.

That the ordered set $\mathcal{C}$ returned by findComputationSequence, indeed, is a minimal and irreducible computation sequence is verified in the following theorem.

Theorem 4. *Let $E' \subseteq E$ be a just-determined set of equations with respect to the variables $X' \subseteq X$, and $\mathcal{T}$ an AE tool. If E', X', and $\mathcal{T}$ are used as arguments to findComputationSequence and a non-empty $\mathcal{C}$ is returned, then $\mathcal{C}$ is a minimal and irreducible computation sequence for X' with $\mathcal{T}$.*

6 Application Studies

The objective of this section is to empirically show the benefits of the method for finding sequential residual generators proposed in Sections 4.2 and 5.3. This is done by applying the method to models of an automotive diesel engine and an auxiliary hydraulic braking system. In addition, we illustrate how a sequential residual generator for the diesel engine, found with the proposed method, can be realized. The realized residual generator is then evaluated using real measurements from a truck.

6.1 Implementation and Configuration of the Method

The analytical models of the two systems were obtained from Simulink models by using the toolbox described in Frisk et al. (2006). The resulting models are complex DAEs containing non-linearities like min- and max-functions, look-up tables, saturations, and polynomials.

The functions findResidualGenerators and findComputationSequence, described in Sections 4.2 and 5.3, were implemented in Matlab. In the implementation of findComputationSequence, the symbolic equation solver in Maple was used as AE tool. To find all MSO sets, the algorithm described in Krysander et al. (2008) was used. The MSO sets were arranged in classes, so that MSOs containing the same set of known variables belongs to the same MSO class.

For comparison, different configurations of findComputationSequence were applied to the models. The following parameters, which naturally influences the possibility to find computation sequences, were used for configuration:

SCC: The ability to handle SCCs of larger size than one, i.e., equation sets containing algebraic or differential loops.

IC: The ability to use integral causality.

```
function findComputationSequence($E', X', \mathcal{T}$)
    $\mathcal{C} := \varnothing$
    $S :=$ findAllSCCs($E', X'$)
    for $i = 1, 2, \ldots, |S|$ do
        $(E_i, X_i) := S\,(i)$
        $D_i :=$ Diff($X_i$)
        $Z_i := \mathrm{var}_D(E_i) \cap D_i$
        $W_i := X_i \setminus$ unDiff($Z_i$)
        if not isInitCondKnown($Z_i$) then
            return $\varnothing$
        end if
        $E_{Z_i} :=$ getDifferentialEquations($E_i, Z_i$)
        $E_{W_i} := E_i \setminus E_{Z_i}$
        $S_{Z_i} :=$ findAllSCCs($E_{Z_i}, Z_i$)
        for $j = 1, 2, \ldots, |S_{Z_i}|$ do
            $\left(E^j_{Z_i}, Z^j_i\right) := S_{Z_i}\,(j)$
            if isToolSolvable($Z^j_i, E^j_{Z_i}, \mathcal{T}$) then
                Append($\mathcal{C}, \left(Z^j_i, E^j_{Z_i}\right)$)
            else
                return $\varnothing$
            end if
        end for
        if isJustDetermined($E_{W_i}, W_i$) then
            $S_{W_i} :=$ findAllSCCs($E_{W_i}, W_i$)
            for $j = 1, 2, \ldots, |S_{W_i}|$ do
                $\left(E^j_{W_i}, W^j_i\right) := S_{W_i}\,(j)$
                if isToolSolvable($W^j_i, E^j_{W_i}, \mathcal{T}$) then
                    Append($\mathcal{C}, \left(W^j_i, E^j_{W_i}\right)$)
                else
                    return $\varnothing$
                end if
            end for
        else
            return $\varnothing$
        end if
    end for
    return $\mathcal{C}$
end function
```

Table 1: The Six Configurations of the Method used in the Studies

	D	I	DI	SD	SI	SDI
SCC				x	x	x
IC		x	x		x	x
DC	x		x	x		x

DC: The ability to use derivative causality.

Note that if a configuration uses integral causality, it is assumed that all initial conditions are available. Moreover, it is assumed that all needed derivatives can be computed when a configuration uses derivative causality.

The six possible different configurations are shown in Table 1. For example, configuration SI is able to handle equation sets containing loops and use integral causality, but can not use derivative causality. The configuration corresponding to the novel approach for finding sequential residual generators proposed in this paper is SDI.

6.2 Performance Measures

A sequential residual generator is sensitive to those faults that influence its residual equation and the equations contained in its computation sequence. Different MSO sets correspond to different subsets of the equations in the model. Sequential residual generators obtained from computation sequences and residual equations originating from different MSO sets will thus naturally be sensitive to different subsets of faults. To achieve good fault isolation, it is hence important that residual generators can be constructed from as many MSO sets as possible.

In the automotive applications studied here, it is especially important to detect and isolate faults present in sensors and actuators, that is, faults affecting measurements of known variables. Hence, it also important that residual generators can be constructed from as many MSO classes as possible.

Additionally, different residual generators constructed from the same MSO set or MSO class may have different properties regarding for example numerical aspects, sensitivity to faults, and sensitivity for disturbances such as measurement noise or modeling errors. Hence it is most desirable to be able to evaluate as many residual generators as possible, with real measurement data, to decide which set of residual generators to use in the final diagnosis system.

Motivated by this discussion, we will use the following performance measures to compare the different configurations of the method:

MSO Sets: In how many of the total number of MSO sets at least one residual generator could be found.

MSO Classes: In how many of the total number of MSO classes at least one residual generator could be found.

Residual Generators: The total number of residual generators found.

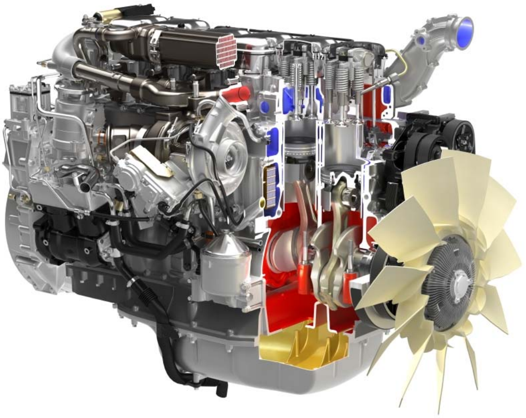


Figure 2: Cutaway of a Scania 13-liter, 6-cylinder diesel engine equipped with EGR and VGT. Illustration by Semcon Informatic Graphics Solutions.

6.3 Automotive Diesel Engine

The studied engine is a 13-liter, 6-cylinder Scania diesel engine equipped with Exhaust Gas Recirculation (EGR) and a Variable Geometry Turbocharger (VGT). A cutaway of the engine can be found in Figure 2.

The model describes the gas-flow in the engine, see Wahlström (2006) for more details. The analytical model extracted from the Simulink model is a non-linear DAE system and contains 282 equations, 272 unknown variables, and 11 known variables. Of the equations, 8 are differential and the rest are algebraic. The differentiated variables represent physical quantities such as pressures, temperatures, and rotational speeds.

In total, 598 MSO sets could be found in the engine model. The MSO sets could be arranged into 210 MSO classes. Theoretically, the total number of potential residual generators that can be constructed from an MSO set is equal to the total number of equations in the MSO set. In this case, 135772 different residual generators could be theoretically constructed from the 598 MSO sets.

The total number of residual generators found and how many of the MSO sets and MSO classes that could be used, for each configuration of the method, is shown in Table 2 and Figure 3. The columns to the left and in the middle of Table 2 shows in how many of the MSO sets and MSO classes at least one residual generator could be found. The column to the right shows the total number of residual generators that could be found for each configuration of the method.

It is obvious that a very small fraction of the potential residual generators were found, about 1.2 %, and that only a small fraction of the MSO sets and MSO classes could be used, independent of configuration. The main reason for this is the complexity of the engine model. The model contains large algebraic and differential loops, including complex non-linear equations, which are impossible to solve analytically. Nevertheless, many more residual generators were found and more MSO sets could be used with configuration

Table 2: Results for Diesel Engine

	MSO Sets	MSO Classes	Residual Generators
D	4	4	46
I	1	1	5
DI	4	4	46
SD	4	4	46
SI	23	20	58
SDI	120	72	1636
Potential	598	210	135772

Table 3: Results for Hydraulic Braking System

	MSO Sets	MSO Classes	Residual Generators
D	21	14	145
I	6	6	18
DI	21	14	147
SD	33	22	288
SI	29	29	71
SDI	65	44	1293
Potential	125	83	4607

SDI, i.e., with mixed causality and the ability to handle loops, in comparison with any other configuration of findComputationSequence.

6.4 Hydraulic Braking System

The Scania auxiliary hydraulic braking system, called *retarder*, is used on heavy duty trucks for long continuous braking, for example to maintain constant speed down a slope. By using the retarder, braking discs can be saved for short time braking.

The model of the hydraulic braking system contains 49 equations, 44 unknown variables, and 9 known variables. It is a non-linear DAE system and contains 4 differential equations and 45 algebraic equations.

The model contains 125 MSO sets, which can be arranged into 83 MSO classes. The total number of possible residual generators for the model of the hydraulic braking system is, theoretically, 4607.

Table 3 and Figure 4 shows, for each configuration of the method, how many of the MSO sets and MSO classes that could be used and the total number of residual generators found for the model of the hydraulic braking system. As seen, a significantly larger fraction of the MSO sets and MSO classes could be used and more residual generators could be found with configuration SDI, in comparison with any other configuration.

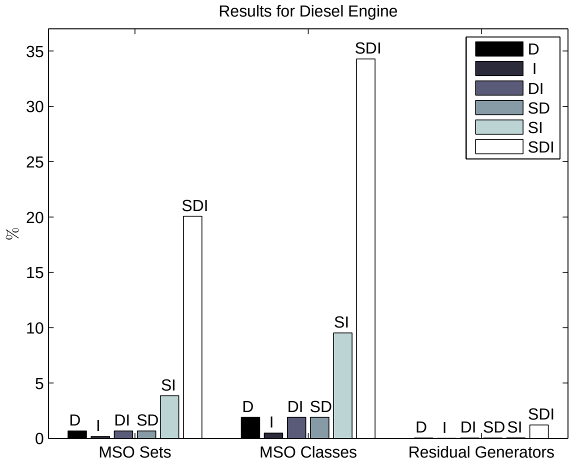


Figure 3: The bars to the left and in the middle shows the fractions of the total number of MSO sets and MSO classes in which a residual generator could be found with each configuration of the method. The bars to the right shows the fractions of the number of potential residual generators that could be found with each configuration of the method.

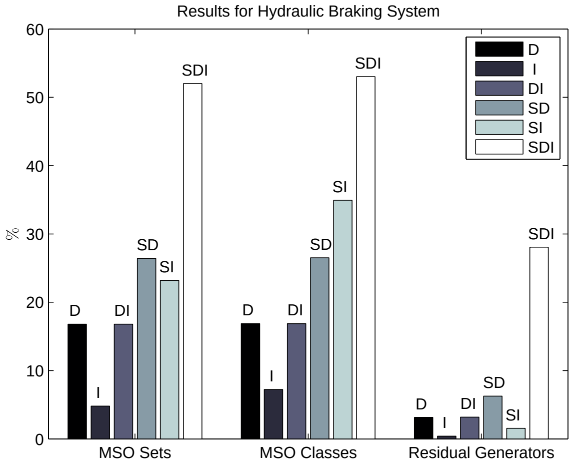


Figure 4: The bars to the left and in the middle shows the fractions of the total number of MSO sets and MSO classes in which a residual generator could be found with each configuration of the method. The bars to the right shows the fractions of the number of potential residual generators that could be found with each configuration of the method.

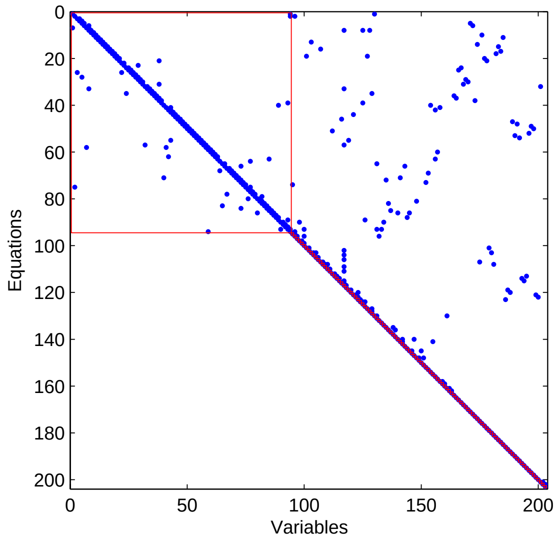


Figure 5: Structure of the 203 equations in the considered computation sequence, with respect to the 203 unknown variables. The SCCs of the structure, corresponding to the elements in the computation sequence, are marked with squares. The large SCC contains 102 equations.

6.5 Realization of a Residual Generator for the Diesel Engine

The purpose of this section is to briefly show how a residual generator for the diesel engine is constructed from a computation sequence obtained with the proposed method.

Properties of the Computation Sequence

The considered computation sequence originates from an MSO set containing in total 204 equations, 203 unknown variables, and 8 known variables. Thus, the computation sequence contains 203 equations and 203 unknown variables. In total 33 residual generators were found in the MSO class to which the MSO set belongs. All 33 residual generators were found with configuration SDI of findComputationSequence.

The computation sequence contains 102 elements. All elements but the last one contains one equation and one variable. The last element contains 102 equations and 102 variables and corresponds to a SCC of size 102. The structure of the 203 equations contained in the computation sequence, with respect to the 203 unknown variables, is shown in Figure 5. The SCCs of the structure, corresponding to the elements in the computation sequence, marked with squares in Figure 5.

The residual equation used in the residual generator, i.e., the equation removed from the MSO set when the corresponding computation sequence was found, compares the measured and computed pressure in the intake manifold of the diesel engine.

Properties of the BLT Semi-Explicit DAE System

The BLT semi-explicit DAE system obtained from the computation sequence contains 102 blocks and has the following form

$$\begin{aligned}
\mathbf{w}_1 &= \mathbf{h}_1\left(\mathbf{y}\right)\\
\mathbf{w}_2 &= \mathbf{h}_2\left(\mathbf{w}_1, \mathbf{y}\right)\\
&\vdots\\
\mathbf{w}_{64} &= \mathbf{h}_{64}\left(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{63}, \mathbf{y}\right)\\
\mathbf{w}_{65} &= \mathbf{h}_{65}\left(\dot{\mathbf{w}}_{64}, \mathbf{w}_1, \dots, \mathbf{w}_{64}, \mathbf{y}\right)\\
\mathbf{w}_{66} &= \mathbf{h}_{66}\left(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{65}, \mathbf{y}\right)\\
&\vdots\\
\mathbf{w}_{76} &= \mathbf{h}_{76}\left(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{75}, \mathbf{y}\right)\\
\mathbf{w}_{77} &= \mathbf{h}_{77}\left(\dot{\mathbf{w}}_{76}, \mathbf{w}_1, \dots, \mathbf{w}_{76}, \mathbf{y}\right)\\
\mathbf{w}_{78} &= \mathbf{h}_{78}\left(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{77}, \mathbf{y}\right)\\
&\vdots\\
\mathbf{w}_{100} &= \mathbf{h}_{100}\left(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{99}, \mathbf{y}\right)\\
\dot{\mathbf{z}}_{101} &= \mathbf{g}_{101}\left(\mathbf{w}_1, \dots, \mathbf{w}_{101}, \mathbf{y}\right)\\
\mathbf{w}_{101}^1 &= \mathbf{h}_{101}^1\left(\mathbf{z}_{101}, \mathbf{w}_1, \dots, \mathbf{w}_{100}, \mathbf{y}\right)\\
\mathbf{w}_{101}^2 &= \mathbf{h}_{101}^2\left(\mathbf{z}_{101}, \mathbf{w}_1, \dots, \mathbf{w}_{100}, \mathbf{w}_{101}^1, \mathbf{y}\right)\\
&\vdots\\
\mathbf{w}_{101}^{99} &= \mathbf{h}_{101}^{99}\left(\mathbf{z}_{101}, \mathbf{w}_1, \dots, \mathbf{w}_{100}, \mathbf{w}_{101}^1, \dots, \mathbf{w}_{101}^{98}, \mathbf{y}\right),
\end{aligned}\tag{29}$$

where $\mathbf{w}_{101} = \left(\mathbf{w}_{101}^1, \mathbf{w}_{101}^2, \dots, \mathbf{w}_{101}^{99}\right)$, and $\mathbf{z}_{101}$ is of dimension three and all $\mathbf{w}_i$, $\mathbf{w}_i^j$ of dimension one. The largest block, denoted 101 in (29), is a semi-explicit DAE of index one with three differential equations with variables $\mathbf{z}_{101}$ and 99 algebraic equations with variables $\mathbf{w}_{101}^1, \dots, \mathbf{w}_{101}^{99}$, corresponding to a differential loop and a SCC of size 102. Since the block is a semi-explicit DAE of index one, integral causality is used in this block. In two of the blocks, denoted 66 and 77 in (29), derivative causality is used. The remaining blocks, denoted 1 - 65, 67 - 76, and 78 - 100 correspond to algebraic equations. In total, the BLT semi-explicit DAE system contains five differential equations and 198 algebraic equations.

Implementation Issues

The residual generator, i.e., the obtained BLT semi-explicit DAE system and the residual equation, was implemented in Matlab. To compute the values of the unknown variables, the approach described in Section 3.1 was used. To solve the resulting explicit ODE, Euler forward with fixed step-size was utilized. All state variables in the residual generators represent physical quantities, hence initial conditions were easy to obtain from the available measurements.

Approximate Differentiation In the two blocks where derivative causality is used, 66 and 77 in (29), derivatives of variables computed in previous blocks had to be computed. By propagating the two differentiated variables through equations in earlier blocks of the obtained BLT semi-explicit DAE system, the differentiated variables could be expressed as derivatives of known variables only, see Section 3.2. The known variables that had to be differentiated were measurements of the pressure in the exhaust manifold, and the rotational speed of the turbo turbine.

The differentiation tool, i.e., the method for differentiation of known variables, used in this case study was a sliding-window least square polynomial fit approach. By finding a linear approximation, in a least square sense, to a set of consecutive measurements, referred to as a window, an approximation of the first-order derivative of the measured signal in the window can be obtained as the slope of the linear approximation, see, e.g., Barford et al. (1999). This approach was used since it is simple and straight-forward to implement, and gave good results. An implementation was done in Matlab, a window-size of 40 measurements, 20 past and 20 future, was used.

Results

Real measurements of the known variables in the engine model were collected by driving a truck on the road. Two sets of measurements were collected, one with a fault-free engine and one with an implemented fault. The implemented fault was a constant bias in the sensor measuring the pressure in the intake manifold of the diesel engine.

The residual generator was run off-board by using the collected measurements. The residual was then low-pass filtered to remove some measurement noise and finally scaled. In Figure 6, the resulting residual is shown. During the first 100 seconds, the measurements are fault-free. The remaining time, the measurements contain the implemented bias fault. It is obvious that the residual can be used to detect the injected fault.

7 Conclusions

We have in Section 1 concluded that it is important that there is a large selection of different candidate residual generators to choose between when designing diagnosis systems. In this spirit we have in this paper presented a method for deriving residual generators with the key property that it is able to find a large number of different residual generators. This property is firstly due to the fact that the method belongs to a class of methods that we refer to as sequential residual generation. This class of methods has in earlier works been shown to be powerful for real non-linear systems (Dustegor et al., 2004; Izadi-Zamanabadi, 2002; Cocquempot et al., 1998; Svärd and Wassén, 2006; Hansen and Molin, 2006). Secondly, which is the key contribution of the paper, we have extended these earlier methods by handling mixed causality and also, in a systematic manner, equation sets containing differential and algebraic loops.

The method has been presented as an algorithm utilizing an assumed given toolbox of, e.g., algebraic equation solvers. We have proven, in Theorem 1, that the algorithm really finds residual generators and, in Theorems 3 and 4, that the residual generators, or

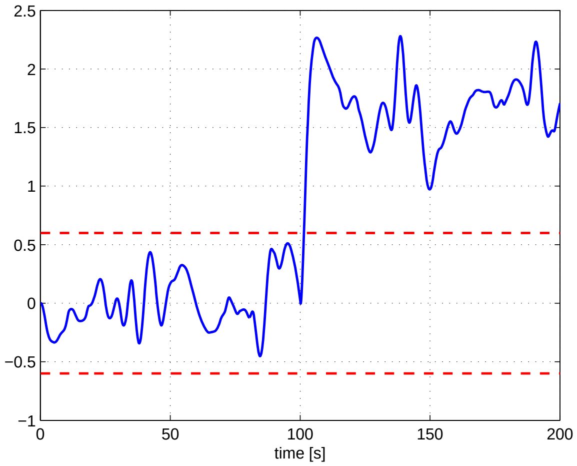


Figure 6: The residual obtained from the constructed residual generator. No fault is present the first 100 seconds. During the remaining 100 seconds, there is a bias fault in the sensor measuring the pressure in the intake manifold. The dashed lines suggests how thresholds could be chosen in order to detect the fault.

rather sequential residual generators, found are *proper*. Properness guarantees that the residual generator is not containing unnecessary computations and that computations are performed from as small equation sets as possible. We have also proven, in Theorem 2, that proper sequential residual generators are always found within MSO sets. This fact has been utilized in the algorithm since there is no need to look for sequential residual generators in other equation sets than MSO sets. Furthermore, this theorem provides a link between structural and analytical methods without the use of any assumptions of generic equations, such as in, e.g., Krysander et al. (2008).

In the empirical study in Section VI, we have evaluated our method on models of two real automotive Systems. The results obtained are compared to results from the special cases of using solely differential or integral causality, or only handling scalar equations. It is evident that our more general method outperforms the other alternatives. Since the two systems have quite different characteristics, e.g., in the number of redundant sensors, we believe that these results are representative also for a larger class of systems.

Acknowledgment

This work was sponsored by Scania CV AB and VINNOVA (Swedish Governmental Agency for Innovation Systems).

A Proofs of Theorems and Lemmas

Proof of Lemma 1. Consider an element $(V_i, E_i) \in \mathcal{C}$, and let $\bar{E}'_i$ denote the set of equations obtained when $\mathcal{T}$ is called with arguments V_i and E_i. It then holds that $\bar{E}' = \bar{E}'_1 \cup \bar{E}'_2 \cup \ldots \cup \bar{E}'_k$. Given $\mathbf{y}$, let $\tilde{\mathbf{x}}$ be an arbitrary solution to $\bar{E}'$, i.e., a trajectory fulfilling every equation $e_i \in \bar{E}'$. Trivially, $\tilde{\mathbf{x}}$ also is a solution to the equations in every $\bar{E}'_i$, since $\bar{E}'_i \subseteq \bar{E}'$. Assumption 1 then implies that $\tilde{\mathbf{x}}$ is a unique solution and also a solution to every E_i, and hence to $E_1 \cup E_2 \cup \ldots \cup E_k$. By taking an arbitrary solution to $E_1 \cup E_2 \cup \ldots \cup E_k$ and applying the same arguments as above, it can be shown that this solution is unique and also satisfies $\bar{E}'$, which completes the proof. □

Proof of Theorem 1. Consider the model $M(E, X, Y)$ and assume that $\tilde{\mathbf{y}} \in \mathcal{O}(M)$. Due to the definition of $\mathcal{O}(M)$ in (4), we know that given $\tilde{\mathbf{y}}$ there exists at least one trajectory of the variables in X that satisfies the equations in E. Since describing $E_1 \cup E_2 \cup \ldots \cup E_k \subseteq E$, it holds that the trajectory $\tilde{\mathbf{y}}$ also belongs to the observation set of the sub-model of $M(E, X, Y)$ given by $E_1 \cup E_2 \cup \ldots \cup E_k$, i.e., the equation set contained in the computation sequence $\mathcal{C}$. Hence, given $\tilde{\mathbf{y}}$, there exists a trajectory $\tilde{\mathbf{x}}$ of the variables in $\text{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)$ that satisfies $E_1 \cup E_2 \cup \ldots \cup E_k$. By Lemma 1 we know that $\tilde{\mathbf{x}}$ is a unique solution that also satisfies the equations of the BLT semi-explicit DAE system obtained by sequentially applying the tool $\mathcal{T}$ to the computation sequence $\mathcal{C}$.

As said in Section 3.1, a BLT semi-explicit DAE system can be transformed to an explicit ODE, with the exception that the ODE will contains derivatives of known variables. Furthermore, after the discussion in Section 3.2, that an explicit ODE always can be solved if initial conditions are available. From this it follows that given $\tilde{\mathbf{y}}$, consistent initial conditions of the states in the BLT semi-explicit DAE system, i.e., $\mathbf{z}_i$ in (7), and the ability the compute all needed derivatives, the trajectory $\tilde{\mathbf{x}}$ can be computed from the BLT semi-explicit DAE system. Since $e_i \in E \setminus E_1 \cup E_2 \cup \ldots \cup E_k$ and $\text{var}_X(e_i) \subseteq X' \subseteq \text{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)$, the trajectory $\tilde{\mathbf{x}}$ will also satisfy e_i. We then have that $f_i(\dot{\tilde{\mathbf{x}}}, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = 0$. Hence, with $r = f_i(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}), \tilde{\mathbf{y}} \in \mathcal{O}(M)$ implies $r = 0$ and we can use r as residual. Thus the BLT semi-explicit DAE system obtained from the computation sequence $\mathcal{C}$ with $\mathcal{T}$, together with e_i is a residual generator for $M(E, X, Y)$. □

Some important properties of a computation sequence, used in sub-sequential proofs, is given by the following lemma.

Lemma 2. *Let $\mathcal{C} = ((V_1, E_1), (V_2, E_2), \ldots (V_k, E_k))$ be a computation sequence for the variables X' with the AE tool $\mathcal{T}$, then $\{\text{unDiff}(V_i)\}$ is pairwise disjoint and*

$$\text{unDiff}(V_1 \cup V_2 \cup \ldots \cup V_k) = \text{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k).$$

Proof. From Definition 3, we have that a system in BLT semi-explicit DAE form can be obtained by sequentially calling $\mathcal{T}$ with arguments V_i and E_i for every $(V_i, E_i) \in \mathcal{C}$. From this fact, it follows that each variable $x_j \in \text{unDiff}(V_i)$ is present in some vector $\mathbf{z}_k$ or $\mathbf{w}_l$ in the obtained BLT semi-explicit DAE system. Since the set of all vectors of known variables in a BLT semi-explicit DAE system by Definition 2 is pairwise disjoint, it follows that $\{\text{unDiff}(V_i)\}$ is pairwise disjoint and we have shown the first claim. For the second claim, we start by noting that $V_i \subseteq \mathrm{var}_X(E_i) \cup \mathrm{var}_D(E_i)$ due to Definition 3. Since a system in BLT semi-explicit DAE form can be obtained from $\mathcal{C}$ and, according to Lemma 1, the solution sets of $E_1 \cup E_2 \cup \ldots \cup E_k$ and the BLT semi-explicit DAE system, with respect to $V_1 \cup V_2 \cup \ldots \cup V_k$, are equal and unique, it holds that each unknown variable in $E_1 \cup E_2 \cup \ldots \cup E_k$, differentiated or undifferentiated, must be present in some V_i. From this fact and by the definitions of the operators $\mathrm{unDiff}\,()$ and $\mathrm{var}_X()$, it must also hold that $\mathrm{unDiff}\,(V_1 \cup V_2 \cup \ldots \cup V_k) = \mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)$. □

For the next proof, we need some additional graph theoretical concepts, see, e.g., Asratian et al. (1998); Murota (1987), therefore consider the bipartite graph $G = (E, X, A)$ describing the structure of E with respect to X, see Section 2.2. A *path* on the graph G is a sequence of distinct vertices $v_1, v_2, \ldots, v_n$ such that $(v_i, v_{i+1}) \in A$ and $v_i \in E \cup X$. An *alternating path* is a path in which the edges belong alternatively to a matching and not to the matching. A vertex is said to be *free*, if it is not an endpoint of an edge in a matching.

Proof of Theorem 2. In this proof we will use a characterization of an MSO set given in Krysander et al. (2008), saying that an equation set E is an MSO set if and only if E is a *Proper Structurally Over-determined* (PSO) set and E contains one redundant equation. Furthermore, an equation set E is a PSO set if $E = E^+$, where E^+ is the structurally over-determined part obtained from the DM-decomposition, recall Section 2.3, or equivalently the equations $e \in E$ such that, for any maximal matching, there exists an alternating path between at least one free equation and e.

Returning to our case, we must show that $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$ is a PSO set and contains one redundant equation, with respect to the variables $\mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)$. We begin with the second property, i.e., that $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$ contains a redundant equation. Since $\mathcal{S} = (\mathcal{T}\,(\mathcal{C})\,, e_i)$ is a proper sequential residual generator, it follows from Definition 7 that $\mathcal{C}$ is a minimal and irreducible computation sequence for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$. If we let

$$\mathcal{C} = ((V_1, E_1)\,, (V_2, E_2)\,, \ldots, (V_k, E_k))\,, \tag{30}$$

we have from Definition 3 that a system in BLT semi-explicit DAE form is obtained by sequentially calling the AE tool $\mathcal{T}$ with arguments V_i and E_i for every $(V_i, E_i) \in \mathcal{C}$. This and Assumption 2, implies that $|V_i| = |E_i|$ for every $(V_i, E_i) \in \mathcal{C}$ and hence $\sum_{i=1}^{k} |V_i| = \sum_{i=1}^{k} |E_i|$. By the definition of the operator $\mathrm{unDiff}\,()$ in (15), we can conclude that $|V_i| = |\mathrm{unDiff}\,(V_i)|$ and therefore it also holds that $\sum_{i=1}^{k} |\mathrm{unDiff}\,(V_i)| = \sum_{i=1}^{k} |E_i|$. By Lemma 2 we have that $\{\mathrm{unDiff}\,(V_i)\}$ is pairwise disjoint which implies that $\sum_{i=1}^{k} |\mathrm{unDiff}\,(V_i)| = |\mathrm{unDiff}\,(V_1) \cup \mathrm{unDiff}\,(V_2) \cup \ldots \cup \mathrm{unDiff}\,(V_k)| = |\mathrm{unDiff}\,(V_1 \cup V_2 \cup \ldots \cup V_k)|$. Definition 3 states that also $\{E_i\}$ is pairwise disjoint and therefore $|E_1 \cup E_2 \cup \ldots \cup E_k| = \sum_{i=1}^{k} |E_i|$. Thus, it holds that $|\mathrm{unDiff}\,(V_1 \cup V_2 \cup \ldots \cup V_k)| = |E_1 \cup E_2 \cup \ldots \cup E_k|$. By Lemma 2, we have that $\mathrm{unDiff}\,(V_1 \cup V_2 \cup \ldots \cup V_k) = \mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)$ and therefore it also holds that $|E_1 \cup E_2 \cup \ldots \cup E_k| = |\mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)|$, i.e., $E_1 \cup E_2 \cup \ldots \cup E_k$ contains as many equations as unknowns. Since $\mathcal{C}$ is a computation sequence for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$, we have from Definition 3 that $\mathrm{var}_X(e_i) \subseteq \mathrm{unDiff}\,(V_1 \cup V_2 \cup \ldots \cup V_k) = \mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)$, where the last equality follows from Lemma 2, implying that adding e_i to

$E_1 \cup E_2 \cup \ldots \cup E_k$ will not introduce any new unknown variables, i.e., e_i is redundant. Hence, the equation set $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$ contains one more equation than unknown variables, since $|E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i| = |E_1 \cup E_2 \cup \ldots \cup E_k| + |e_i| = |\mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)| + 1$.

We will now show that $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$ is a PSO set with respect to $\mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i)$. To show this, we must show that for any maximum matching on the bipartite graph describing the structure of $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$, with respect to $\mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i)$, there exists an alternating path between a free equation and every equation in $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$. We start by constructing a maximum matching and finding a free equation. Consider the computation sequence $\mathcal{C}$ described by (30) and recall that $\mathcal{C}$, given by (30), is a minimal and irreducible computation sequence for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$. The irreducibility of $\mathcal{C}$ implies that for each element $(V_i, E_i) \in \mathcal{C}$, it holds that the structure of E_i with respect to unDiff (V_i) corresponds to a SCC. To see this, assume that (V_i, E_i) not corresponds to a SCC. This implies that it is possible to partition V_i and E_i into $V_i = V_{i1} \cup V_{i2} \cup \ldots \cup V_{is}$ and $E_i = E_{i1} \cup E_{i2} \cup \ldots \cup E_{is}$ so that

$$\mathcal{C}' = ((V_1, E_1), \ldots, (V_{i1}, E_{i1}), \ldots, (V_{is}, E_{is}), \ldots, (V_k, E_k)),$$

is also a computation sequence for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$, due to Assumption 3. This contradicts the irreducibility of $\mathcal{C}$ and hence (V_i, E_i) must be a SCC. From this property it follows, by the definition of a SCC, that there exists a maximum matching Γ_i on the bipartite graph the structure of E_i with respect to unDiff (V_i). This implies that a maximum matching, let it be denoted Γ, in the structure of $E_1 \cup E_2 \cup \ldots \cup E_k$ with respect to unDiff $(V_1 \cup V_2 \cup \ldots \cup V_k)$ can be constructed as $\Gamma = \bigcup_i^k \Gamma_i$, see, e.g., Murota (1987). By Lemma 2, we have that unDiff $(V_1 \cup V_2 \cup \ldots \cup V_k) = \mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)$ and therefore Γ is also a maximum matching in the structure of $E_1 \cup E_2 \cup \ldots \cup E_k$ with respect to $\mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k)$. In the first part of this proof, we concluded that the equation e_i is redundant and therefore Γ is also a maximum matching on the structure of $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$ with respect to $\mathrm{var}_X(E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i)$ and e_i is a free equation, since it is not contained in Γ.

Since it trivially exists a path between e_i and e_i, it is sufficient to show that there exists an alternating path between the free equation e_i and every equation in $E_1 \cup E_2 \cup \ldots \cup E_k$. Due to the fact that each $(V_i, E_i) \in \mathcal{C}$ corresponds to a SCC, there exists an alternating path between any two vertices, i.e., equations or variables, in the bipartite graph describing the structure of E_i with respect to unDiff (V_i), see, e.g., Asratian et al. (1998). Moreover, the minimality of $\mathcal{C}$ implies that for $(V_k, E_k) \in \mathcal{C}$ there exists at least one variable $x_m \in$ unDiff (V_k) such that $x_m \in \mathrm{var}_X(e_i)$, since otherwise $\mathcal{C}' = \mathcal{C} \setminus (V_k, E_k)$ is a computation sequence for $\mathrm{var}_X(e_i)$ and $\mathcal{C}$ is not minimal. With the same argument, we have that for $(V_i, E_i) \in \mathcal{C}$, $i = 1, 2, \ldots, k-1$, there exists at least one variable $x_m \in$ unDiff (V_i) such that either $x_m \in \mathrm{var}_X(e_i)$, or else $x_m \in \mathrm{var}_X(E_j)$ where $\left(V_j, E_j\right) \in \mathcal{C}$ and $j \in \{i+1, i+2, \ldots, k\}$. This means that there exists an alternating path between at least one variable in each $(V_i, E_i) \in \mathcal{C}$ to e_i, either directly or via one or several other $\left(V_j, E_j\right) \in \mathcal{C}$. Thus, there exists an alternating path between e_i and every equation in $E_1 \cup E_2 \cup \ldots \cup E_k$. We have by this shown that $E_1 \cup E_2 \cup \ldots \cup E_k \cup e_i$ is a PSO set. □

The proof of Theorem 3 is based on the following lemma.

Lemma 3. *Let $\bar{E} \subseteq E$ be an MSO set, $\mathcal{T}$ an AE tool, $X' = \mathrm{var}_X(\bar{E})$, and $E' = \bar{E} \setminus e_i$, where $e_i \in \bar{E}$. A minimal and irreducible computation sequence*

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \ldots, (V_k, E_k)),$$

for X' with $\mathcal{T}$, where $E_i \subseteq \bar{E}$, is also a minimal and irreducible computation sequence for $\mathrm{var}_X(e_i)$ *with $\mathcal{T}$.*

Proof. Assume that $\mathcal{C}$ is a minimal and irreducible computation sequence for X' with $\mathcal{T}$. First of all, since $e_i \in \bar{E}$ and $X' = \mathrm{var}_X(\bar{E})$ it trivially holds that $\mathrm{var}_X(e_i) \subseteq X'$ and hence $\mathcal{C}$ is a computation sequence for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$. As well, it directly follows from Definition 6 that $\mathcal{C}$ is an irreducible computation sequence for any subset of X', in particular $\mathrm{var}_X(e_i)$. To show that $\mathcal{C}$ also is a minimal computation sequence for $\mathrm{var}_X(e_i)$, assume that there exists a computation sequence $\mathcal{C}' \subset \mathcal{C}$ for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$. Let $\bar{E}'$ and $\bar{X}' = \mathrm{var}_X(\bar{E}')$ denote the equations and variables, contained in the elements of $\mathcal{C}'$ and note that since $\mathcal{C}' \subset \mathcal{C}$, it holds that $\bar{E}' \subset \bar{E}$. By the argumentation in the proof to Theorem 2, we can conclude that $|\bar{E}'| = |\bar{X}'|$, i.e., $\bar{E}$ contains as many equations as unknowns. Since $\mathcal{C}'$ is a computation sequence for $\mathrm{var}_X(e_i)$, it must hold that $\mathrm{var}_X(e_i) \subseteq \bar{X}'$. This means that $\bar{E}' \cup e_i$ is a structurally over-determined set of equations with respect to $\bar{X}'$, which shows that there exists a proper structurally over-determined subset of $\bar{E}$. This contradicts the fact that $\bar{E}$ is an MSO set, and hence there can not exist a computation sequence $\mathcal{C}' \subset \mathcal{C}$ for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$. Thus, $\mathcal{C}$ is a minimal computation sequence for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$. □

Proof of Theorem 3. Consider the model $M(E, X, Y)$ and let $(\mathcal{T}(\mathcal{C}), e_i) \in R$. Due to line 9 in findResidualGenerators, we can conclude that $\mathcal{C}$ is non-empty. Let

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \ldots, (V_k, E_k)),$$

where $E_i \subseteq E'$, be the minimal and irreducible computation sequence for X' with $\mathcal{T}$, returned by the function findCompuatationSequence on line 8. Due to lines 3-7, we have that $E' = \bar{E} \setminus e_i$ and $X' = \mathrm{var}_X(\bar{E})$, where $\bar{E} \subseteq E$ is an MSO set and $e_i \in \bar{E} \setminus E'$. Lemma 3 then implies that $\mathcal{C}$ also is a minimal and irreducible computation sequence for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$. Now note that since $e_i \in \bar{E} \setminus E'$ and it holds that $\bar{E} \subseteq E$, we have that $e_i \in E \setminus E'$. Trivially, since $X' = \mathrm{var}_X(\bar{E})$ and $X' \subseteq X$ it also holds that $\mathrm{var}_X(e_i) \subseteq X' \subseteq X$. Thus the computation sequence $\mathcal{C}$ for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$ and the equation e_i fulfills the prerequisites of Theorem 1. Hence, since all initial conditions are known and all needed derivatives can be computed, we can by Theorem 1 conclude that the BLT semi-explicit DAE system obtained from $\mathcal{C}$ with $\mathcal{T}$ and e_i is a residual generator for $M(E, X, Y)$. Thus, $(\mathcal{T}(\mathcal{C}), e_i)$ is a sequential residual generator. Since, in fact, $\mathcal{C}$ is a minimal and irreducible computation sequence for $\mathrm{var}_X(e_i)$ with $\mathcal{T}$, $(\mathcal{T}(\mathcal{C}), e_i)$ is a proper sequential residual generator. □

Proof of Theorem 4. On line 3 in findComputationSequence the SCCs of the structure of E' with respect to X' are computed. If we assume that the structure contains s SCCs, the ordered set returned by the function findAllSCC can be written as

$$S = ((E_1, X_1), (E_2, X_2), \ldots, (E_s, X_s)), \tag{31}$$

where each element $(E_i, X_i) \in S$ corresponds to a SCC of the structure of E' with respect to X'. Note that since E' is just-determined with respect X', the SCCs of the structure of E' with respect X' are unique, see Section 2.3. As said in Section 5.3, we assume that the SCCs in S are ordered according to Figure 1. Note that this ordering implies the important property

$$\mathrm{var}_X(E_i) \cap \{X_{i+1} \cup X_{i+2} \cup \ldots \cup X_s\} = \varnothing, \tag{32}$$

for $i = 1, 2, \ldots, s-1$. On lines 6-8, the variables in X_i are partitioned into differentiated variables Z_i and undifferentiated variable W_i, i.e., $X_i = \mathrm{unDiff}\,(Z_i) \cup W_i$, where Z_i contains variables that appear as differentiated in some equation in E_i. On lines 12-14, a corresponding partitioning of the equations in E_i into $E_i = E_{Z_i} \cup E_{W_i}$ is done, where E_{Z_i} are equations that contain any of the differentiated variables Z_i, and E_{W_i} are equations that do not contain any of the differentiated variables Z_i, but may contain variables from $\mathrm{unDiff}\,(Z_i)$. Now note that, due to the assumptions regarding the model in Section 2, each equation in E_{Z_i} contains only one differentiated, which furthermore only is present in that equation. This means first of all that E_{Z_i} is just-determined with respect to the variables in Z_i, and second that the structure of E_{Z_i} with respect to Z_i only contains SCCs of size one. On line 14, these SCCs are computed. Assuming that the structure contains s_i SCCs, the ordered set returned by findAllSCC on line 14 can be written as

$$S_{Z_i} = \left(\left(Z_i^1, E_{Z_i}^1\right), \left(Z_i^2, E_{Z_i}^2\right), \ldots, \left(Z_i^{s_i}, E_{Z_i}^{s_i}\right) \right). \tag{33}$$

Due to line 23, we know that the equation set E_{W_i} is just-determined with respect to W_i, and hence the structure of E_{W_i} with respect to W_i can be uniquely partitioned into SCCs. On line 24 these SCCs are computed and as above, the ordered set of SCCs can be written as

$$S_{W_i} = \left(\left(W_i^1, E_{W_i}^1\right), \left(W_i^2, E_{W_i}^2\right), \ldots, \left(W_i^{p_i}, E_{W_i}^{p_i}\right) \right). \tag{34}$$

Furthermore, as in the case with the set S in (31), the ordering of the SCCs in S_{W_i} implies that

$$\mathrm{var}_X(E_{W_i}^j) \cap \left\{ W_i^{j+1} \cup W_i^{j+2} \cup \ldots \cup W_i^{p_i} \right\} = \varnothing, \tag{35}$$

for $j = 1, 2, \ldots, p_i - 1$. From the discussion above, we have that a non-empty $\mathcal{C}$ returned by findComputationSequence have the form

$$\begin{aligned} \mathcal{C} = \Big(& \left(Z_1^1, E_{Z_1}^1\right), \left(Z_1^2, E_{Z_1}^2\right), \ldots, \left(Z_1^{s_1}, E_{Z_1}^{s_1}\right), \\ & \left(W_1^1, E_{W_1}^1\right), \left(W_1^2, E_{W_1}^2\right), \ldots, \left(W_1^{p_1}, E_{W_1}^{p_1}\right), \ldots, \\ & \left(Z_2^1, E_{Z_2}^1\right), \left(Z_2^2, E_{Z_2}^2\right), \ldots, \left(Z_2^{s_2}, E_{Z_2}^{s_2}\right), \\ & \left(W_2^1, E_{W_2}^1\right), \left(W_2^2, E_{W_2}^2\right), \ldots, \left(W_2^{p_2}, E_{W_2}^{p_2}\right), \ldots, \\ & \left(Z_s^1, E_{Z_s}^1\right), \left(Z_s^2, E_{Z_s}^2\right), \ldots, \left(Z_s^{s_s}, E_{Z_s}^{s_s}\right), \\ & \left(W_s^1, E_{W_s}^1\right), \left(W_s^2, E_{W_s}^2\right), \ldots, \left(W_s^{p_s}, E_{W_p}^{p_s}\right) \Big), \end{aligned} \tag{36}$$

where every $\left(Z_i^j, E_{Z_i}^j\right) \in \mathcal{C}$ and $\left(W_i^j, E_{W_i}^j\right) \in \mathcal{C}$ corresponds to a SCC.

We will now utilize Definition 3 to show that the the ordered set $\mathcal{C}$ in (36) is a computation sequence for X' with $\mathcal{T}$. First note that $Z_i^j \subseteq \text{var}_D(E_{Z_i}^j)$ and $W_i^j \subseteq \text{var}_X(E_{W_i}^j)$. When the structure of a just-determined equation set with respect to a set of variables is decomposed into its SCCs, unique partitions of the equation and variable sets are also obtained, see for example Dulmage and Mendelsohn (1958) and Figure 1 for an illustration. From this fact it follows that every equation in E' is present in some E_i in (31) only once. When the equations in E_i are split into differential equations E_{Z_i} and algebraic equations E_{W_i} on line 13, it is guaranteed that $E_{Z_i} \cap E_{W_i} = \varnothing$. Moreover, again due to the fact that a decomposition into SCCs gives an unique partition of the equation and variable set, we have that every equation in E_{Z_i} is present in some equation set $E_{Z_i}^j$ in (33) only once and that every equation in E_{W_i} is present in some $E_{W_i}^j$ in (34) only once. Thus, we can conclude that each equation in E' is contained in only one equation set in $\mathcal{C}$, that is, all equation sets in $\mathcal{C}$ are disjoint. Hence, the ordered set $\mathcal{C}$ fulfills the prerequisites in Definition 3. According to conditions 1) and 2) in Definition 3, $\mathcal{C}$ is a computation sequence for X' with $\mathcal{T}$ if

$$X' \subseteq \bigcup_{i=1}^{s} \left(\bigcup_{j=1}^{s_i} \text{unDiff}\left(Z_i^j\right) \cup \bigcup_{j=1}^{p_i} W_i^j \right) \tag{37}$$

and a system in BLT semi-explicit DAE form is obtained by sequentially calling the tool $\mathcal{T}$, with arguments Z_i^j and $E_{Z_i}^j$ for every element $\left(Z_i^j, E_{Z_i}^j\right) \in \mathcal{C}$, and with arguments W_i^j and $E_{W_i}^j$ for every element $\left(W_i^j, E_{W_i}^j\right) \in \mathcal{C}$.

We start by showing condition 1), i.e., (37). From the fact mentioned above that a decomposition of a structure into its SCCs also induces a partitioning of the corresponding equation and variable sets, it follows that every variable in X' is present in some X_i in (31). That is, we have that $X' = \bigcup_i^s X_i$. When the variables in X_i are split into differentiated variables Z_i and undifferentiated variables W_i, it holds that $X_i = \text{unDiff}\,(Z_i) \cup W_i$. In addition, it holds that every variable in Z_i is present in some variable set Z_i^j in (33) and that every variable in W_i is present in some W_i^j in (34), so that $Z_i = \bigcup_{j=1}^{s_i} Z_i^j$ and $W_i = \bigcup_{j=1}^{p_i} W_i^j$. Hence,

$$\begin{aligned} X' &= \bigcup_{i}^{s} X_i = \bigcup_{i}^{s} \left(\text{unDiff}\,(Z_i) \cup W_i\right) \\ &= \bigcup_{i}^{s} \left(\text{unDiff}\left(\bigcup_{j=1}^{s_i} Z_i^j \right) \cup \bigcup_{j=1}^{p_i} W_i^j \right) \\ &= \bigcup_{i}^{s} \left(\bigcup_{j=1}^{s_i} \text{unDiff}\left(Z_i^j\right) \cup \bigcup_{j=1}^{p_i} W_i^j \right), \end{aligned} \tag{38}$$

where the last equality trivially follows from the definition of unDiff () in (15). The property (37) and thus condition 1) has then been verified.

Condition 2) of Definition 3 will now be verified, that is, that $\mathcal{C}$ can be used to obtain a system in BLT semi-explicit DAE form. Consider an element $\left(Z_i^j, E_{Z_i}^j\right) \in \mathcal{C}$. Since $E_{Z_i}^j \subseteq E_{Z_i} \subseteq E_i$, and we have that $(X_i, E_i) \in S$, the property (32) implies that

$$\mathrm{var}_X(E_{Z_i}^j) \cap \{X_{i+1} \cup X_{i+2} \cup \ldots \cup X_s\} = \varnothing, \tag{39}$$

for $i = 1, 2, \ldots, s-1$. From lines 17-21 in the algorithm, it follows that the AE tool $\mathcal{T}$ can be used to solve the equations in $E_{Z_i}^j$ for the variables in Z_i^j. Since we have assumed that each differential equation contains at most one differentiated variable and (39) holds, we can use $\left(Z_i^j, E_{Z_i}^j\right) \in \mathcal{C}$ and the AE tool $\mathcal{T}$ to obtain

$$\dot{\mathbf{z}}_i^j = \mathbf{g}_i^j\left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \mathbf{y}\right), \tag{40}$$

where $\dot{\mathbf{z}}_i^j$ is a vector of the variables in Z_i^j, $\mathbf{x}_k$ a vector of the variables in X_k, $\mathbf{y}$ a vector of the known variables in E', and $\mathbf{g}_i^j$ a function returned by $\mathcal{T}$ when the arguments are Z_i^j and $E_{Z_i}^j$. From the elements $\left(Z_i^j, E_{Z_i}^j\right) \in \mathcal{C}$, $j = 1, 2, \ldots, s_i$, we can thus, by using (40) and also that $X_i = \mathrm{unDiff}\left(Z_i\right) \cup W_i$, obtain

$$\dot{\mathbf{z}}_i = \mathbf{g}_i\left(\mathbf{z}_1, \mathbf{z}_1, \ldots, \mathbf{z}_i, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_i, \mathbf{y}\right), \tag{41}$$

where $\mathbf{z}_i = \left(\mathbf{z}_i^1, \mathbf{z}_i^2, \ldots, \mathbf{z}_i^{s_i}\right)$ and a vector of the variables in Z_i, $\mathbf{w}_i$ a vector of the variables in W_i, $\mathbf{y}$ a vector of the known variables in E', and $\mathbf{g}_i = \left(\mathbf{g}_i^1, \mathbf{g}_i^2, \ldots, \mathbf{g}_i^{s_i}\right)$.

Now instead consider an element $\left(W_i^j, E_{W_i}^j\right) \in \mathcal{C}$. Since also $\left(W_i^j, E_{W_i}^j\right) \in S_{W_i}$, where S_{W_i} is given by (34) the property (35) holds. Since $E_{W_i}^j \subseteq E_{W_i} \subseteq E_i$, and $(X_i, E_i) \in S$ we also have that

$$\mathrm{var}_X(E_{W_i}^j) \cap \{X_{i+1} \cup X_{i+2} \cup \ldots \cup X_s\} = \varnothing, \tag{42}$$

for $i = 1, 2, \ldots, s-1$. By using that the AE tool $\mathcal{T}$ can solve $E_{W_i}^j$ for W_i^j due to lines 27-31, that $X_i = \mathrm{unDiff}\left(Z_i\right) \cup W_i$ and $\mathrm{var}_D(E_{W_i}) \cap Z_i = \varnothing$ due to lines 6-8 and 12-14, and then utilize (35) and (42), we can obtain

$$\mathbf{w}_i^j = \mathbf{h}_i^j\left(\dot{\mathbf{w}}_1, \ldots, \dot{\mathbf{w}}_{i-1}, \mathbf{z}_1, \ldots, \mathbf{z}_i, \mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \mathbf{w}_i^1, \ldots, \mathbf{w}_i^{j-1}, \mathbf{y}\right), \tag{43}$$

from $\left(W_i^j, E_{W_i}^j\right) \in \mathcal{C}$, where $\mathbf{w}_i^j$ is a vector of the variables in W_i^j, $\mathbf{z}_i$ a vector of the variables in Z_i, and $\mathbf{h}_i^j$ a function returned by $\mathcal{T}$ when the arguments are W_i^j and $E_{W_i}^j$. Note that the absence of vectors $\dot{\mathbf{z}}_i$ in (43) is a direct implication of the assumption that each differentiated variable is present in only one equation in the original model and therefore also in the BLT semi-explicit DAE system. Since $\dot{\mathbf{z}}_i$, obviously, is present in (41), it can not be present in (43).

By using (43), we can then obtain

$$\begin{aligned}
\mathbf{w}_i^1 &= \mathbf{h}_i^1\left(\dot{\mathbf{w}}_1, \ldots, \dot{\mathbf{w}}_{i-1}, \mathbf{z}_1, \ldots, \mathbf{z}_i, \mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \mathbf{y}\right)\\
\mathbf{w}_i^2 &= \mathbf{h}_i^2\left(\dot{\mathbf{w}}_1, \ldots, \dot{\mathbf{w}}_{i-1}, \mathbf{z}_1, \ldots, \mathbf{z}_i, \mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \mathbf{w}_i^1, \mathbf{y}\right)\\
&\vdots\\
\mathbf{w}_i^{p_i} &= \mathbf{h}_i^{p_i}\left(\dot{\mathbf{w}}_1, \ldots, \dot{\mathbf{w}}_{i-1}, \mathbf{z}_1, \ldots, \mathbf{z}_i, \mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \mathbf{w}_i^1, \ldots, \mathbf{w}_i^{p_i}, \mathbf{y}\right) \qquad (44)
\end{aligned}$$

from the elements $\left(W_i^j, E_{W_i}^j\right) \in \mathcal{C}$, $j = 1, 2, \ldots, p_i$. Comparing (41) and (44) with the system in Definition 2, shows that the elements $\left(Z_i^j, E_{Z_i}^j\right) \in \mathcal{C}$, $j = 1, 2, \ldots, s_i$ and $\left(W_i^j, E_{W_i}^j\right) \in \mathcal{C}$, $j = 1, 2, \ldots, p_i$, corresponds to the i:th block of a BLT semi-explicit DAE form. Applying the above arguments for $i = 1, 2, \ldots, s$ then implies that the ordered set $\mathcal{C}$ in (36) can be used to obtain a system in BLT semi-explicit DAE form with s blocks. Thus, $\mathcal{C}$ is computation sequence for X' with $\mathcal{T}$.

It now remains to show that $\mathcal{C}$ is a minimal and irreducible computation sequence for X' with $\mathcal{T}$. We begin with the irreducibility of $\mathcal{C}$. In the beginning of this proof, we showed that all elements of $\mathcal{C}$, given by (36), correspond to SCCs. We have also concluded that due to the assumptions regarding the model in Section 1, all elements $\left(Z_i^j, E_{Z_i}^j\right) \in \mathcal{C}$ are of size one, i.e., trivially irreducible. Now consider an element $\left(W_i^j, E_{W_i}^j\right) \in \mathcal{C}$ and assume that we partition W_i^j as $W_i^j = W_{i1}^j \cup W_{i2}^j$ and $E_{W_i}^j$ as $E_{W_i}^j = E_{W_i1}^j \cup E_{W_i2}^j$ and form the two new elements $\left(W_{i1}^j, E_{W_i1}^j\right)$ and $\left(W_{i2}^j, E_{W_i2}^j\right)$. Due to the fact that $\left(W_i^j, E_{W_i}^j\right)$ corresponds to a SCC, $E_{W_i}^j$ is a dependent equation set with respect to the variables in W_i^j. This implies that when applying $\mathcal{T}$ to the elements $\left(W_{i1}^j, E_{W_i1}^j\right)$ and $\left(W_{i2}^j, E_{W_i2}^j\right)$, we obtain the two equations

$$\begin{aligned}
\mathbf{w}_{i1}^j &= \mathbf{h}_{i1}^1\left(\ldots, \mathbf{w}_{i2}^j, \ldots\right)\\
\mathbf{w}_{i2}^j &= \mathbf{h}_{i2}^1\left(\ldots, \mathbf{w}_{i1}^j, \ldots\right),
\end{aligned}$$

which clearly not has the structure of equations contained in a BLT semi-explicit DAE system, due to the cyclic dependence between the equations. Hence, a system in BLT semi-explicit DAE form can not be obtained when the element $\left(W_i^j, E_{W_i}^j\right) \in \mathcal{C}$ is partitioned, which violates condition 2) in Definition 3. We can then conclude that no elements of $\mathcal{C}$ can be further partitioned and hence $\mathcal{C}$ is an irreducible computation sequence for X' with $\mathcal{T}$.

The minimality of $\mathcal{C}$ for X' with $\mathcal{T}$ trivially follows from the fact that (38) holds. Since as (38) is fulfilled, all elements in $\mathcal{C}$ is needed to compute the variables in X'. This implies that any attempt to form a computation sequence for X' with $\mathcal{T}$ by using a subset of $\mathcal{C}$ will violate condition 1) in Definition 3. This completes the proof. □

References

U. M. Ascher and L. M. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Siam, 1998.

A. S. Asratian, T. M. J. Denley, and R. Häggkvist. *Bipartite Graphs and their Applications*. Cambridge University Press, 1998.

L. Barford, E. Manders, G. Biswas, P. Mosterman, V. Ram, and J. Barnett. Derivative estimation for diagnosis. Technical report, HP Labs Technical Reports, 1999.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, second edition, 2006.

K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Siam, 1989.

R. W. Brockett. *Finite-Dimensional Linear Systems*. Wiley, New York, 1970.

J. P. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *Proceedings of IFAC Control Ind. Syst.*, pages 841–846, Belfort, France, 1997.

F. E. Cellier and H. Elmqvist. Automated formula manipulation supports object-oriented continuous-system modeling. *IEEE Control Systems Magazine*, 13(2):28–38, April 1993.

F. E. Cellier and E. Kofman. *Continuous System Simulation*. Springer, 2006.

V. Cocquempot, R. Izadi-Zamanabadi, M. Staroswiecki, and M. Blanke. Residual generation for the ship benchmark using structural approach. In *Proceedings of the UKACC International Conference on Control '98*, pages 1480–1485, September 1998.

C. De Persis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46:853–865, 2001.

A. L. Dulmage and N. S. Mendelsohn. Coverings of bi-partite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

D. Dustegor, V. Cocquempot, and M. Staroswiecki. Structural analysis for residual generation: Towards implementation. In *Proceedings of the 2004 IEEE Inter. Conf. on Control App.*, pages 1217–1222, 2004.

E. Frisk, M. Krysander, M. Nyberg, and J. Åslund. A toolbox for design of diagnosis systems. In *Proceedings of IFAC Safeprocess'06*, Beijing, China, 2006.

P. Fritzon. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. IEEE Press, 2004.

E. Hairer and G. Wanner. *Solving Ordinary Equations II - Stiff and Differential-Algebraic Problems*. Springer, 2002.

J. Hansen and J. Molin. Design and evaluation of an automatically generated diagnosis system. Master's thesis, Linköpings Universitet, SE-581 83 Linköping, 2006.

R. Izadi-Zamanabadi. Structural analysis approach to fault diagnosis with application to fixed-wing aircraft motion. In *Proceedings of the 2002 American Control Conference*, volume 5, pages 3949–3954, 2002.

G. Katsillis and M. Chantler. Can dependency-based diagnosis cope with simultaneous equations? In *Proceedings of the 8th Inter. Workshop on Princ. of Diagnosis, DX'97*, pages 51–59, Le Mont-Saint-Michel, France, 1997.

H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

G. Kron. *Diakoptics - The Piecewise Solution of Large-scale Systems*. Macdonald, London, 1963.

M. Krysander and E. Frisk. Sensor placement for fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(6):1398–1410, Nov. 2008.

M. Krysander, J. Åslund, and M. Nyberg. An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(1):197–206, 2008.

P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations - Analysis and Numerical Solution*. European Mathematical Society, 2006.

S. Mattson, H. Elmqvist, and M. Otter. Physical system modeling with modelica. *Control Engineering Practice*, 6(4):501–510, 1998.

K. Murota. *System Analysis by Graphs and Matroids*. Springer-Verlag Berlin Heidelberg, 1987.

S. Narasimhan and G. Biswas. Model-based diagnosis of hybrid systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37(3):348–361, 2007.

M. Nyberg. Automatic design of diagnosis systems with application to an automotive engine. *Control Engineering Practice*, 87(8):993–1005, 1999.

M. Nyberg and M. Krysander. Statistical properties and design criterions for AI-based fault isolation. In *Proceedings of the 17th IFAC World Congress*, pages 7356–7362, Seoul, Korea, 2008.

J. M. Ortega and W. C Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM Classics, 2000.

S. Ploix, M. Desinde, and S. Touaf. Automatic design of detection tests in complex dynamic systems. In *Proceedings of 16th IFAC World Congress*, Prague, Czech Republic, 2005.

B. Pulido and C. Alonso-González. Possible conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics, Special Issue on Diagnosis of Complex Systems*, 34(5):2192–2206, 2004.

B. Pulido, C. Alonso, A. Bregón, V. Puig, and T. Escobet. Analyzing the influence of temporal constraints in possible conflicts calculation for model-based diagnosis. In *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07)*, pages 186–193, Nashville, TN, USA, 2007.

B. Pulido, A. Bregón, and C. Alonso. Combining state estimation and simulation in consistency-based diagnosis using possible conflicts. In *Proceedings of the 19th International Workshop on Principles of Diagnosis (DX-08)*, pages 339–346, Blue Mountains, NSW, Australia, 2008.

M. Staroswiecki. *Fault Diagnosis and Fault Tolerant Control*, chapter Structural Analysis for Fault Detection and Isolation and for Fault Tolerant Control. Encyclopedia of Life Support Systems, Eolss Publishers, Oxford, UK, 2002.

M. Staroswiecki and P. Declerck. Analytical redundancy in non-linear interconnected systems by means of structural analysis. In *Proceedings of IFAC AIPAC'89*, pages 51–55, Nancy, France, 1989.

D. V. Steward. On an approach to techniques for the analysis of the structure of large systems of equations. *SIAM Review*, 4(2):321–342, October 1962.

D. V. Steward. Partitioning and tearing systems of equations. *SIAM Journal on Numerical Analysis*, 2(2):345–365, 1965.

C. Svärd and H. Wassén. Development of methods for automatic design of residual generators. Master's thesis, Linköpings Universitet, SE-581 83 Linköping, 2006.

R. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component-supported analytical redundancy. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 36(6):1146–1160, November 2006.

United Nations. Regulation no. 49: Uniform provisions concerning the measures to be taken against the emission of gaseous and particulate pollutants from compressionignition engines for use in vehicles, and the emission of gaseous pollutants from positive-ignition engines fuelled with natural gas or liquefied petroleum gas for use in vehicles, 2008. ECE-R49.

J. Wahlström. Control of EGR and VGT for emission control and pumping work minimization in diesel engines. Technical report, Linköpings Universitet, 2006. LiU-TEK-LIC-2006:52, Thesis No. 1271.

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

Paper B

Realizability Constrained Selection of Residual Generators for Fault Diagnosis with an Automotive Engine Application[☆]

[☆] Submitted to *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2011.

Realizability Constrained Selection of Residual Generators for Fault Diagnosis with an Automotive Engine Application

Carl Svärd, Mattias Nyberg, and Erik Frisk

Vehicular Systems, Department of Electrical Engineering,
Linköping University, SE-581 83 Linköping, Sweden.

Abstract

This paper considers the problem of selecting a set of residual generators, fulfilling requirements regarding fault isolability and minimal cardinality, for inclusion in a model-based FDI-system. Two novel algorithms for solving the selection problem are proposed. The first one provides an exact solution fulfilling both requirements and is suitable for small problems. The second one, which constitutes the main contribution, is suitable for large problems and provides an approximate solution by means of a greedy heuristic by relaxing the minimal cardinality requirement. The foundation for the algorithms is a novel formulation of the selection problem which enables an efficient reduction of the search-space by taking the realizability properties of the model, with respect to the considered residual generation method, into account. Both algorithms are general in the sense that they are aimed at supporting any computerized residual generation method. In a case study the greedy selection algorithm is successfully applied to the complex problem of finding a suitable set of residual generators for detection and isolation of faults in an automotive engine system. In this study a prior known sequential residual generation method is considered.

1 Introduction

Model-based Fault Detection and Isolation (FDI) systems typically contains the three sub-systems: residual generation, residual evaluation, and fault isolation, see, e.g., Blanke et al. (2006). In this work, as in for example Nyberg (1999); Krysander (2006); Nyberg and Krysander (2008); Svärd and Nyberg (2010), design of the residual generation sub-system is considered to be a two-step approach. In the first step, a large set of candidate residual generators are found. In general, it may be possible to find thousands of candidate residual generators for large models and regarding implementation aspects such as complexity and computational load it is infeasible, or even impossible, to use all these in the FDI-system. In addition, it is often possible to meet stated requirements with a, possibly small, subset of all residual generators. Therefore, in the second step, the set of candidate residual generators most suitable to be included in the FDI-system are selected. The topic of this paper is the selection problem emerging in the second step.

The selection problem is formulated by considering two different requirements on the final set of residual generators. Firstly, it is required that the set of residual generators fulfills an isolability requirement stating which faults that should be isolated from each other. Motivated by the implementation aspects mentioned above, a set of residual generators of low cardinality is preferred before a set of high cardinality, given that the two sets have equal isolability properties. Therefore, secondly, it is required that the set of residual generators is of minimal cardinality.

Two novel algorithms for solving the selection problem are proposed in this paper. The first one provides an exact solution fulfilling both the isolability and the minimal cardinality requirements and is suitable for small problems. The second one, which is the main contribution, relaxes the minimal cardinality requirement and provides an approximate solution by means of a greedy heuristic. This algorithm is suitable for large, real-world, problems for which the approach used in the first algorithm is intractable. Both algorithms are general in the sense that they are aimed at supporting any computerized residual generation method.

In general, all the candidate residual generators found in the first step of the design process are not realizable, i.e., it is not possible to create residual generators from all found candidate residual generators. Typically, evaluation of realizability is a computational demanding task. Therefore, in those cases where the number of found candidate residual generators is large, it may not be feasible to first evaluate the realizability of all found candidate residual generators and then make the selection. To handle this, the proposed algorithms exploits a novel formulation of the selection problem which takes the realizability aspect into account. This, in addition, enables an efficient reduction of the search-space which typically is quite large for practical problems. In this formulation, which in fact is an optimization problem, isolability and realizability properties are stated in terms of attributes of subsets of the model equations.

In Section 2, a motivating industrial application example is presented. Section 3 presents preliminaries regarding realizability and fault isolability, given a residual generation method. The residual generator selection problem is formalized in Section 4. The first selection algorithm is presented and discussed in Section 5. The second, greedy, algorithm is presented and justified in Section 6. Section 7 briefly describes the residual

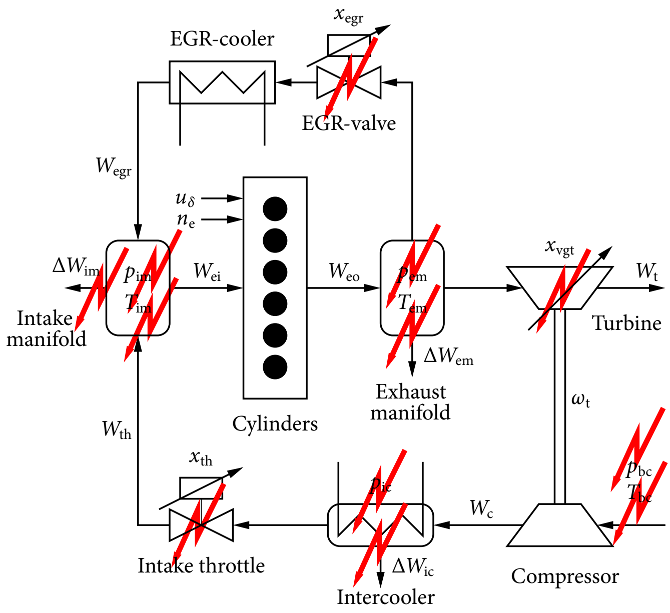


Figure 1: Overview of the automotive engine System. Considered faults marked with red arrows.

generation method (Svärd and Nyberg, 2010) which is used in the application example. In Section 8 the greedy selection algorithm is used to solve the industrial application problem described in Section 2. The paper is concluded in Section 9.

2 Motivating Application Example

As a motivating industrial application example, consider the problem of selecting a set of suitable residual generators for detecting and isolating faults in an automotive engine system. The studied engine is a 13-L six-cylinder Scania truck diesel engine equipped with Exhaust Gas Recirculation (EGR), Variable Geometry Turbine (VGT), and intake throttle.

There are in total 12 faults that should be detected and isolated from each other in this system. An overview of the system with the considered faults, is shown in Figure 1. More details regarding the system, and the faults, are given in Section 8.

For the model of this system, and for the specific residual generation method developed in Svärd and Nyberg (2010), which is briefly described in Section 7, it is possible to find in total *14,242* candidate residual generators. Indeed, as argued in Section 1, it is not possible to include all these residual generators in the FDI-system.

In order to isolate a certain fault from an other, it is necessary to find a residual generator sensitive to the fault but not to the other. Intuitively, a set of approximately 12

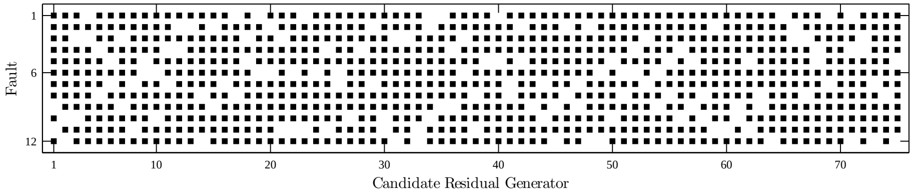


Figure 2: Fault sensitivity for a small subset of the 14,242 candidate residual generators found for the automotive engine system. A square in position (i,j) denotes that the residual generator corresponding to column j is sensitive to the fault corresponding to row i.

residual generators would be sufficient in order to isolate the 12 considered faults from each other. Thus, a set of 12 residual generators, capable of isolating the 12 faults, should be selected from the set of 14,242 candidate residual generators which means that the search-space is quite large.

The fault sensitivity for a small subset of the found candidate residual generators, with respect to the 12 considered faults, are shown in Figure 2. According to the figure, most residual generators are sensitive to most faults and it is therefore not straightforward to perform the selection. In addition, as said in Section 1, the sought set of residual generators should be realizable and preferably of minimal cardinality. Due to the vast number of candidate residual generators it is not possible to perform a complete search in order to find the set of residual generators, which makes the selection problem non-trivial. In Section 8 this selection problem will be reconsidered and solved.

3 Preliminaries

The purpose of this section is to formally introduce the notions of realizability and isolability, given a residual generation method, and ultimately derive necessary and sufficient conditions for fault isolability in terms of properties of model equation subsets.

Consider a *model*, $M = (E, X, Y, F)$, containing an equation set E relating the unknown variables X, known variables Y, and fault variables F. Without loss of generality, the following is assumed regarding the model.

Assumption 1. *Each fault $f \in F$ is contained in one, and only one, of the equations in the model M.*

Note that if a fault $f \in F$ is contained in more than one equation, the fault f can be replaced with a new variable x_f in these equations, and the equation $x_f = f$ added to the equation set E. This added equation will then be the only equation where f occurs.

Given a model, a residual generator is formally defined as follows.

Definition 1 (Residual Generator). *Let $M = (E, X, Y, F)$ be a model. A system R with input Y and output r is a* residual generator *for M, and r is a* residual, *if $f = 0$ implies $r = 0$ for all $f \in F$.*

An important property of a residual generator is whether or not it responds to a certain fault.

Definition 2 (Fault Sensitivity). *Let R be a residual generator for the model M. Then R is* sensitive *to fault $f \in F$ if $f \neq 0$ implies $r \neq 0$.*

Note that in practice, residuals typically deviate from zero even in the case when all faults are zero due to for example unknown initial conditions, changes in operating conditions, and uncertainties such as modeling errors and noise. Therefore, residuals are often thresholded as a part of the residual evaluation mentioned in Section 1, where the aim is to detect changes in the residual behavior caused by faults.

The notions of residual generator and fault sensitivity are possible to make more precise and formal, see for example Blanke et al. (2006); Patton et al. (2000); Chen and Patton (1999), and references therein. This is however not necessary in the context of this work for which the above definitions are sufficient.

3.1 Realizability

The method used for design of residual generators plays a central role in this work. A residual generation method is formally defined as follows.

Definition 3 (Residual Generation Method $\mathcal{M}$). *Let $M = (E, X, Y, F)$ be a model. A* residual generation method, *$\mathcal{M}$, is a procedure, denoted $\mathcal{M}(\cdot)$, taking as input a set of equations $S \subseteq E$ and giving as output a residual generator R for M, or an empty set $\varnothing$.*

Given a residual generation method and an equation set, an important issue is whether the output from the method is non-empty, or not. That is, if a residual generator can be created with the method given the equation set as input. This property of an equation set, with respect to a method, is formalized below.

Definition 4 (Realizability with method $\mathcal{M}$). *Let S be an equation set and $\mathcal{M}$ a residual generation method. Then S is* realizable *with $\mathcal{M}$ if $\mathcal{M}(S) \neq \varnothing$.*

For an example, consider a model containing the following set of differential and algebraic equations

$$\begin{aligned} e_1 :&\quad \dot{x}_1 = -x_1 + u + f_1 \\ e_2 :&\quad y_1 = x_1 + f_2 \\ e_3 :&\quad y_2 = x_1 + f_3, \end{aligned} \tag{1}$$

where x_1 is an unknown variable, $\{u, y_1, y_2\}$ known variables, and $\{f_1, f_2, f_3\}$ fault variables. Let $\mathcal{M}'$ be a residual generation method capable of handling linear, static, equation sets. It can then be concluded that the equation set $\{e_2, e_3\}$ is realizable with $\mathcal{M}'$, but not for instance the equation set $\{e_1, e_2\}$ since e_1 is a differential equation.

Let e_f denote the equation in an equation set containing fault f. From now on, the following is assumed regarding a residual generation method.

Assumption 2. *Let S be an equation set and $\mathcal{M}$ a residual generation method. Further, let S be realizable with $\mathcal{M}$ and $R = \mathcal{M}(S)$ the corresponding residual generator. Then, R is sensitive to fault f if and only of $e_f \in S$.*

The important implication of Assumption 2, in the context of this work, is that a residual generation method preserves structural fault information, in the sense that it does not discard, nor add, equations containing faults, during the realization process.

For an example, consider again the model (1) and assume that the equation set $\{e_1, e_2, e_3\}$ is realizable with a method $\mathcal{M}$. If $\mathcal{M}$ fulfills Assumption 2, it is guaranteed that the residual generator obtained from $\mathcal{M}(\{e_1, e_2, e_3\})$ is sensitive to the faults f_1, f_2, and f_3. Thus, the output from $\mathcal{M}$ can neither be the residual generator $r = y_1 - y_2$ since this residual generator not is sensitive to f_1, nor the trivial residual generator $r = 0$.

3.2 Fault Isolability

In this section fault isolability is formally defined from two different perspectives. First, fault isolability is defined as a property of a given set of residual generators. Second, fault isolability is defined as a property of a model given a method for residual generation. The main motivation for introducing both definitions is to prove soundness and completeness of the selection algorithms in Sections 5 and 6. More specific, that the algorithms find a set of residual generators fulfilling the stated isolability requirement if, and only if, the corresponding faults are isolable in the model with the considered method for residual generation.

Given a set of residual generators, fault isolability is defined as follows.

Definition 5 (Fault Isolability with residual generators $\mathcal{R}$). *Let $M = (E, X, Y, F)$ be a model and $\mathcal{R}$ a given set of residual generators for M. A fault $f_i \in F$ is* isolable *from fault $f_j \in F$ with $\mathcal{R}$ if there exists a residual generator $R \in \mathcal{R}$ that is sensitive to f_i but not to f_j.*

Note that Definition 5 is not dependent on the residual generation method. Next, fault isolability is defined as a property of a model, given a residual generation method.

Definition 6 (Fault Isolability with method $\mathcal{M}$). *Let $M = (E, X, Y, F)$ be a model and $\mathcal{M}$ a residual generation method. A fault $f_i \in F$ is* isolable *from fault $f_j \in F$ in M with $\mathcal{M}$ if a residual generator R for M can be created with $\mathcal{M}$ such that R is sensitive to f_i but not to f_j.*

Note that if $S \subseteq E$ and fault $f_i \in F$ is isolable from fault $f_j \in F$ with the residual generator $R = \mathcal{M}(S)$ then, by Definition 6, f_i is isolable from f_j in the model M with the method $\mathcal{M}$. The converse is also true. For future reference, this trivial result is stated below.

Proposition 1. *Let $M = (E, X, Y, F)$ be a model and $\mathcal{M}$ a residual generation method. Then, fault $f_i \in F$ is isolable from fault $f_j \in F$ in M with $\mathcal{M}$ if and only if there exists $S \subseteq E$ such that f_i is isolable from f_j with $R = \mathcal{M}(S)$.*

By exploiting the notion of realizability and Assumption 2, necessary and sufficient conditions for fault isolability, given a model and a residual generation method, in terms of properties of subsets of the model equations can be established.

Proposition 2. *Let $M = (E, X, Y, F)$ be a model and $\mathcal{M}$ a residual generation method. Then, for each $S \subseteq E$ it holds that fault $f_i \in F$ is isolable from fault $f_j \in F$ with $R = \mathcal{M}(S)$ if and only if S is realizable with $\mathcal{M}$, $e_{f_i} \in S$, and $e_{f_j} \notin S$.*

Proof. Assume first that f_i is isolable from f_j with $R = \mathcal{M}(S)$. By assumption, R is a residual generator and therefore $\mathcal{M}(S) \neq \varnothing$ and it follows that S is realizable with $\mathcal{M}$ from Definition 4. Further, by Definition 5, R is sensitive to f_i but not to f_j. Assumption 2 then implies that $e_{f_i} \in S$ and $e_{f_j} \in S$, and the first part of the proof is complete. For the converse, assume that S that is realizable with $\mathcal{M}$, i.e., $\mathcal{M}(S) \neq \varnothing$, $e_{f_i} \in S$, and $e_{f_j} \notin S$. Since $S \in E$ and $\mathcal{M}(S) \neq \varnothing$, it follows from Definition 3, that $R = \mathcal{M}(S)$ is a residual generator for M. Assumption 2 then states that R is sensitive to f_i but not to f_j. Definition 5 completes the proof. □

Consider again the model in (1) and the linear, static, residual generation method $\mathcal{M}'$ with which the equation set $\{e_2, e_3\}$ is realizable. Due to this fact and since $e_{f_2} = e_2 \in \{e_2, e_3\}$, $e_{f_3} = e_3 \in \{e_2, e_3\}$, and $e_{f_1} = e_1 \notin \{e_2, e_3\}$, it can be deduced from Proposition 2 that faults f_2 and f_3 are both isolable from fault f_1 with the residual generator $R' = \mathcal{M}'(\{e_2, e_3\})$.

Note that even though additive faults were considered in this example above, the framework in this paper is general and independent on the fault model, i.e., also multiplicative faults are allowed.

4 The Residual Generator Selection Problem

In this section, the residual generator selection problem is formalized and stated as an optimization problem: fulfill an isolability requirement while minimizing the number of residual generators. This formulation exploits the notion of realizability introduced in the previous section and enables an efficient reduction of the search-space.

As input to the residual generator selection procedure the following are assumed to be given: a model $M = (E, X, Y, F)$, a method for residual generation $\mathcal{M}$, and an isolability requirement $\mathcal{F}$. The output from the selection procedure is a set of residual generators, $\mathcal{R}$. As said in Section 1, two different requirements on $\mathcal{R}$ are considered:

1. $\mathcal{R}$ should fulfill the isolability requirement $\mathcal{F}$, and

2. $\mathcal{R}$ should be of minimal cardinality.

4.1 The Isolability Requirement

The *isolability requirement*, $\mathcal{F}$, is defined as a set of ordered fault pairs $(f_i, f_j) \in F \times F$, where the interpretation of (f_i, f_j) is that f_i should be isolable from f_j with the set of residual generators $\mathcal{R}$. Consequently, $\mathcal{F}$ is *fulfilled* with $\mathcal{R}$ if for each $(f_i, f_j) \in \mathcal{F}$ it holds that f_i is isolable from f_j with $\mathcal{R}$.

From Proposition 2 it can be deduced that to fulfill the isolability requirement it is necessary, and sufficient, to find for each fault pair $(f_i, f_j) \in \mathcal{F}$ an equation set $S_{f_i f_j} \subseteq E$

such that $S_{f_i f_j}$ is realizable with $\mathcal{M}$, and for which $e_{f_i} \in S_{f_i f_j}$ and $e_{f_j} \notin S_{f_i f_j}$. Given the equation subsets $S_{f_i f_j}$, a set of residual generators fulfilling $\mathcal{F}$ can be constructed as

$$\mathcal{R} = \left\{ \mathcal{M}\left(S_{f_i f_j}\right) : \forall \left(f_i, f_j\right) \in \mathcal{F} \right\}. \tag{2}$$

4.2 Candidate Equation Set

If E is a small set, it may be tractable to evaluate all subsets of E in the search for the sets $S_{f_i f_j}$ in (2). In the general case, however, it is not. In order to reduce the search-space, all subsets of E that not by necessity are realizable are discarded. To this end, the notions of *necessary realizability criterion* and *candidate equation set* are introduced.

Definition 7 (Necessary Realizability Criterion for method $\mathcal{M}$). *Let S be an equation set and $\mathcal{M}$ a residual generation method. A constraint on S is a* necessary realizability criterion *for $\mathcal{M}$ if the constraint is satisfied when S is realizable with $\mathcal{M}$.*

Definition 8 (Candidate Equation Set for method $\mathcal{M}$). *Let S be an equation set and $\mathcal{M}$ a residual generation method for which a necessary realizability criterion is defined. Then S is a* candidate equation set *for $\mathcal{M}$ if S fulfills the necessary realizability criterion for $\mathcal{M}$.*

Regarding the choice of necessary realizability criterion for a given residual generation method, it is desirable that it fulfills at least two requirements. First of all, in order to meaningful, the necessary realizability criterion should reduce the search-space, in terms of number of discarded non-realizable subsets of the model equations, to a high extent. Secondly, in order to be of practical use, it should be possible to extract all candidate equation sets for a method, given a model, in an efficient way.

As an example, a candidate equation set for several observer-based residual generation methods is an equation set in, or that trivially can be cast in, state-space form, see, e.g., Blanke et al. (2006); Chen and Patton (1999) and references therein. An additional example is given by the class of methods referred to as sequential residual generation, see, e.g., Staroswiecki and Declerck (1989); Cassar and Staroswiecki (1997); Ploix et al. (2005); Blanke et al. (2006); Svärd and Nyberg (2010), for which Minimal Structurally Over-determined (MSO) sets of equations Krysander et al. (2008); Gelso et al. (2008); Travé-Massuyès et al. (2006), constitute candidate equation sets.

4.3 Formalization of the Selection Problem

Consider now the isolability requirement $\mathcal{F}$ and let $\mathcal{S}_{\mathcal{M}} \subseteq 2^E$ be the set of all candidate equation sets for the residual generation method $\mathcal{M}$.

Define the *isolability class*, $I_{f_i f_j}$, of $\mathcal{S}_{\mathcal{M}}$ for the fault pair $\left(f_i, f_j\right) \in \mathcal{F}$ as the collection of all candidate equation sets in $\mathcal{S}_{\mathcal{M}}$ containing fault f_i but not fault f_j, that is,

$$I_{f_i f_j} = \left\{ S \in \mathcal{S}_{\mathcal{M}} : e_{f_i} \in S \wedge e_{f_j} \notin S \right\}. \tag{3}$$

Let the set

$$\mathcal{I} = \left\{ I_{f_i f_j} : \forall \left(f_i, f_j\right) \in \mathcal{F} \right\}, \tag{4}$$

contain the isolability classes of $\mathcal{S}_{\mathcal{M}}$ for all fault pairs in $\mathcal{F}$.

The next result formulates the problem of fulfilling the isolability requirement in terms of properties of the candidate equation sets.

Lemma 1. *Let $M = (E, X, Z, F)$ be a model, $\mathcal{M}$ a residual generation method, and $\mathcal{F}$ an isolability requirement. Also, let $\mathcal{S}_{\mathcal{M}}$ be the set of all candidate equation sets for $\mathcal{M}$ and $\mathcal{I}$ the set of all isolability classes of $\mathcal{S}_{\mathcal{M}}$ for $\mathcal{F}$, defined according to (3) and (4). Then, for each $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$ where all $S \in \mathcal{S}$ is realizable with $\mathcal{M}$ it holds that $\mathcal{F}$ is fulfilled with*

$$\mathcal{R} = \{\mathcal{M}(S) : \forall S \in \mathcal{S}\}, \tag{5}$$

if and only if

$$\forall I \in \mathcal{I}, \quad \mathcal{S} \cap I \neq \emptyset. \tag{6}$$

Proof. Assume first that $\mathcal{F}$ is fulfilled with $\mathcal{R}$ defined according to (5). First note that this implies that for each $(f_i, f_j) \in \mathcal{F}$ there exists a residual generator $R \in \mathcal{R}$ such that f_i is isolable from f_j with R. This, Proposition 2, and (5), imply that for each $(f_i, f_j) \in \mathcal{F}$ there exists a $S \in \mathcal{S}$ such that $R = \mathcal{M}(S) \in \mathcal{R}$, $e_{f_i} \in S$, and $e_{f_j} \notin S$. This implies, since $S \in \mathcal{S}$ and $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$, that $S \cap I_{f_i f_j} \neq \emptyset$ where $I_{f_i f_j}$ is defined according to (3). Hence, for each $(f_i, f_j) \in \mathcal{F}$ there exists $S \in \mathcal{S}$ such that $S \cap I_{f_i f_j} \neq \emptyset$. Since (4) holds, this implies that (6) is satisfied and the first part of the proof is complete. For the converse, assume that (6) is satisfied. This, (3) and (4) implies that for each $(f_i, f_j) \in \mathcal{F}$ there exists $S \in \mathcal{S}$ such that $e_{f_i} \in S$ and $e_{f_j} \notin S$. This and the fact that all $S \in \mathcal{S}$ are realizable with $\mathcal{M}$, implies via Proposition 2 that for each $(f_i, f_j) \in \mathcal{F}$ there exists $S \in \mathcal{S}$ such that f_i is isolable from f_j with $R = \mathcal{M}(S)$. Thus, if $\mathcal{R} = \{\mathcal{M}(S) : \forall S \in \mathcal{S}\}$ there exists $R \in \mathcal{R}$ such that f_i is isolable from f_j with R for each $(f_i, f_j) \in \mathcal{F}$ and the proof is complete. □

For the set of residual generators $\mathcal{R}$ to fulfill also the stated minimal cardinality requirement, the cardinality of the set $\mathcal{S}$ in Lemma 1 should be minimized. Thus, the residual generator selection problem can be stated as the problem of finding the smallest set within $\mathcal{S}_{\mathcal{M}}$ which satisfies (6). To conclude, the selection problem is stated as the minimization problem

$$\min_{\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}} \quad |\mathcal{S}| \tag{7a}$$

$$\text{s.t.} \quad \forall S \in \mathcal{S}, \quad \mathcal{M}(S) \neq \emptyset \tag{7b}$$

$$\forall I \in \mathcal{I}, \quad \mathcal{S} \cap I \neq \emptyset, \tag{7c}$$

where $|\cdot|$ returns the cardinality of a set.

5 Minimal Hitting Set Based Selection

A *hitting set* is a set that has a non-empty intersection with every set in a collection of sets. In fact, the isolability requirement, given by (7c), on the set of candidate equation sets $\mathcal{S}$ implies that $\mathcal{S}$ should be a hitting set for the collection of sets $\mathcal{I}$. Further, to also fulfill the minimal cardinality requirement (7a), $\mathcal{S}$ should be a hitting set for $\mathcal{I}$ of minimal cardinality, i.e., a so called *minimal cardinality hitting set*. By necessity, a minimal cardinality hitting set is a *minimal hitting set*, i.e., a hitting set of which no proper subset is a hitting set.

This fact suggests the following naive, but nevertheless simple, approach for solving the selection problem (7). First find the collection of all minimal hitting sets for $\mathcal{I}$, denoted $\mathcal{H}$, and then find the smallest set $H \in \mathcal{H}$, where all candidate equation sets $S \in H$ are realizable.

5.1 MHS-Based Selection Algorithm

The naive selection approach outlined above is the basis for the procedure selectResGenMHS presented in Algorithm 1, taking as input a model M, a residual generation method $\mathcal{M}$, and an isolability requirement $\mathcal{F}$. The output is a set of residual generators $\mathcal{R}$.

Algorithm 1 MHS-Based Selection of Residual Generators

Input: Model M, residual generation method $\mathcal{M}$, isolability requirement $\mathcal{F}$
Output: Set of residual generators $\mathcal{R}$

```
procedure selectResGenMHS($M, \mathcal{M}, \mathcal{F}$)
    $\mathcal{S} \leftarrow \varnothing$
    $\mathcal{R} \leftarrow \varnothing$
    $\mathcal{S}_{\mathcal{M}} \leftarrow$ findCES($\mathcal{M}, M$)
    $\mathcal{I} \leftarrow$ isolClasses($\mathcal{S}_{\mathcal{M}}, \mathcal{F}$)
    $\mathcal{H} \leftarrow$ findMHS($\mathcal{I}$)
    while $\mathcal{H} \neq \varnothing$ do
        $H^* \leftarrow \arg\min_{H \in \mathcal{H}} |H|$
        for all $S \in H^*$ do
            $R \leftarrow \mathcal{M}(S)$
            if $R \neq \varnothing$ then
                $\mathcal{S} \leftarrow \mathcal{S} \cup \{S\}$, $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$
            else
                $\mathcal{H} \leftarrow \mathcal{H} \setminus \{H^*\}$
                $\mathcal{S} \leftarrow \varnothing$, $\mathcal{R} \leftarrow \varnothing$
                break
            end if
        end for
        if $\mathcal{R} \neq \varnothing$ then
            break
        end if
    end while
    return $\mathcal{R}$
end procedure
```

The others procedures used in Algorithm 1 are listed below:

- findCES finds all candidate equation sets for the method $\mathcal{M}$ given a model M and a necessary realizability criterion for $\mathcal{M}$.
- isolClasses returns the set of all isolability classes of a set of candidate equation sets $\mathcal{S}_{\mathcal{M}}$ for the isolability requirement $\mathcal{F}$ according to (3) and (4).
- findMHS finds all minimal hitting sets for the collection of sets $\mathcal{I}$ given as input.

Note that in an efficient implementation of Algorithm 1, it is preferable to keep book of those candidate equation sets that have been realized, successfully or not, in previous iterations in order to avoid unnecessary calls to the procedure $\mathcal{M}(\cdot)$, which may be expensive.

5.2 Properties of the MHS-Based Selection Algorithm

Algorithm 1 is formally justified by Theorem 1 below. The theorem states that if, and only if, the given isolability requirement can be fulfilled with any set of residual generators created with the given method, then Algorithm 1 finds a set of residual generators fulfilling the requirement. In addition, it is guaranteed that this set of residual generators is of minimal cardinality, i.e., there is no residual generator set of lower cardinality that fulfills the isolability requirement.

Theorem 1. *Let $M = (E, X, Y, F)$ be a model, $\mathcal{M}$ a residual generation method, and $\mathcal{F}$ an isolability requirement. Further, let M, $\mathcal{M}$, and $\mathcal{F}$ be input to Algorithm 1 and $\mathcal{R}$ the output. Then, $\mathcal{F}$ is fulfilled in M with $\mathcal{M}$ if and only if $\mathcal{F}$ is fulfilled with $\mathcal{R}$. Further, if $\mathcal{F}$ is fulfilled with $\mathcal{R}$ then $\mathcal{R}$ is of minimal cardinality.*

Proof. Consider first the claim concerning the isolability requirement $\mathcal{F}$ and assume that $\mathcal{R} \neq \emptyset$. Due to rows 10-17 in Algorithm 1, and the fact that $\mathcal{R} \neq \emptyset$, it holds that $\mathcal{R}$ equals (5) and consequently there is a $\mathcal{S} \in \mathcal{H}$ where all $S \in \mathcal{S}$ is realizable with $\mathcal{M}$. From rows 4-6 and 7 and the definition of $\mathcal{I}$, see (3) and (4), it can also be deduced that $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$. Hence, $\mathcal{S}$ fulfills the prerequisites of Lemma 1. Further, due to rows 4-6, it can be concluded that $\mathcal{S}$ is a (minimal) hitting set for $\mathcal{I}$ and thus $\mathcal{S}$ fulfills (6). From Lemma 1 it then follows that this property of $\mathcal{S}$ is equivalent to that $\mathcal{F}$ is fulfilled with $\mathcal{R}$ which, according to Proposition 1, is equivalent to that $\mathcal{F}$ is fulfilled in M with $\mathcal{M}$.

If instead $\mathcal{R} = \emptyset$, rows 4-7 and 10-17 implies that there is no minimal hitting set in $\mathcal{H}$ where all candidate equation sets are realizable with $\mathcal{M}$. Hence, there is no $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$, where all $S \in \mathcal{S}$ are realizable with $\mathcal{M}$, that fulfills (6). This is, due to Lemma 1, equivalent to that $\mathcal{F}$ not is fulfilled with $\mathcal{R}$ which is equivalent to that $\mathcal{F}$ not is fulfilled in M with $\mathcal{M}$, due to Proposition 1. This completes the part of the proof considering the isolability requirement.

Regarding the cardinality of $\mathcal{R}$, or equivalently $\mathcal{S}$, it is first noted that a minimal cardinality hitting set also is a minimal hitting set, that is, a hitting set of which no proper subset is a hitting set. Thus, a minimal cardinality hitting set is by necessity found within the collection $\mathcal{H}$ of all minimal hitting sets computed in row 6. Since the search for a realizable minimal hitting set in $\mathcal{H}$, rows 7-22, is exhaustive and performed by considering the sets in $\mathcal{H}$ in increasing order with respect to cardinality, row 8, it is guaranteed that the first found, and then returned, realizable minimal hitting set is of minimal cardinality. □

The minimal hitting set problem, or the equivalent minimal set covering problem (Ausiello et al., 1980), is unfortunately known to be NP-complete, see, e.g., Karp (1972); Aho et al. (1974); Garey and Johnson (1979). Thus, for large problems, that is, cases when the number of candidate equation sets $|\mathcal{S}_{\mathcal{M}}|$, as well as the number of isolability classes $|\mathcal{I}|$, is large, it may be impossible, or at least intractable, to obtain the collection of all minimal hitting sets for $\mathcal{I}$. Two possible improvements of Algorithm 1, which may overcome this complexity issue, are discussed below.

Using an Approximate MHS Algorithm

There are several algorithms that give approximate solutions, typically in the form of a subset of all minimal hitting sets, to the NP-complete minimal hitting set problem, see for example Abreu and van Gemund (2009) and references therein. A complicating issue is however that for large and complex models, typically, only a fraction of the candidate equation sets are realizable. Indeed, this situation applies to the automotive engine system considered in Section 8. Typical causes of non-realizability are non-invertible functions in the model, see for example Svärd and Nyberg (2010), but also numerical issues or instability. For Algorithm 1, this implies that a vast amount of the found minimal hitting sets, possibly all, would be discarded since only a fraction of the found minimal hitting sets contain realizable candidate equation sets. To maximize the possibilities of finding a minimal hitting set in which all candidate equation sets are realizable, it is important to start with as many minimal hitting sets as possible. The reduced number of minimal hitting sets found by an approximate algorithm may therefore not be large enough.

Reducing the Problem Size

Another alternative approach is to find the realizable subset of all candidate equation sets, $\mathcal{S}'_{\mathcal{M}} = \{S \in \mathcal{S}_{\mathcal{M}} : \mathcal{M}(S) \neq \varnothing\}$, calculate $\mathcal{I}'$ according to (3) and (4) using $\mathcal{S}'_{\mathcal{M}}$ instead of $\mathcal{S}_{\mathcal{M}}$, and then apply a minimal hitting set algorithm to $\mathcal{I}'$ to obtain $\mathcal{S}$. In general, it holds that $|\mathcal{S}'_{\mathcal{M}}| < |\mathcal{S}_{\mathcal{M}}|$ and $|\mathcal{I}'| < \mathcal{I}$, and therefore it is more likely that the set of all minimal hitting sets can be computed for $\mathcal{I}'$ than for $\mathcal{I}$. The set $\mathcal{S}'_{\mathcal{M}}$ can be computed by applying $\mathcal{M}(\cdot)$ to each $S \in \mathcal{S}_{\mathcal{M}}$. However, realization of an equation set may be a computational demanding task, see Section 8.2 for an example. It is therefore desirable to keep the number of realizations, or realization attempts, at a minimum. Consequently, this approach may not be preferable if $\mathcal{S}_{\mathcal{M}}$ is a large set.

It should however be noted that for small problems, where all minimal hitting set can be found, Algorithm 1 works satisfactory and in those cases it provides an exact, and yet straightforward and simple, solution to the selection problem.

6 Greedy Selection

Taking into account the complexity issues associated with finding all minimal hitting sets, and the urge of keeping the number of realizations at a minimum, a more appealing approach is instead to build the set of candidate equation sets $\mathcal{S}$ iteratively, and only realize those candidate equation sets that are likely to be part of $\mathcal{S}$. To employ this iterative approach, a heuristic is needed for identifying and selecting a candidate equation set in each iteration.

6.1 Greedy Heuristic

For the general minimal hitting set problem, or the equivalent set covering problem, a greedy heuristic (Black, 2005) has shown Johnsson (1974); Lovász (1975); Chvatal (1979) to provide an approximate solution at a reasonable cost. Using a greedy approach, the candidate equation set with the largest *utility*, is selected in each iteration of the algorithm and added to the solution if it is realizable. The iterations continue until the solution is *complete*. In order to use this approach, a utility function that evaluates the usefulness of a given candidate equation set must be defined, and the properties of a complete solution to the selection problem must be stated to know when to stop the iterations.

Given the set of isolability classes $\mathcal{I}$ of the candidate equation sets $\mathcal{S}_{\mathcal{M}}$ for the isolability requirement $\mathcal{F}$, define the *isolability class coverage* of a set $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$ as

$$\sigma_{\mathcal{I}}(\mathcal{S}) = \{I \in \mathcal{I} : \exists S \in \mathcal{S}, S \in I\}. \tag{8}$$

Basically, $\sigma_{\mathcal{I}}(\mathcal{S})$ states which of the isolability classes in $\mathcal{I}$ that are covered by the candidate equation sets in $\mathcal{S}$.

Complete Solution

A complete solution to the selection problem is characterized as a set of candidate equation sets $\mathcal{S}$ that fulfills (7b) and (7c). The hitting set requirement (7c) can with the isolability class coverage notion be formulated as $\sigma_{\mathcal{I}}(\mathcal{S}) = \mathcal{I}$.

Utility Function

The aim is fulfill the isolability requirement, formalized by (7b) and (7c), with as few candidate equation sets as possible (7a). In line with this, the following utility function will be used to evaluate a specific candidate equation set,

$$\mu_{\mathcal{I}}(S) = |\sigma_{\mathcal{I}}(\{S\})|, \tag{9}$$

reflecting how many of the isolability classes in $\mathcal{I}$ that are covered by the candidate equation set $S \in \mathcal{S}_{\mathcal{M}}$. According to the greedy approach the candidate equation set that maximizes $\mu_{\mathcal{I}}(S)$, i.e., covers most isolability classes, should be selected in each iteration.

6.2 Greedy Selection Algorithm

The procedure selectResGenGreedy for greedy selection of residual generators is presented in Algorithm 2. Input to the algorithm is a model M, a residual generation method $\mathcal{M}$, and an isolability requirement $\mathcal{F}$. The output is a set of residual generators $\mathcal{R}$.

Algorithm 2 Greedy Selection of Residual Generators

```
**Input:** Model $M$, residual generation method $\mathcal{M}$, isolability requirement $\mathcal{F}$
**Output:** Set of residual generators $\mathcal{R}$
procedure selectResGenGreedy($\mathcal{M}, E, \mathcal{F}$)
    $\mathcal{S} \leftarrow \emptyset$
    $\mathcal{R} \leftarrow \emptyset$
    $\mathcal{S}_{\mathcal{M}} \leftarrow$ findCES($\mathcal{M}, E$)
    $\mathcal{I} \leftarrow$ isolClasses($\mathcal{S}_{\mathcal{M}}, \mathcal{F}$)
    while $\mathcal{I} \neq \emptyset$ do
        if $\mathcal{S}_{\mathcal{M}} \neq \emptyset$ then
            $\mathrm{H} \leftarrow \{S' \in \mathcal{S}_{\mathcal{M}} : S' = \arg\max_{S \in \mathcal{S}_{\mathcal{M}}} \mu_{\mathcal{I}}(S)\}$
            $S^* \leftarrow$ pickCES(H)
            $R \leftarrow \mathcal{M}(S^*)$
            if $R \neq \emptyset$ then
                $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$
                $\mathcal{S} \leftarrow \mathcal{S} \cup \{S^*\}$
                $\mathcal{I} \leftarrow \mathcal{I} \setminus \sigma_{\mathcal{I}}(\{S^*\})$
            end if
            $\mathcal{S}_{\mathcal{M}} \leftarrow \mathcal{S}_{\mathcal{M}} \setminus \{S^*\}$
        else
            return $\mathcal{R}$
        end if
    end while
    return $\mathcal{R}$
end procedure
```

The procedures findCES and isolClasses are the same as in Algorithm 1 and described in Section 5.2. The procedure pickCES, taking a set H containing candidate equation sets as input, returns one of the equation sets in H. This function enables usage of an additional, user-provided, heuristic for selecting one single candidate equation set among candidate equation sets of equal utility by analyzing both structural and analytical properties of equation sets. For instance, pickCES can be used to pick the candidate equation set of lowest cardinality, i.e., containing fewest equations or to pick a candidate equation set not containing a troublesome non-linearity.

Note that the complexity of Algorithm 2 is linear in the number of elements of $\mathcal{S}_{\mathcal{M}}$, in comparison with the NP-completeness of Algorithm 1 originating from the search for all minimal hitting sets. For a further complexity analysis of Algorithm 2, the complexity of the procedure findCES is of most interest. The complexity of findCES is however dependent of the actual method used for residual generation. For the method employed in Section 8, the procedure corresponding to findCES has nice complexity properties (Krysander et al., 2008).

6.3 Properties of the Greedy Selection Algorithm

This section explores the properties of Algorithm 2 in terms of providing a solution to the residual generator selection problem, i.e., return a set of residual generators fulfilling the isolability and minimal cardinality requirements. The following result justifies Algorithm 2 with regard to the isolability requirement. That is, if, and only if, the isolability requirement can be fulfilled with the given method, then Algorithm 2 finds a set of residual generators with which the isolability requirement is fulfilled.

Theorem 2. *Let $M = (E, X, Y, F)$ be a model, $\mathcal{M}$ a residual generation method, and $\mathcal{F}$ an isolability requirement. Further, let M, $\mathcal{M}$, and $\mathcal{F}$ be input to Algorithm 2 and $\mathcal{R}$ the output. Then, $\mathcal{F}$ is fulfilled for M with $\mathcal{M}$ if and only if $\mathcal{F}$ is fulfilled with $\mathcal{R}$. If $\mathcal{F}$ is not fulfilled for M with $\mathcal{M}$, then $\mathcal{R}$ gives the maximum attainable isolability for M with $\mathcal{M}$, with respect to $\mathcal{F}$.*

Proof. According to rows 5, 6, 14, and 21, and rows 4, 7, 16, and 18, there are two different termination conditions in Algorithm 2; either $\mathcal{I} = \varnothing$ or $\mathcal{S}_{\mathcal{M}} = \varnothing$.

Consider first the case when Algorithm 2 terminates because of the condition on row 6, i.e., $\mathcal{I} = \varnothing$, and let n denote the total number of iterations performed by Algorithm 2 in which the condition on row 11 is met. Further let $\mathcal{S}_i$, $\mathcal{R}_i$, $\mathcal{I}_i$, S_i^*, and R_i, denote the values of the variables $\mathcal{S}$, $\mathcal{R}$, $\mathcal{I}$, S^*, and R, respectively, after iteration i. By assumption, and due to row 6, it holds that $\mathcal{I}_n = \varnothing$. Further, it holds that $\mathcal{S}_0 = \mathcal{R}_0 = \varnothing$, and $\mathcal{I}_0 = \mathcal{I}$. By assumption also $\mathcal{R} \neq \varnothing$ and therefore $\mathcal{R}_n \neq \varnothing$ and $\mathcal{S}_n \neq \varnothing$, due to rows 12 and 13. In fact, due to rows 10-12, it can be concluded that $\mathcal{R}_n = \bigcup_{i=1}^{n-1}\{R_i\}$, and $\mathcal{S}_n = \bigcup_{i=1}^{n-1}\{S_i^*\}$, where $R_i = \mathcal{M}(S_i^*)$, and thus each $S_i^* \in \mathcal{S}_n$ is realizable with $\mathcal{M}$ and the relation between $\mathcal{R}_n$ and $\mathcal{S}_n$ is the same as between $\mathcal{R}$ and $\mathcal{S}$ in (5). Moreover, due to rows 7-9, it holds that each $S_i^* \in \mathcal{S}_n$ is contained in $\mathcal{S}_{\mathcal{M}}$ and therefore $\mathcal{S}_n$ fulfills the prerequisites of Lemma 1. From row 14 it can be deduced that $\mathcal{I}_0 = \bigcup_{i=1}^{n-1} \sigma_{\mathcal{I}}(\{S_i^*\})$. From (8), it follows that for $i = 1, 2, \ldots, n-1$ and for all $I \in \sigma_{\mathcal{I}}(\{S_i^*\})$ it holds by definition that $S_i^* \in I$. Therefore, since $\mathcal{S}_n = \bigcup_{i=1}^{n-1}\{S_i^*\}$, it holds that $\mathcal{S}_n \bigcap I \neq \varnothing$ for all $I \in \mathcal{I}_0 = \bigcup_{i=1}^{n-1} \sigma_{\mathcal{I}}(\{S_i^*\})$. According to Lemma 1, this property of $\mathcal{S} = \mathcal{S}_n$ is equivalent to that $\mathcal{F}$ is fulfilled with $\mathcal{R} = \mathcal{R}_n$ which, due to Proposition 1, is equivalent to that $\mathcal{F}$ is fulfilled in M with $\mathcal{M}$.

Consider now instead the case when Algorithm 2 terminates because of the condition on row 7 and let n denote the total number of iterations in which the condition on row 11 is met. With similar arguments and notations as above, it holds that $\mathcal{R}_n = \bigcup_{i=1}^{n-1}\{R_i\}$ and $\mathcal{S}_n = \bigcup_{i=1}^{n-1}\{S_i^*\}$, where $R_i = \mathcal{M}(\mathcal{S}_i)$. Since termination of Algorithm 2 by assumption was due to the condition on row 7, it holds that $\mathcal{I}_n = \mathcal{I}_0 \setminus \left\{\bigcup_{i=1}^{n-1} \sigma_{\mathcal{I}}(\{S_i^*\})\right\} \neq \varnothing$. Thus, there exists $I \in \mathcal{I}_0$ such that $\mathcal{S}_n \bigcap I = \varnothing$ and consequently, by Lemma 1, it can be deduced that $\mathcal{F}$ not is fulfilled with $\mathcal{R} = \mathcal{R}_n$. However, if $\mathcal{I}' = \bigcup_{i=1}^{n-1} \sigma_{\mathcal{I}}(\{S_i^*\})$ and $\mathcal{F}' = \left\{(f_i, f_j,) \in \mathcal{F}' : I_{f_i f_j} \in \mathcal{I}'\right\}$, Lemma 1 implies that $\mathcal{F}'$ is fulfilled with $\mathcal{R}$. By assumption and row 7, it holds that $\mathcal{S}_{\mathcal{M}}^n = \varnothing$. Therefore, there are no $S \in \mathcal{S}_{\mathcal{M}}^n$ that can be used to isolate the fault pairs in $\mathcal{F} \setminus \mathcal{F}'$ and thus $\mathcal{F}'$ is the maximum attainable isolability for M with $\mathcal{M}$. □

Note that if the isolability requirement not can be fulfilled, the MHS-based Algorithm 1 will return an empty set due to the non-existence of minimal hitting sets. Algorithm 2 will instead provide the best possible solution, in terms of fault isolability, with regard to the given method. However, if the output from Algorithm 2 is an empty set, there are no realizable candidate equation sets that contribute to fulfill the stated isolability requirement.

The Minimal Cardinality Requirement

Theorem 2 does not regard the minimal cardinality requirement, i.e., nothing is said whether the set of residual generators obtained as the output from Algorithm 2 is of minimal cardinality or not. The purpose of this section is to analyze this.

To this end, consider the optimization problem formulation (7) of the residual generator selection problem. To be able to exploit a previous result regarding the qualification of the greedy heuristic used in Algorithm 2, a different but equivalent formulation of the underlying minimal hitting set problem, given by (7a) and (7c), is considered. Define the set

$$\mathcal{U}_{\mathcal{M}} = \{\sigma_{\mathcal{I}}(\{S\}) : \forall S \in \mathcal{S}_{\mathcal{M}}\}, \tag{10}$$

that is, $\mathcal{U}_{\mathcal{M}}$ is the collection of all isolability classes covered by each candidate equation set in $\mathcal{S}_{\mathcal{M}}$. Consider now the problem of finding a set $\mathcal{U} \subseteq \mathcal{U}_{\mathcal{M}}$ of minimal cardinality that covers $\mathcal{U}_{\mathcal{M}}$, i.e.,

$$\min_{\mathcal{U} \subseteq \mathcal{U}_{\mathcal{M}}} \quad |\mathcal{U}|, \quad \text{s.t.} \quad \bigcup_{U \in \mathcal{U}} U = \bigcup_{U \in \mathcal{U}_{\mathcal{M}}} U. \tag{11}$$

The problem (11) is referred to as a *set covering problem*, and can be shown to be equivalent to the previously considered minimal hitting set problem

$$\min_{\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}} \quad |\mathcal{S}|, \quad \text{s.t.} \quad \forall I \in \mathcal{I}, \quad \mathcal{S} \bigcap I \neq \varnothing, \tag{12}$$

that is, the selection problem (7) with the realizability condition (7b) relaxed. In fact, if $\mathcal{U}^*$ is a solution to the set covering problem (11), a solution $\mathcal{S}^*$ to the minimal hitting set problem (12) can be constructed by finding for each $U \in \mathcal{U}^*$ a $S \in \mathcal{S}_{\mathcal{M}}$ such that $\sigma_{\mathcal{I}}(\{S\}) = U$. The converse is given by (10) with $\mathcal{U}_{\mathcal{M}}$ and $\mathcal{S}_{\mathcal{M}}$ replaced by $\mathcal{U}^*$ and $\mathcal{S}^*$, respectively.

Consider now solving (11) approximately with a greedy heuristic equivalent to the one described in Section 6. Namely, in each iteration, until all isolability classes in $\mathcal{U}_{\mathcal{M}}$ are covered, select the one $U \in \mathcal{U}_{\mathcal{M}}$ that covers most uncovered isolability classes, i.e., the $U \in \mathcal{U}_{\mathcal{M}}$ of highest cardinality. Denote the resulting solution $\mathcal{U}$. It can be shown (Johnsson, 1974; Lovász, 1975), that

$$\frac{|\mathcal{U}|}{|\mathcal{U}^*|} \leq \sum_{j=1}^{k} \frac{1}{j} \leq \ln k + 1, \tag{13}$$

where $\mathcal{U}^*$ is an exact solution to (11) and k is the cardinality of the largest set in $\mathcal{U}_{\mathcal{M}}$.

As said, the greedy heuristic described above for solving problem (11) coincide with the heuristic described in Section 6 for solving problem (12). Since the two problems are equivalent, it can be concluded that the worst case bound (13) also holds for approximate solutions to (12) obtained by usage of the greedy heuristic described in Section 6. This fact is summarized in the following result.

Theorem 3. *Let $M = (E, X, Y, F)$ be a model, $\mathcal{M}$ a residual generation method, and $\mathcal{F}$ an isolability requirement. Further, let M, $\mathcal{M}$, and $\mathcal{F}$ be input to Algorithm 2 and $\mathcal{R}$ a non-empty output. Then,*

$$\frac{|\mathcal{R}|}{|\mathcal{R}^*|} \leq \sum_{j=1}^{k} \frac{1}{j} \leq \ln k + 1, \tag{14}$$

where $\mathcal{R}^*$ is the exact solution to the residual generator selection problem, and k is the cardinality of the largest set in $\mathcal{U}_{\mathcal{M}}$, defined according to (10).

Theorem 3 provides a measure, by means of a worst-case error bound, of how well the minimal cardinality requirement is met when solving the selection problem with Algorithm 2. Theorem 3 and Theorem 2 together provide a theoretical justification of Algorithm 2.

Note that if each candidate equation set in $\mathcal{S}_{\mathcal{M}}$ only covers a few of the isolability classes in $\mathcal{I}$, i.e., k is small, then Algorithm 2 performs well in the sense that the cardinality of its output is close to the cardinality of the exact solution to the selection problem. However, the larger the coverage, the worse the performance. Nevertheless, the approximation ratio (14) increases slowly with k, due to the function ln().

7 Sequential Residual Generation

The purpose of this section is to briefly describe the residual generation method (Svärd and Nyberg, 2010), which is considered in the application study in Section 8, and discuss its use in the framework of Section 3. Note however that the algorithms developed in Sections 5 and 6 are general in the sense that they are aimed at supporting any computerized residual generation method fulfilling Assumption 2, and not only this particular method.

The considered residual generation method belongs to a class of methods referred to as *sequential residual generation*, which has shown to be successful for real applications and also has the potential to be automated to a high extent. Sequential residual generation is based upon the ideas originally described in Staroswiecki and Declerck (1989), where unknown variables in a model are computed by solving equation sets one at a time in a sequence and a residual is obtained by evaluating a redundant equation. Similar approaches are described and exploited in for example Cassar and Staroswiecki (1997); Pulido and Alonso-González (2004); Ploix et al. (2005); Travé-Massuyès et al. (2006); Blanke et al. (2006).

7.1 Computation Sequence

Recall the model $M = (E, X, Z, F)$ considered in Section 3, where E is a set of equations, X a set of unknown variables, Y a set of known variables, and F a set of fault variables. An essential component in the design of a sequential residual generator is a *computation sequence*, describing the order and from which equations variables are computed. In Svärd and Nyberg (2010) a computation sequence is defined as an ordered set of variable and equation pairs

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \ldots, (V_k, E_k)), \tag{15}$$

where $V_i \subseteq X \bigcup D$, $E_i \subseteq E$, and D contains the first-order derivatives of the variables in X. The computation sequence $\mathcal{C}$ implies that first the variables in V_1 are computed from equations E_1, then the variables in V_2 from equations E_2 and so forth.

7.2 Sequential Residual Generator

Having computed the unknown variables in $V_1 \bigcup V_2 \bigcup \ldots \bigcup V_k$ according to the computation sequence $\mathcal{C}$ in (15), a residual can be obtained by evaluating a redundant equation e, i.e., $e \in E \setminus E_1 \bigcup E_2 \ldots \bigcup E_k$ with $\mathrm{var}_X(e) \subseteq \mathrm{var}_X(E_1 \bigcup E_2 \ldots \bigcup E_k)$, where the operator $\mathrm{var}_X(\cdot)$ returns the unknown variables that are contained in an equation set. A residual generator based on a computation sequence $\mathcal{C}$ and redundant residual equation e is referred to as a *sequential residual generator*.

For an example, consider again the model (1) considered in Section 3, where $E = \{e_1, e_2, e_3\}$, $X = \{x_1\}$, $Y = \{u, y_1, y_2\}$, and $F = \{f_1, f_2, f_3\}$. A computation sequence for the unknown variable x_1 is given by $\mathcal{C}_1 = ((\{\dot{x}_1\}, \{e_1\}))$. Given $\mathcal{C}_1$, e_2 is a redundant residual equation and the corresponding sequential residual generator is

$$\dot{x}_1 = -x_1 + u \tag{16a}$$

$$r = y_1 - x_1. \tag{16b}$$

In fact, also $\mathcal{C}_2 = ((\{x_1\}, \{e_2\}))$ and $\mathcal{C}_3 = ((\{x_1\}, \{e_3\}))$ are computation sequences for x_1. For instance, the sequential residual generator corresponding to $\mathcal{C}_2$ and the residual equation e_3 is

$$x_1 = y_1 \tag{17a}$$

$$r = y_2 - x_1. \tag{17b}$$

7.3 Residual Generation Method

Algorithm 3, see Svärd and Nyberg (2010), constructs a sequential residual generator given an equation set S. The output from the algorithm is a sequential residual generator R, if S is realizable with the method, else an empty set.

The realization of an equation set with the considered sequential residual generation method relies heavily on the procedure findComputationSequence, which finds a *minimal* and *irreducible* computation sequence $\mathcal{C}$ for the variables X. Whether it is possible or not to find a computation sequence for a set of variables depends naturally

Algorithm 3 Realization of a Sequential Residual Generator

```
Input: Equation set $S$
Output: Sequential residual generator $R$
procedure sequentialResidualGeneration($S$)
    $X \leftarrow \mathrm{var}_X(S)$
    for all $e \in S$ do
        $S' \leftarrow S \setminus \{e\}$
        $\mathcal{C} \leftarrow$ findComputationSequence($S', X$)
        if $\mathcal{C} \neq \emptyset$ then
            $R \leftarrow \{\mathcal{C} \cup e\}$
            return $R$
        end if
    end for
    return $\emptyset$
end procedure
```

on the properties of the equations. Equally important are however prerequisites in terms of *causality assumption*, i.e., regarding integral and/or derivative causality, and the properties of the *computational tools*, that are available for use.

7.4 Fault Sensitivity

In Section 3, it was assumed that a residual generation method satisfies Assumption 2. If a residual generation method $\mathcal{M}$ satisfies Assumption 2 it is guaranteed that the residual generator $R = \mathcal{M}(S)$ is sensitive to a fault f if $e_f \in S$. Thus, to verify that the residual generation method given as Algorithm 3 satisfies Assumption 2 it must be shown that a non-empty output R from Algorithm 3 is sensitive to fault f if and only if $e_f \in S$, when $S \subseteq E$ is input.

Assume first that $e_f \notin S$ and note that this implies that no equation in S is affected if fault f is present. Since only equations in S are used in the sequential residual generator $R = \mathcal{M}(S)$ it follows that R can not be sensitive to f.

For the converse, assume that $e_f \in S$ and note that a sequential residual generator consists of a computation sequence and a residual equation. It therefore holds that $R = \{\mathcal{C} \cup e\}$, where $\mathcal{C}$ is a computation sequence for $\mathrm{var}_X(S)$ and e a residual equation. For $R = \{\mathcal{C} \cup e\}$ to be sensitive to fault f, it is necessary that $e = e_f$ or that e_f is contained in any of the equations in $\mathcal{C}$, i.e., $e_f \in E_1 \cup E_2 \cup \ldots \cup E_k$ where $E_i \subseteq E$ when $\mathcal{C}$ is given by (15). Since the former case is trivial due to the fact that $e \in S$, consider the latter and assume that e_f is *not* used in $\mathcal{C}$. This implies that there exists a computation sequence $\mathcal{C}'$ for $\mathrm{var}_X(S)$ such that $\mathcal{C}' \subset \mathcal{C}$. However, according to Theorem 4 in Svärd and Nyberg (2010), a non-empty $\mathcal{C}$ returned by findComputationSequence in Algorithm 3 is a *minimal* and *irreducible* computation sequence for $\mathrm{var}_X(S)$. Therefore $\mathcal{C}' \subset \mathcal{C}$ contradicts the minimality of $\mathcal{C}$ and it follows that e_f must be used in $\mathcal{C}$.

It then remains to show that $R = \{\mathcal{C} \cup e\}$ is sensitive to f if $e_f \in E_1 \cup E_2 \cup \ldots \cup E_k$, where $E_i \subseteq E$, or $e_f = e$. Since no restrictions are placed on the model equations E, nothing can in general be guaranteed regarding the analytical properties of the equations in $E_1 \cup E_2 \cup \ldots \cup E_k \cup e$. In particular, nothing can be said regarding how the fault f influences the equation e_f in $E_1 \cup E_2 \cup \ldots \cup E_k \cup e$ and consequently nor how f influences the residual generator $R = \{\mathcal{C} \cup e\}$. In addition, the effect of f in R is highly dependent on the size and temporal properties of f, and also on for example the current operating conditions. In order verify that R is sensitive to f, it is thus necessary to implement and run R using representative data from relevant fault cases.

In conclusion, it is hard to theoretically verify that R is sensitive to fault f, given the prerequisites and the general model class considered in this work. It should though be noted that under the idealized assumption that $R = \{\mathcal{C} \cup e\}$ is sensitive to f if $e_f \in \mathcal{C}$ or $e_f = e$, the residual generation method given as Algorithm 3 satisfies Assumption 2. Empirical studies have however shown that Assumption 2 mostly holds in practice. In particular, this is true for the automotive engine system considered in Sections 2 and 8. This is discussed in Section 8.5 and exemplified in Figure 7.

7.5 Necessary Realizability Criterion

In Svärd and Nyberg (2010, Theorem 2), it is shown that the equations in a minimal and irreducible computation sequence together with a redundant residual equation, in fact correspond to a Minimal Structurally Overdetermined (MSO) set, see Krysander et al. (2008). As said above, a non-empty computation sequence returned by findComputationSequence in Algorithm 3 is indeed minimal and irreducible. Thus, if an equation set S is realizable with the sequential residual generation method then S is an MSO set. Consequently, a necessary realizability criterion for the method is that the equation set used as input is an MSO set and hence an MSO set is a candidate equation set for the method. There are efficient algorithms for finding all MSO sets in a large set of equations, see, e.g., Krysander et al. (2008).

For the model (1), it is possible to find in total three MSO sets. These are given by $S_1 = \{e_1, e_2\}$, $S_2 = \{e_1, e_3\}$, and $S_3 = \{e_2, e_3\}$. In fact, the sequential residual generators (16) and (17) are created from the MSO sets S_1 and S_3, respectively.

As a side remark, note that the maximum number of sequential residual generators that can be constructed from an MSO set equals the number of equations in the set. All residual generators created from the same MSO set however have equal fault sensitivity properties according Assumption 2. Nevertheless, their actual fault sensitivity may differ due for example different sensitivity for noise, etc. To make the final selection of which of the residual generators created from an MSO set that should be included in the final diagnosis system, evaluation by means on execution using real measurements from different fault cases might be needed. For this purpose, Algorithm 3 can be trivially modified to return all residual generators that can be created from the MSO set used input, and not only one.

Table 1: Considered Faults

Fault	Description
$f_{W_{\mathrm{ic}}}$	Leakage, intercooler
$f_{W_{\mathrm{im}}}$	Leakage, intake manifold
$f_{W_{\mathrm{em}}}$	Leakage, exhaust manifold
$f_{u_{x_{\mathrm{th}}}}$	Fault, throttle position actuator
$f_{u_{x_{\mathrm{egr}}}}$	Fault, EGR-valve position actuator
$f_{u_{x_{\mathrm{vgt}}}}$	Fault, VGT-valve position actuator
$f_{y_{p_{\mathrm{amb}}}}$	Fault, ambient pressure sensor
$f_{y_{T_{\mathrm{amb}}}}$	Fault, ambient temperature sensor
$f_{y_{p_{\mathrm{ic}}}}$	Fault, intercooler pressure sensor
$f_{y_{p_{\mathrm{im}}}}$	Fault, intake manifold pressure sensor
$f_{y_{T_{\mathrm{im}}}}$	Fault, intake manifold temperature sensor
$f_{y_{p_{\mathrm{em}}}}$	Fault, exhaust manifold pressure sensor

8 Application Example

In this section, the selection algorithms presented in Section 5 and 6 are applied to the automotive engine system introduced in Section 2. The residual generation method considered in this study is briefly outlined in Section 7.

8.1 The Automotive Engine System

Consider again the Scania truck diesel engine system introduced in Section 2, which is shown in Figure 1. The main incentive for diagnosis of this system is the stricter emission legislation requirements for heavy-duty trucks, which in turn implies stricter on-board diagnosis (OBD) legislation requirements. The OBD-legislation states that all manufactured vehicles must be equipped with a diagnosis system capable of detecting and isolating faults in all components that, if broken, result in emissions above pre-defined OBD-thresholds during a specified test cycle.

For the considered system, emission critical components include all actuators and sensors, and to meet the OBD-requirements it is desirable that, at least, single faults in these can be detected and isolated. Other emission critical components are pipes and hoses. In particular, a broken pipe or hose may lead to gas-leakage which may increase emissions. Leakages in or near the intercooler, intake manifold, and exhaust manifold are particularly critical. It is desirable that these leakages can be detected and isolated, from each other, but also from all sensor and actuator faults. In total, there are 12 emission critical components and consequently 12 faults that should be isolated from each other in the system. All the 12 considered faults for the system, along with their description, can be found in Table 1.

The Model

The model of the system used in this work is described in Wahlström and Eriksson (2011) and relies on both fundamental first principle physics and gray-box modeling. The model describes the behavior of the system in the no-fault case, i.e., it is a *nominal* model. To incorporate fault information in the nominal model, faults are modeled as additive signals in corresponding equations. For example, fault $f_{y_{p_{\text{im}}}}$, representing a fault in the intake manifold pressure sensor $y_{p_{\text{im}}}$, is modeled by simply adding $f_{y_{p_{\text{im}}}}$ to the equation describing the relation between the sensor value $y_{p_{\text{im}}}$ and the actual intake manifold pressure p_{im} according to $y_{p_{\text{im}}} = p_{\text{im}} + f_{y_{p_{\text{im}}}}$.

The model contains in total 46 equations, 43 unknown variables, 11 known variables, and the 12 faults in Table 1. Of the 11 known variables, 3 are actuators, 6 are sensors, and 2 are control inputs. Of the 46 equations, 5 are differential equations and the rest are algebraic equations. The model contains several non-linear functions.

The Isolability Requirement

Since it is required that the 12 considered faults can be isolated from each other, the isolability requirement $\mathcal{F}$ for the truck diesel engine system consists of all unique pairwise combinations of the faults in Table 1. That is,

$$\mathcal{F} = \left\{ (f_{W_{\text{ic}}}, f_{W_{\text{im}}}), (f_{W_{\text{ic}}}, f_{W_{\text{em}}}), \dots, \left(f_{y_{T_{\text{im}}}}, f_{y_{p_{\text{em}}}}\right) \right\}, \tag{18}$$

with $|\mathcal{F}| = 12 \times 11 = 132$.

8.2 Appliance of the MHS-Based Algorithm

There exists in total 270 candidate equation sets, here MSO sets, for the considered sequential residual generation method in the truck diesel engine system model, i.e., $|\mathcal{S}_{\mathcal{M}}| = 270$. The MSO sets were found using the algorithm (Krysander et al., 2008), which was implemented as the procedure findCES.

As said in Section 7.5, the largest possible number of sequential residual generators that can be constructed from an MSO set equals the number of equations in the set. Thus, the maximum number of residual generators that can be constructed from a set of MSO sets is the sum of the number of equations for all MSO sets. From the set of 270 MSO sets found in the automotive engine system model this number equals 14,242. This is the rationale behind the total number of candidate residual generators mentioned in Section 2.

Given the 270 candidate equation sets and the isolability requirement $\mathcal{F}$ defined in (18), 132 isolability classes were created according to (3) and (4), that is, $|\mathcal{I}| = 132$. Due to the complexity of the selection problem, in terms of the cardinalities of the sets $\mathcal{S}_{\mathcal{M}}$ and $\mathcal{I}$, it was impossible to find the collection of all minimal hitting sets for $\mathcal{I}$ and consequently impossible to use the MHS-based Algorithm 1 to solve the automotive engine selection problem.

Some insight regarding the complexity of the selection problem can be gained by studying the total number of minimal hitting sets for smaller instances of the problem.

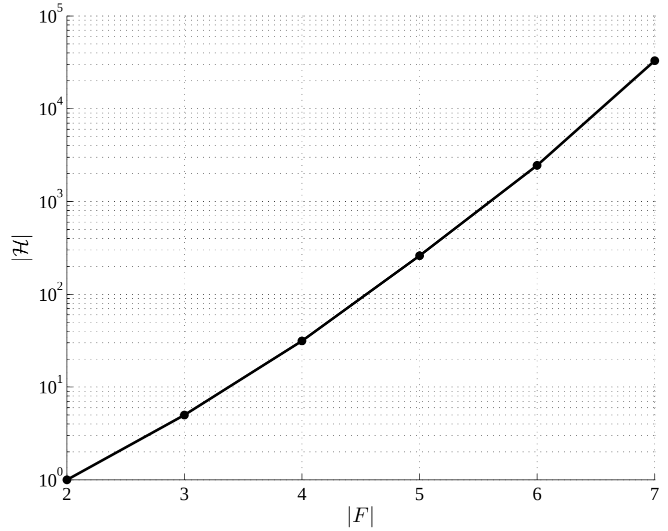


Figure 3: The total number of minimal hitting sets, $|\mathcal{H}|$, as function of the cardinality of the set of considered faults, $|F|$. The number of minimal hitting sets grows rapidly with the number of faults.

One simple way to reduce the size of the selection problem is to consider only a subset of the faults in Table 1, and then calculate $\mathcal{F}$ and $\mathcal{I}$ for this smaller set of faults. For each cardinality number, several randomized subsets of faults were chosen from the set of 12 faults. Figure 3 presents, in logarithmic scale, the mean cardinality of the set of all minimal hitting sets, $|\mathcal{H}|$, as a function of the cardinality of the set of considered faults, $|F|$. The minimal hitting sets were computed using a C++ implementation of the algorithm presented in de Kleer and Williams (1987). From Figure 3 it can be seen that the number of minimal hitting sets grows rapidly with the number of faults, and that the total number of minimal hitting sets is over 30,000 already for 7 faults. Given this, it is not that surprising that the problem with 12 faults was not possible to solve.

Using Improvements of the Algorithm

Two possible improvements of Algorithm 1 were suggested in Section 5.2. One of the proposed improvements was to consider the realizable subset of all candidate equation sets and thereby reduce the size of the involved minimal hitting set problem.

This approach however requires that the realizability of all candidate equation sets are evaluated which, as argued in Section 5.2, may be a computational demanding task. With a Matlab implementation of the sequential residual generation method outlined in Section 7, the realizability evaluation required 15,778 s ≈ 4.38 h on a 2.4 GHz Intel Core 2 Duo PC running Windows XP. In total, only 59 of the 270 candidate equation sets (21.9%) were realizable with the considered sequential residual generation method. The main cause of this relatively large fraction of non-realizable candidate equation sets is non-invertible non-linear functions in the automotive engine model, see Svärd et al. (2011) for a discussion of a similar result regarding a similar model.

By using the set of 59 realizable candidate equation sets, the size of the selection problem is substantially reduced. Even for this smaller problem, it was unfortunately not possible to compute the set of all minimal hitting sets within feasible time, no termination after 24 h, using the same C++ implementation as above of the minimal hitting set algorithm (de Kleer and Williams, 1987).

The other improvement of Algorithm 1 suggested in Section 5.2 is to use an approximative MHS-algorithm to compute a subset of all minimal hitting sets. Neither this approach did succeed, since it was impossible to find a realizable minimal hitting set within feasible time due to the large number of non-realizable candidate equation sets.

8.3 Appliance of the Greedy Algorithm

Since it was impossible to use the MHS-based Algorithm 1, or any of the two suggested improvements, to solve the automotive engine selection problem, the greedy Algorithm 2 was employed.

Algorithm 2 was implemented in Matlab. The realization procedure $\mathcal{M}(\cdot)$ was implemented according to Algorithm 3, and the procedure findComputationSequence, for finding computation sequences, according to the corresponding algorithm in Svärd and Nyberg (2010).

Given the isolability requirement (18) and the automotive engine system model, Algorithm 2 returned a set of 11 residual generators. All of the 11 residual generators were dynamic, 3 used only integral causality and the remaining 8 both integral and derivative causality, i.e., mixed causality. Before terminating, the algorithm discarded in total 119 non-realizable candidate equation sets, mainly due to non-invertible non-linear functions in the model.

Table 2 shows the fault signature matrix for the 11 selected residual generators with respect to the faults in Table 1. The fault signature for a residual generator R contains an "x" in the column corresponding to fault f, if R is sensitive to f in the context of Assumption 2.

As seen in Table 2, all of the 11 selected residual generators are sensitive to the faults $f_{y_{p_{\text{amb}}}}$ and $f_{u_{x_{\text{vgt}}}}$. This is also indicated in Table 3, which shows the resulting isolability matrix for the set of selected residual generators. Clearly, faults $f_{y_{p_{\text{amb}}}}$ and $f_{u_{x_{\text{vgt}}}}$ are not isolable from the other faults and the isolability requirement $\mathcal{F}$, defined in (18), is not met. However, according to Theorem 2, Table 3 shows the maximum attainable isolability in the automotive engine model with the considered sequential residual generation method.

8.4 Analysis of the Cardinalities of Greedy Solutions

As said in Section 6.3, the greedy Algorithm 2 provides an approximate solution when it comes to fulfillment of the minimal cardinality requirement. Thus, the above mentioned solution to the automotive engine selection problem, i.e., the set of 11 residual generators, may therefore not be of minimal cardinality.

Table 2: Fault Signature Matrix

	$f_{W_{\text{ic}}}$	$f_{W_{\text{im}}}$	$f_{W_{\text{em}}}$	$f_{y_{p_{\text{amb}}}}$	$f_{y_{p_{\text{amb}}}}$	$f_{y_{p_{\text{ic}}}}$	$f_{y_{p_{\text{im}}}}$	$f_{y_{T_{\text{im}}}}$	$f_{y_{p_{\text{em}}}}$	$f_{u_{x_{\text{th}}}}$	$f_{u_{x_{\text{egr}}}}$	$f_{u_{x_{\text{vgt}}}}$
R_1			x	x	x	x	x	x	x		x	x
R_2	x	x	x	x	x		x	x	x	x		x
R_3	x	x		x	x	x	x		x	x	x	x
R_4	x	x	x	x	x	x		x		x	x	x
R_5		x	x	x		x	x	x	x	x	x	x
R_6	x	x	x	x	x	x	x	x	x			x
R_7	x		x	x	x		x	x	x	x	x	x
R_8	x	x		x	x	x		x	x	x	x	x
R_9	x	x	x	x	x	x	x		x		x	x
R_{10}	x	x	x	x	x	x	x	x			x	x
R_{11}	x		x	x		x	x	x	x	x	x	x

To investigate the performance of Algorithm 2 with respect to the minimal cardinality requirement, it is necessary to know the cardinality of an exact, i.e., minimal cardinality, solution to the selection problem. As said in Section 8.2 it is unfortunately not possible to find all minimal hitting sets for the selection problem when all 12 faults are considered and consequently not possible to find an exact solution using Algorithm 1. There are however algorithms (de Kleer, 2011) that are able to compute *one* minimal cardinality hitting set for this problem. In practice, this is not sufficient since the obtained minimal cardinality hitting set may contain non-realizable candidate equation sets, see Section 5.2. However, from a theoretical point of view and for this investigation, this is sufficient.

For several different instances of the selection problem, and under the assumption that all candidate equation sets were realizable, one greedy solution and one exact, i.e., minimal cardinality, solution were computed. The different instances were obtained by using randomized subsets, of varying cardinality, of the 12 faults in Table 1. Figure 4 shows the median cardinalities of the exact, $|\mathcal{R}^*|$, and greedy, $|\mathcal{R}|$, solutions as functions of the cardinality of the set of considered faults, $|F|$.

According to Figure 4, the median cardinalities of the greedy and exact solutions coincide in a majority of the cases. Consequently, it can be concluded that this selection problem suits the greedy selection approach well. Thus, it is likely that the set of 11 residual generators obtained as solution to the selection problem with 12 considered faults in Section 8.3, is of minimal cardinality, or at least in close proximity.

Figure 5 shows the mean execution times, in logarithmic scale, for the exact and greedy algorithms for the runs described above. Both algorithms were implemented in Matlab and executed on a 2.4 GHz Intel Core 2 Duo PC running Windows XP. Clearly, the greedy algorithm is magnitudes faster than the exact algorithm. Note that the execution time for computing a minimal cardinality hitting set for the problem with 12 faults is in the magnitude of hundreds of hours.

It is also interesting to evaluate the greedy solution to the truck diesel engine selection problem by comparing it with the worst-case bound (14), given in Theorem 3. This bound,

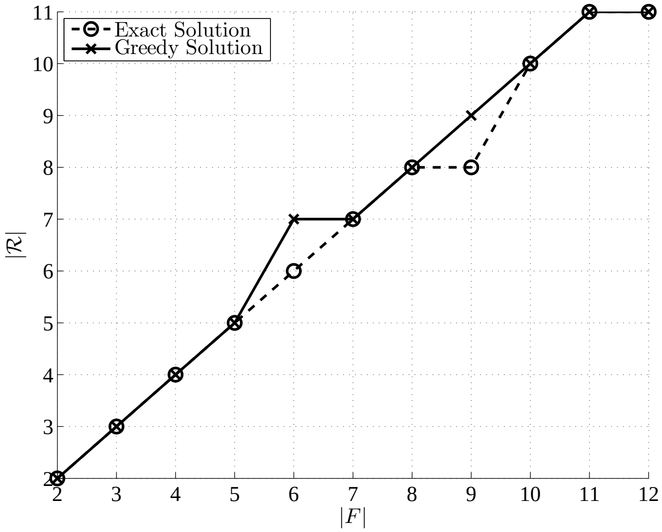


Figure 4: Median cardinalities of exact and greedy solutions, as functions of the cardinality of the set of considered faults, to the automotive engine selection problem.

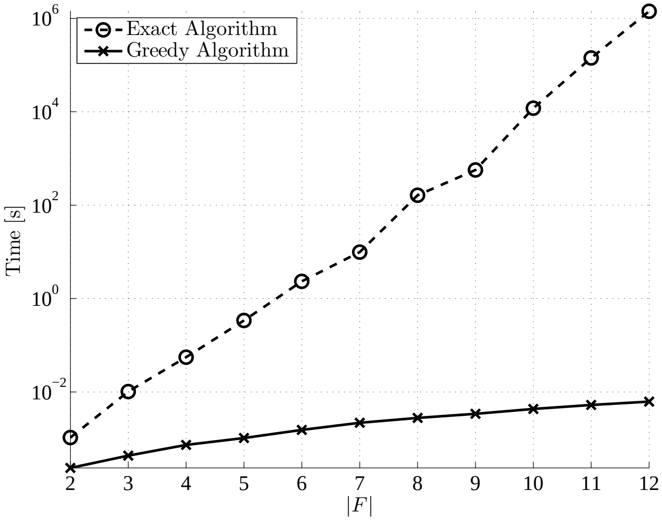


Figure 5: Mean execution times for the exact and greedy minimal cardinality hitting sets algorithms, as functions of the cardinality of the set of considered faults, for the automotive engine selection problem.

Table 3: Isolability Matrix

	$f_{W_{\mathrm{ic}}}$	$f_{W_{\mathrm{im}}}$	$f_{W_{\mathrm{em}}}$	$f_{y_{p_{\mathrm{amb}}}}$	$f_{y_{T_{\mathrm{amb}}}}$	$f_{y_{p_{\mathrm{ic}}}}$	$f_{y_{p_{\mathrm{im}}}}$	$f_{y_{T_{\mathrm{im}}}}$	$f_{y_{p_{\mathrm{em}}}}$	$f_{u_{x_{\mathrm{th}}}}$	$f_{u_{x_{\mathrm{egr}}}}$	$f_{u_{x_{\mathrm{vgt}}}}$
$f_{W_{\mathrm{ic}}}$	X			X								X
$f_{W_{\mathrm{im}}}$		X		X								X
$f_{W_{\mathrm{em}}}$			X	X								X
$f_{y_{p_{\mathrm{amb}}}}$				X								X
$f_{y_{T_{\mathrm{amb}}}}$				X	X							X
$f_{y_{p_{\mathrm{ic}}}}$				X		X						X
$f_{y_{p_{\mathrm{im}}}}$				X			X					X
$f_{y_{T_{\mathrm{im}}}}$				X				X				X
$f_{y_{p_{\mathrm{em}}}}$				X					X			X
$f_{u_{x_{\mathrm{th}}}}$				X						X		X
$f_{u_{x_{\mathrm{egr}}}}$				X							X	X
$f_{u_{x_{\mathrm{vgt}}}}$				X								X

along with the median cardinalities of the greedy solutions are shown in Figure 6, for the same instances of the selection problem used above. It can be seen that the cardinalities of the greedy solution differ substantially from the worst-case bound. From this and the fact that the cardinalities of the greedy solutions are more or less equal to the cardinalities of the exact solutions, according to Figure 4, it can be concluded that for the automotive engine selection problem, the bound (14) is very conservative.

8.5 Case Study of Fault Sensitivity

In this section it is shown that the considered approach for design of residual generators, i.e., the proposed selection algorithm together with the residual generation method (Svärd and Nyberg, 2010), is applicable to real-world systems characterized by, e.g., uncertain models and noisy measurements. This is done by illustrating how two of 11 residual generators obtained in Section 8.3 can be used to isolate a pair of faults from each other.

The first residual generator, denoted R_2 in Table 2, adopts mixed causality with three state variables and two numerically differentiated measurement signals. The estimated derivatives are of first-order. The residual generator uses in total 11 of the 12 known variables as input. The second residual generator, denoted R_4, contains 5 state variables and uses 9 known variables as input. This residual generator uses integral causality only.

The considered faults are $f_{y_{p_{\mathrm{im}}}}$ and $f_{y_{p_{\mathrm{ic}}}}$, i.e., faults in the intake manifold pressure sensor and intercooler pressure sensor, respectively. According to Table 2, residual generator R_2 is sensitive to fault $f_{y_{p_{\mathrm{im}}}}$ but not to fault $f_{y_{p_{\mathrm{ic}}}}$. The residual generator R_4, on the other hand, is sensitive to $f_{y_{p_{\mathrm{ic}}}}$ but not to $f_{y_{p_{\mathrm{im}}}}$. Note that the fault sensitivity in Table 2 is in the context of Assumption 2, see Section 7.4 for a further discussion regarding this.

The residual generators were implemented in a Matlab/Simulink environment

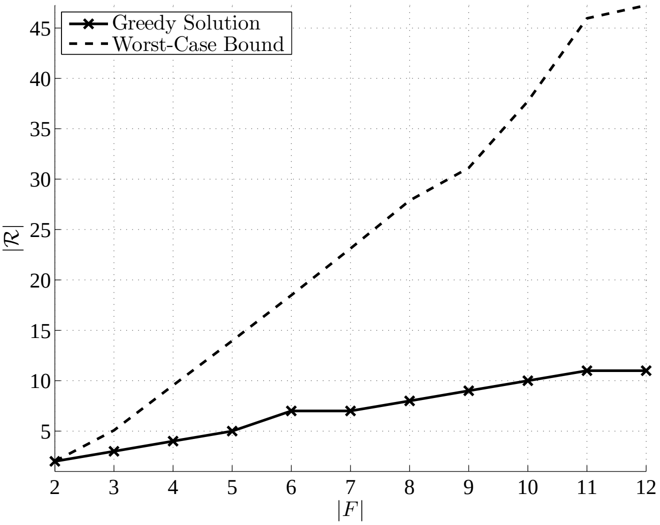


Figure 6: The median cardinalities of the greedy solution to the truck diesel engine selection problem compared with the worst-case bound provided in Theorem 3.

and run off-line. As input data, a set of measurements from an engine test bed during a World Harmonized Test Cycle (WHTC) was used. In two separate runs, faults in the intake manifold pressure sensor p_{im} and intercooler pressure sensor p_{ic} were injected. Both faults were in the form of a 20% positive gain of the corresponding pressure sensor signal, i.e., $y_{p_{\text{im}}} = 1.2 \cdot p_{\text{im}}$ and $y_{p_{\text{ic}}} = 1.2 \cdot p_{\text{ic}}$ where p_{im} and p_{ic} are the actual intake manifold pressure and intercooler pressure signals, respectively.

The residuals obtained as output from the residual generators R_2 and R_4, for each of the faults $f_{y_{p_{\text{im}}}}$ and $f_{y_{p_{\text{ic}}}}$, are shown in Figure 7. From the figure it can be seen that residual generator R_2 (top figure) responds to the fault $f_{y_{p_{\text{im}}}}$ but not to fault $f_{y_{p_{\text{ic}}}}$, and that residual generator R_4 (bottom figure) responds to fault $f_{y_{p_{\text{ic}}}}$ but not to fault $f_{y_{p_{\text{im}}}}$. Clearly, for these fault cases, R_2 is indeed sensitive to $f_{y_{p_{\text{im}}}}$ but not to $f_{y_{p_{\text{ic}}}}$, and R_4 sensitive to $f_{y_{p_{\text{ic}}}}$ but not to $f_{y_{p_{\text{im}}}}$. Thus, fault $f_{y_{p_{\text{im}}}}$ is isolable from fault $f_{y_{p_{\text{ic}}}}$ and vice versa, with the residual generators R_2 and R_4.

9 Conclusions

Two novel algorithms for solving the residual generator selection problem have been proposed. The foundation for both algorithms was a formulation of the selection problem, in the form of an optimization problem, where the isolability requirement was equivalently stated in terms of properties of subsets of the model equations. The formulation enabled an efficient reduction of the search-space by taking the realizability properties of equation subsets, with respect to the considered residual generation method, into account. Both algorithms are general in the sense that they are aimed at supporting any computerized

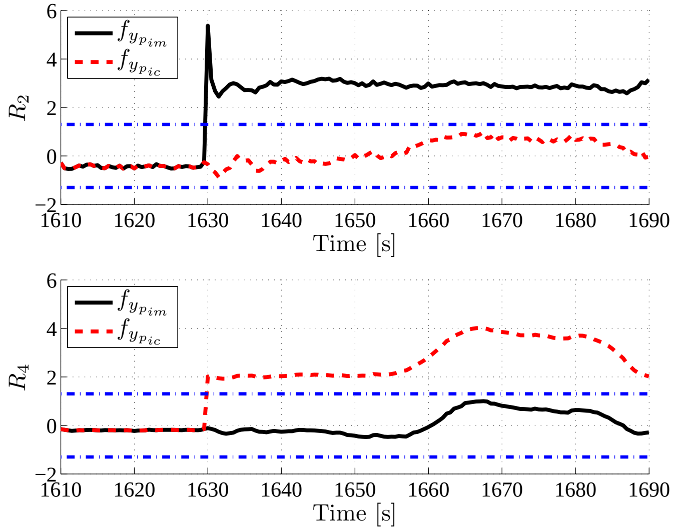


Figure 7: Residuals from residual generator R_2 (top figure) and residual generator R_4 (bottom figure) for the fault cases $f_{y_{p_{\rm im}}}$ (solid lines) and $f_{y_{p_{\rm ic}}}$ (dashed lines). Both faults are injected at $t = 1630s$. The dash dotted lines suggest how thresholds may be set in order to detect the faults.

residual generation method.

Algorithm 1, based on the naive approach of finding all minimal hitting sets, gives an exact solution fulfilling both the isolability and the minimal cardinality requirements but is intractable for large problems. Algorithm 2 is suitable for large, real-world, problems and is based on a greedy heuristic. It provides an approximate solution in terms of fulfilling the minimal cardinality requirement. A theoretical characterization of the approximation error, in the form of a worst-case bound, was given in Theorem 3, and that the output of Algorithm 2 indeed fulfills the isolability requirement was guaranteed by Theorem 2.

The problem of selecting a set of residual generators for detection and isolation of faults in a complex automotive engine system was considered as an industrial application example. Due to the significant complexity of this problem, it was not possible to use the exact MHS-based Algorithm 1 and instead the approximative greedy Algorithm 2 was employed. For this selection problem, the greedy algorithm provides a near-exact solution at a very low cost.

Acknowledgment

This work was sponsored by Scania and VINNOVA (Swedish Governmental Agency for Innovation Systems).

References

R. Abreu and A. J. C van Gemund. A low-cost approximate minimal hitting set algorithm and its application to model-based diagnosis. In V. Bulitko and J. C. Beck, editors, *Proceedings of the Eighth Symposium on Abstraction, Reformulation, and Approximation*, pages 2–9, Lake Arrowhead, California, USA, September 2009.

A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

G. Ausiello, A. D'Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1):136 – 153, 1980. doi:10.1016/0022-0000(80)90046-X.

P. E. Black. Greedy algorithm. Dictionary of Algorithms and Data Structures (online), U.S. National Institute of Standards and Technology, February 2005. http://tinyurl.com/3x5zzpp, Accessed: 2010-09-13.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, second edition, 2006.

J. P. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *Proceedings of IFAC Control Ind. Syst.*, pages 841–846, Belfort, France, 1997.

J. Chen and R. J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

J. de Kleer. Hitting set algorithms for model-based diagnosis. In *Proceedings of 22nd International Workshop on Principles of Diagnosis (DX-11)*, Murnau, Germany, 2011.

J. de Kleer and B. C Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1): 97–130, 1987.

M. R. Garey and D. S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

E. R. Gelso, S. M. Castillo, and J. Armengol. An algorithm based on structural analysis for model-based fault diagnosis. *Artificial Intelligence Research and Development*, 184: 138–147, 2008.

D. S Johnsson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computation*, pages 85–103, New York, 1972. Plenum Pres.

M. Krysander. *Design and Analysis of Diagnosis Systems Using Structural Methods.* PhD thesis, Linköpings universitet, June 2006.

M. Krysander, J. Åslund, and M. Nyberg. An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(1):197–206, 2008.

L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math*, 1975.

M. Nyberg. Automatic design of diagnosis systems with application to an automotive engine. *Control Engineering Practice*, 87(8):993–1005, 1999.

M. Nyberg and M. Krysander. Statistical properties and design criterions for AI-based fault isolation. In *Proceedings of the 17th IFAC World Congress*, pages 7356–7362, Seoul, Korea, 2008.

R. J. Patton, P. M. Frank, and R. N. Clark, editors. *Issues of Fault Diagnosis for Dynamic Systems.* Springer, 2000.

S. Ploix, M. Desinde, and S. Touaf. Automatic design of detection tests in complex dynamic systems. In *Proceedings of 16th IFAC World Congress*, Prague, Czech Republic, 2005.

B. Pulido and C. Alonso-González. Possible conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics, Special Issue on Diagnosis of Complex Systems*, 34(5):2192–2206, 2004.

M. Staroswiecki and P. Declerck. Analytical redundancy in non-linear interconnected systems by means of structural analysis. In *Proceedings of IFAC AIPAC'89*, pages 51–55, Nancy, France, 1989.

C. Svärd and M. Nyberg. Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(6):1310–1328, 2010.

C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Residual evaluation for fault diagnosis by data-driven analysis of non-stationary probability distributions. In *Proceedings of the 50:th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011)*, pages 95–102, 2011.

L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component-supported analytical redundancy. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 36(6):1146–1160, 2006.

J. Wahlström and L. Eriksson. Modeling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 225(7), July 2011.

Paper C

Data-Driven and Adaptive Statistical Residual Evaluation for Fault Detection with an Automotive Application[☆]

[☆] A revised version has been submitted to *Mechanical Systems and Signal Processing*, 2012.

Data-Driven and Adaptive Statistical Residual Evaluation for Fault Detection with an Automotive Application

Carl Svärd, Mattias Nyberg, Erik Frisk, and Mattias Krysander

Vehicular Systems, Department of Electrical Engineering,
Linköping University, SE-581 83 Linköping, Sweden.

Abstract

An important step in model-based fault detection is residual evaluation, where residuals are evaluated with the aim to detect changes in their behavior caused by faults. To handle residuals subject to time-varying uncertainties and disturbances, which indeed are present in practice, a novel statistical residual evaluation approach is presented. The main contribution is to base the residual evaluation on an explicit comparison of the probability distribution of the residual, estimated online using current data, with a no-fault residual distribution. The no-fault distribution is based on a set of a-priori known no-fault residual distributions, and is continuously adapted to the current situation. As a second contribution, a method is proposed for estimating the required set of no-fault residual distributions off-line from no-fault training data. The proposed residual evaluation approach is evaluated with measurement data on a residual for diagnosis of the gas-flow system of a Scania truck diesel engine. Results show that small faults can be reliable detected with the proposed approach in cases where regular methods fail.

1 Introduction

Fault diagnosis is becoming more and more important with the increasing demand for dependable technical systems, driven mostly by economical, environmental, and safety, incentives. One example is automotive systems, where good fault diagnosis is essential in order to meet customer demands regarding up-time, efficient repair and maintenance, and also to fulfill on-board diagnosis (OBD) legislative regulations.

Model-based fault diagnosis typically comprises fault detection and isolation (Blanke et al., 2006), and the fault detection part contains the essential steps residual generation and residual evaluation. In the first step, a model of the system is used together with measurements to generate residuals. In the second step, the residuals are evaluated with the aim to detect changes in the residual behavior caused by faults in the system. This works concerns the second step, residual evaluation.

Ideally, residuals are signals that are zero when no faults are present in the system, and non-zero otherwise. Due to the presence of uncertainties and disturbances, caused by for instance modeling errors, measurement noise, and unmodeled phenomena, residuals typically however deviate from zero even in the no-fault case. Moreover, due to changes in the operating mode of the system, the magnitude of these uncertainties and disturbances is time-varying, causing the behavior of residuals to be non-stationary. An illustration is given by Figure 1, where a residual for fault detection in the gas-flow system of a truck diesel engine is shown. Clearly, the residual is not zero in the no-fault case, and it is obvious that the residual exhibit non-stationary features. It can also be noted that the difference between the residual in the no-fault and fault cases is time-varying. Nevertheless, the fact that there is a difference implies that the present fault is potentially detectable.

There are two main approaches (Ding et al., 2007) for residual evaluation; statistical (Willsky and Jones, 1976; Gertler, 1998; Basseville and Nikiforov, 1993; Peng et al., 1997; Al-Salami et al., 2006; Blas and Blanke, 2011; Wei et al., 2011) and norm-based (Emami-Naeini et al., 1988; Frank, 1995; Frank and Ding, 1997; Sneider and Frank, 1996; Chen and Patton, 1999; Zhang et al., 2002; Zhong et al., 2007/03/; Ingimundarson et al., 2008; Al-Salami et al., 2010; Li et al., 2011; Abid et al., 2011). Statistical approaches exploits the framework of statistical hypothesis testing in order to detect changes in some parameter of the probability distribution of the residual, typically by means of likelihood ratio testing (Gustafsson, 2000). In norm-based approaches, residual evaluation is typically done by adaptive or constant thresholding of some norm of the residual.

Apparently, when encountering a residual as the one depicted in Figure 1, neither statistical-based approaches assuming stationary probability distributions, nor norm-based approaches using constant thresholds, would be successful. A potential solution is to consider adaptive thresholds (Clark, 1989; Frank, 1994), and use a-priori knowledge, either qualitative (Ingimundarson et al., 2008; Zhang et al., 2002; Höfling and Isermann, 1996; Emami-Naeini et al., 1988) or quantitative (Sneider and Frank, 1996; Frank, 1995; Nyberg and Stutte, 2004), to derive non-constant thresholds to take the time-varying uncertainties and disturbances into account. This paper instead proposes an adaptive statistical residual evaluation method, which exploits quantitative a-priori knowledge in the form of data.

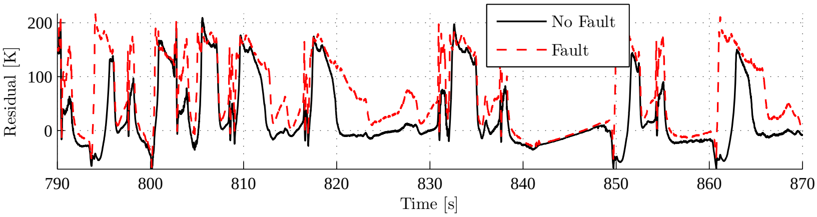


Figure 1: A residual for fault detection in the gas-flow system of a heavy-duty truck diesel engine in the no-fault (solid) and fault (dashed) cases.

The main contribution is to base the residual evaluation on an explicit comparison of the probability distribution of the residual, estimated on-line using current data, with a no-fault residual distribution. The no-fault distribution is based on a set of a-priori known no-fault distributions and to handle changes in the operating mode of the system, and thus time-varying residual features, it is continuously adapted to the current operating mode of the system. The comparison is done in the framework of statistical hypothesis testing by application of the Generalized Likelihood Ratio (GLR). As a second contribution, a method is proposed for estimating the required set of no-fault residual distributions off-line from no-fault training data. Thus, using the method for distribution estimation, the overall residual evaluation method becomes fully data-driven and no assumptions regarding the properties of the probability distribution of the residual, nor the properties of the faults to be detected, are made.

The paper is organized as follows. Section 2 discusses and formalizes the problem setup and the residual evaluation problem is formulated in the framework of statistical hypothesis testing. In Section 3, the GLR is utilized to design a preliminary test statistic for the residual evaluation hypotheses, and the emerging likelihood maximization problems are considered. In Section 4, the preliminary test statistic is improved in terms of required computational effort, and a residual evaluation algorithm suitable for implementation in an online environment is given. Section 5 presents an off-line algorithm for learning no-fault residual distributions from no-fault training data. In Section 6 the proposed residual evaluation approach is applied to a residual for fault detection in the gas-flow system of a real Scania truck diesel engine. Finally, Section 7 concludes the paper. In order to improve readability, lengthy proofs of theorems and lemmas are collected in Appendix A.

2 Problem Formulation

The residual evaluation problem, as considered in this work, is formally stated in this section.

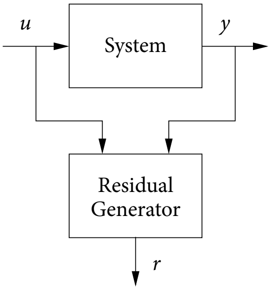


Figure 2: A system and a residual generator.

2.1 Prerequisites

A *residual*, r, is considered to be the output from a *residual generator*, taking measurements from a *system* as input. Typically, the measurements consists of the input u and output y, see Figure 2. The system is considered to be subject to faults, and the intention is to detect if any fault is present in the system by monitoring the behavior of the residual. Note that if a set of residuals sensitive to different faults is used, faults can also be isolated, see for example Blanke et al. (2006).

The system typically operates in a number of different *operating modes*, and normal operation usually involves several of these modes. For an example, consider a heavy-duty truck diesel engine, for which a residual is shown in Figure 1. Naturally, this system is designed to operate in a number of different operating modes typically characterized by engine torque, engine speed, ambient temperature, ambient pressure, etc.

The setup depicted in Figure 2 most often contains uncertainties in the form of measurement noise or, in the case of a model-based residual generator, modeling errors. Typically, the magnitudes and nature of the uncertainties are different for different operating modes of the system. For example, a sensor may be more or less sensitive to noise in different operating modes, and a model may be more accurate in one operating mode than another. Since the operating mode of the system varies in time, so does the magnitudes and nature of the uncertainties. This is the cause of the non-ideal residual behavior illustrated in Figure 1.

It is assumed that during on-line operation, the current operating mode of the system is unknown. In addition, it is also assumed that the probability that the system is in a specific mode is unknown. In this sense, the system can be considered to be subject to an unknown, i.e., unmeasurable, input signal, determining the current operating mode. Regarding in particular the first assumption, it is considered to be hard to quantify and measure all factors, internal and external, that determine the current operating mode of a system. Furthermore, these factors may be different for different individuals of the system, or may change over time. However, even if its is possible to determine a set of measured signals that determines the operating mode, all signals may not be available for the residual evaluation scheme due to for example fault decoupling principles, or architectural constraints in the control system software. In addition, even if all signals are available, they may as well be subject to faults. The second assumption is mainly motivated by the fact that the operation of a system differs between different individuals of the same system, and may change over time or due to external unmeasurable factors.

2.2 Probabilistic Framework

To handle the uncertain environment described above, a probabilistic framework is adopted. Let the discrete random variable R with range $\mathcal{X} = \{x_1, x_2, \ldots, x_M\}$, represent the discretized and sampled value of the residual, and let r denote a particular outcome of R.

For a given specific operating mode i of the system, the probability that $R = r$ is assumed to be characterized by the probability mass function (pmf)

$$p\left(r|\theta_i\right) = \Pr\left(R = r|\theta_i\right) = \theta_{ij}, \quad \text{if } r = x_j, \tag{1}$$

for $j = 1, \ldots, M$. The pmf (1) is fully parametrized by $\theta_i = \left(\theta_{i1}, \theta_{i2}, \ldots, \theta_{iM}\right)$, where the θ_{ij} are required to fulfill

$$\begin{aligned} &\theta_{ij} \geq 0, \quad j = 1, 2, \ldots, M \\ &\sum_{j=1}^{M} \theta_{ij} = 1. \end{aligned} \tag{2}$$

Under the assumption that there is in total K operating modes, the probability that $R = r$ can be characterized by the K-component mixture distribution given by the pmf

$$p\left(r|\alpha, \theta\right) = \sum_{i=1}^{K} \alpha_i \, p\left(r|\theta_i\right) \tag{3}$$

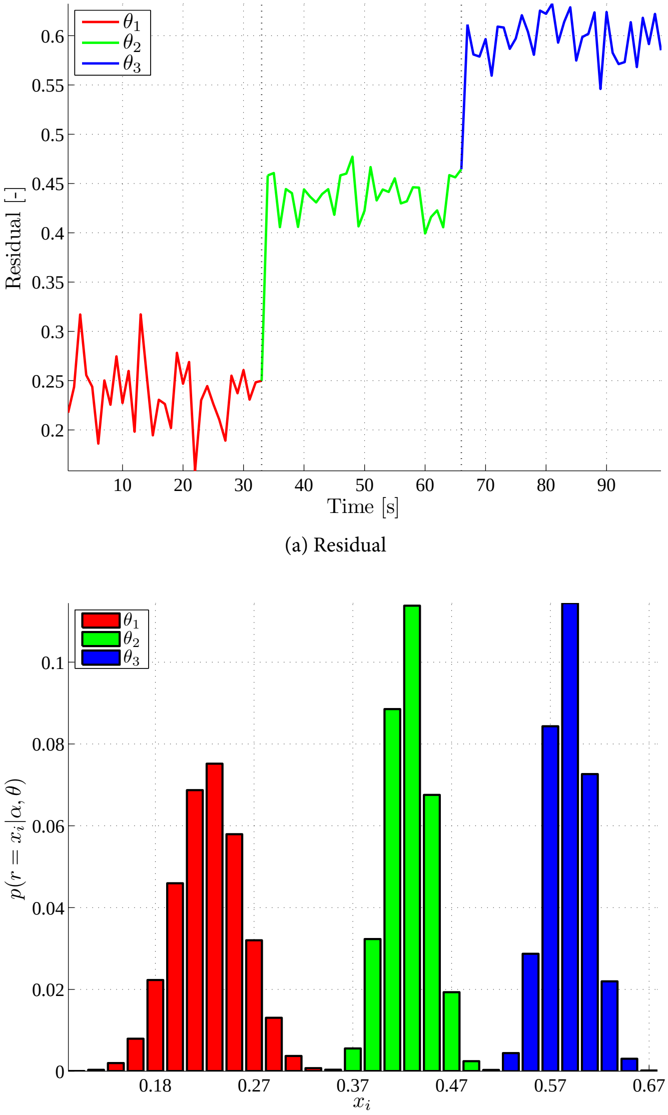
with $\alpha = \left(\alpha_1, \alpha_2, \ldots, \alpha_K\right)$ and

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_K \end{pmatrix} = \begin{pmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1M} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{K1} & \theta_{K2} & \cdots & \theta_{KM} \end{pmatrix}, \tag{4}$$

where α_i, $i = 1, 2, \ldots, K$, are referred to as mixture weights required to fulfill

$$\begin{aligned} &\alpha_i \geq 0, \quad i = 1, 2, \ldots, K, \\ &\sum_{i=1}^{K} \alpha_i = 1. \end{aligned} \tag{5}$$

In the context of this work, the mixture weight α_i specifies the probability that the system is in mode i. As said in Section 2.1, the probability that the system is in a specified operating mode is considered to be unknown. Consequently, α_i, $i = 1, 2, \ldots, K$, are assumed to be unknown and will in the following be considered as nuisance parameters.



(a) Residual

(b) Distribution of the Residual

Figure 3: Example of a sample from a mixture distribution in the form (3) with 3 components θ_1, θ_2, and θ_3, and mixture weights $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$.

Figure 3a shows a set of residual samples with underlying distributions described by the pmf (3), with 3 components θ_1, θ_2, and θ_3, shown in Figure 3b, and mixture weights $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$.

Note that the probabilistic model (3) can be used to describe the distribution of the residual for both the no-fault and faulty system.

In the context of residual evaluation, it is assumed that the distribution of the residual is known in the no-fault case. Let θ^{NF} denote the no-fault distribution parameter, where the i-th row θ_i^{NF} describes the distribution of the residual in operating mode i of the no-fault system. Section 5 describes how the required parameters θ_i^{NF} can be learned from no-fault training data, without the need of any detailed a-priori knowledge of the system. For a different approach, utilizing expert knowledge regarding the system, see Svärd et al. (2011).

Typically, the distribution of the residual is different for all K operating modes of the no-fault system, which implies that the matrix θ^{NF} has full row rank. For the model (3) to make sense it is required that $M > K$, since otherwise θ^{NF} can be used to describe any residual distribution, including ones originating from faulty cases.

2.3 Residual Evaluation in a Hypothesis Testing Framework

Consider now a set $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$ of sampled residual values. Given θ^{NF} and $\mathcal{R}$, the residual evaluation problem is, in the context of this work, to determine if the probability distribution of the residual samples in $\mathcal{R}$ can be characterized by the pmf (3) with $\theta = \theta^{\mathrm{NF}}$ for some $\alpha \in \Upsilon$, where

$$\Upsilon = \left\{ \alpha \in \mathbb{R}^K \;:\; \alpha_i \geq 0,\; \sum_{i=1}^{K} \alpha_i = 1 \right\}, \tag{6}$$

denotes the space of α as specified by (5).

The residual evaluation problem as described above can be formulated by means of the hypotheses

$$\begin{aligned} H_0: &\quad \theta = \theta^{\mathrm{NF}},\ \alpha \in \Upsilon \\ H_1: &\quad \theta \neq \theta^{\mathrm{NF}},\ \alpha \in \Upsilon \end{aligned} \tag{7}$$

where the null hypothesis H_0 corresponds to the no-fault case, i.e., when no fault is present in the system, and the alternative hypothesis H_1 to the faulty case, i.e., when one or several faults are present in the system. Next section deals with the problem of designing a test statistic for the hypotheses (7).

3 GLR Test Statistic

A standard approach when encountering composite hypotheses, is to utilize the Generalized Likelihood Ratio (GLR), see, e.g., Casella and Berger (2001); Basseville and

Nikiforov (1993). For testing hypothesis H_0 versus H_1 in (7), the GLR is

$$\Lambda\left(\mathcal{R}\right) = \frac{\max\limits_{\alpha \in \Upsilon} \mathcal{L}\left(\alpha, \theta^{\mathrm{NF}} | \mathcal{R}\right)}{\max\limits_{\alpha \in \Upsilon,\ \theta \in \Theta} \mathcal{L}\left(\alpha, \theta | \mathcal{R}\right)}, \tag{8}$$

where $\mathcal{L}\left(\theta, \alpha | \mathcal{R}\right)$ is the *likelihood function* of α and θ, given the set $\mathcal{R}$ of residual samples, and

$$\Theta = \left\{ \theta \in \mathbb{R}^{K \times M} \ : \ \theta_{ij} \geq 0, \ \sum_{j=1}^{M} \theta_{ij} = 1 \right\}, \tag{9}$$

denotes the space of the distribution parameter θ as specified by (2). The GLR test statistic becomes

$$\lambda\left(\mathcal{R}\right) = -2 \log \Lambda\left(\mathcal{R}\right), \tag{10}$$

and the hypothesis H_0 is rejected in favor of hypothesis H_1 if $\lambda\left(\mathcal{R}\right) > J$, where J is a constant threshold.

In order to employ the GLR test statistic $\lambda\left(\mathcal{R}\right)$, the maximization problems in the denominator and numerator of the GLR (8) must be solved. Before considering these maximization problems, the objective function, i.e., the likelihood function $\mathcal{L}\left(\theta, \alpha | \mathcal{R}\right)$, will be studied in some more detail.

3.1 The Likelihood Function

The likelihood function of the parameters θ and α given the set $\mathcal{R}$ of residual samples is given by

$$\mathcal{L}\left(\alpha, \theta | \mathcal{R}\right) = p\left(\mathcal{R} | \alpha, \theta\right), \tag{11}$$

where $p\left(\mathcal{R} | \theta, \alpha\right)$ is the joint pmf for the residual samples in $\mathcal{R}$. In the general case, the expression for the joint pmf is cumbersome to deal with. To make subsequent derivations tractable, or even possible, it is necessary to pose the following assumption.

Assumption 1. *Samples from (3) are independent and identically distributed (iid).*

Note that Assumption 1 not may be valid in the general case, since residuals often are obtained as output from dynamic systems and thereby exhibit Markovian properties. It can however often be fulfilled in practice by sampling the residual at a sufficiently low rate. In addition, residuals based on innovation filters (Gustafsson, 2000), e.g., the Kalman Filter, fulfills the assumption. The residual evaluation approach developed in this paper has also been shown to be applicable in practical settings, for example in the application example presented in Section 6.

By using Assumption 1, the joint pmf can be written as

$$p\left(\mathcal{R} | \alpha, \theta\right) = \prod_{r_k \in \mathcal{R}} p\left(r_k | \alpha, \theta\right), \tag{12}$$

where $p(\cdot|\alpha,\theta)$ is given by (3). By using (12), the likelihood (11) takes the form $\mathcal{L}(\alpha,\theta|\mathcal{R}) = \prod_{r_k\in\mathcal{R}} p(r_k|\alpha,\theta)$.

Next, let c_j denote how many of the samples in $\mathcal{R}$ that have value x_j, i.e.,

$$c_j = \left|\left\{r_k \in \mathcal{R} : r_k = x_j,\ x_j \in \mathcal{X}\right\}\right|, \quad j = 1,2,\ldots,M. \tag{13}$$

By definition, it holds that $\sum_{j=1}^{M} c_j = N$.

It is worth noting that the quantities $c_1, c_2, \ldots, c_M$ can be obtained from a regular histogram, with M bins, calculated from $\mathcal{R}$.

By using (12), (3), (13), and (1), the likelihood function (11) reduces to

$$\begin{aligned}\mathcal{L}(\alpha,\theta|\mathcal{R}) &= \prod_{r_k\in\mathcal{R}} p(r_k|\alpha,\theta)\\ &= \prod_{r_k\in\mathcal{R}} \sum_{i=1}^{K} \alpha_i\, p(r_k|\theta_i)\\ &= \prod_{j=1}^{M}\left(\sum_{i=1}^{K}\alpha_i\, p\left(x_j|\theta_i\right)\right)^{c_j}\\ &= \prod_{j=1}^{M}\left(\sum_{i=1}^{K}\alpha_i\,\theta_{ij}\right)^{c_j}.\end{aligned} \tag{14}$$

To simplify the calculations, the log-likelihood function

$$\begin{aligned}l(\alpha,\theta|\mathcal{R}) &= \log\left[\mathcal{L}(\alpha,\theta|\mathcal{R})\right]\\ &= \log\left[\prod_{j=1}^{M}\left(\sum_{i=1}^{K}\alpha_i\,\theta_{ij}\right)^{c_j}\right]\\ &= \sum_{j=1}^{M} c_j \log\left[\sum_{i=1}^{K}\alpha_i\,\theta_{ij}\right],\end{aligned} \tag{15}$$

will be used instead of (14).

Before proceeding, the following is assumed without loss of generality regarding $c_1, c_2, \ldots, c_M$, as specified by (13).

Assumption 2. $c_j > 0, \quad j = 1,2,\ldots,M.$

To see that Assumption 2 can be done without loss of generality, assume that $c_k = 0$, i.e., that there are no samples in $\mathcal{R}$ with value x_k. Then the corresponding factor in (14) is $\left(\sum_{i=1}^{K}\alpha_i\,\theta_{ij}\right)^0 \equiv 1$, or equivalently the corresponding term in (15) is $0\cdot\log\left[\sum_{i=1}^{K}\alpha_i\,\theta_{ij}\right] \equiv 0$, independent of α_j and θ_{ij}. Thus, this term, or factor in the case of the likelihood, can be neglected and the log-likelihood function (15) instead written as

$$l(\alpha,\theta|\mathcal{R}) = \sum_{j\in\{1,2,\ldots,M\}\setminus\{k\}} c_j \log\left[\sum_{i=1}^{K}\alpha_i\,\theta_{ij}\right].$$

3.2 Likelihood Maximizations

This section is devoted to explore in detail how to solve the two maximization problems in the GLR (8). Both problems correspond to finding parameter values that maximize the likelihood function (14), given the residual samples in $\mathcal{R}$, i.e., finding Maximum Likelihood Estimators (MLE's).

Denominator MLE Problem

Consider first the MLE problem

$$\max_{\alpha\in\Upsilon,\ \theta\in\Theta} \quad \mathcal{L}\left(\alpha,\theta|\mathcal{R}\right), \tag{16}$$

in the denominator of (8). Under Assumption 1, and by using the log-likelihood function (15) as well as the structure of the parameter spaces (9) and (6), the MLE problem (16) can be equivalently stated as

$$\begin{aligned} \max_{\alpha\in\mathbb{R}^K,\ \theta\in\mathbb{R}^{K\times M}} \quad & \sum_{j=1}^{M} c_j \log\left[\sum_{i=1}^{K}\alpha_i\,\theta_{ij}\right] \\ \text{subject to} \quad & \alpha_i \geq 0, \quad i=1,2,\ldots,K, \\ & \theta_{ij}\geq 0, \quad i=1,2,\ldots,K, j=1,2,\ldots,M, \\ & \sum_{i=1}^{K}\alpha_i = 1, \\ & \sum_{j=1}^{M}\theta_{ij}=1, \quad i=1,2,\ldots,K, \end{aligned} \tag{17}$$

which is a general non-linear constrained maximization problem.

It turns out that (17), and equivalently the MLE problem (16), can be solved explicitly. The key step in obtaining the expression for an explicit solution to (16) is given by the following lemma.

Lemma 1. *Let $c_1, c_2, \ldots, c_M$ fulfill Assumption 2. Then,*

$$\phi^\star = \left(\phi_1^\star, \phi_2^\star, \ldots, \phi_M^\star\right) \tag{18}$$

where

$$\phi_j^\star = \frac{c_j}{N}, \quad j=1,2,\ldots,M, \tag{19}$$

and $N=\sum_{j=1}^{M} c_j$, is the global solution to the maximization problem

$$\max_{\phi\in\mathbb{R}^M} \quad \prod_{j=1}^{M}\phi_j^{c_j} \tag{20a}$$

$$\text{subject to} \quad \phi_j\geq 0, \quad j=1,2,\ldots,M \tag{20b}$$

$$\sum_{j=1}^{M}\phi_j = 1. \tag{20c}$$

Proof. First note that by (20b) and Assumption 2 it holds that $\phi_j \geq 0$ and $c_j > 0$ for $j = 1, 2, \ldots, M$. Furthermore, by definition of c_j in (13), it also noted that $\sum_{j=1}^{M} c_j = N$. Consider now the weighted arithmetical and geometrical averages of the quantities $\frac{\phi_j}{c_j} \geq 0$ with weights $c_j > 0$ for $j = 1, 2, \ldots, M$. According to the inequality of weighted arithmetic and geometric means, see, e.g., Hardy et al. (1934), it then holds that

$$\frac{1}{N}\left(\sum_{j=1}^{M} \frac{\phi_j}{c_j} \cdot c_j\right) \geq \sqrt[N]{\prod_{j=1}^{M}\left(\frac{\phi_j}{c_j}\right)^{c_j}}, \tag{21}$$

with equality if and only if $\frac{\phi_1}{c_1} = \frac{\phi_2}{c_2} = \cdots = \frac{\phi_M}{c_M}$. For the left hand side of (21), it holds that $\frac{1}{N}\left(\sum_{j=1}^{M} \frac{\phi_j}{c_j} \cdot c_j\right) = \frac{1}{N}\sum_{j=1}^{M} \phi_j = \frac{1}{N}$ due to (20c). Exploiting this fact and re-writing the right hand side of (21) as $\sqrt[N]{\prod_{j=1}^{M}\left(\frac{\phi_j}{c_j}\right)^{c_j}} = \sqrt[N]{\frac{\prod_{j=1}^{M} \phi_j^{c_j}}{\prod_{j=1}^{M} c_j^{c_j}}}$, the inequality (21) can be equivalently stated as

$$\prod_{j=1}^{M} \phi_j^{c_j} \leq \frac{1}{N^N}\prod_{j=1}^{M} c_j^{c_j} = \prod_{j=1}^{M}\left(\frac{c_j}{N}\right)^{c_j}. \tag{22}$$

Now assume that equality holds in (21), and let $C = \frac{\phi_1}{c_1} = \frac{\phi_2}{c_2} = \cdots = \frac{\phi_M}{c_M}$. Under (20c), it then holds that $1 = \sum_{j=1}^{M} \phi_j = \sum_{j=1}^{M} C \cdot c_j = C\sum_{j=1}^{M} c_j = C \cdot N$ which is equivalent to that $C = \frac{1}{N}$. Hence, for the objective function $\prod_{j=1}^{M} \phi_j^{c_j}$ in (20a) it holds that $\prod_{j=1}^{M} \phi_j^{c_j} \leq \prod_{j=1}^{M}\left(\frac{c_j}{N}\right)^{c_j}$ under (20b), with equality under (20c) if and only if $\frac{\phi_j}{c_j} = \frac{1}{N} \Leftrightarrow \phi_j = \frac{c_j}{N}$, $j = 1, 2, \ldots, M$. This completes the proof. □

Note that since $\log[\cdot]$ is a strictly increasing function, Lemma 1 is also applicable to the problem of maximizing the function $\log\prod_{j=1}^{M} \phi_j^{c_j} = \sum_{j=1}^{M} c_j \log\phi$ subject to the conditions (20b) and (20c).

By using Lemma 1, a condition for a solution to the maximization problem (17), and thereby the MLE problem (16), can be obtained.

Theorem 1. *Let $\mathcal{R}$ be a set of residual samples, define $c_1, c_2, \ldots, c_M$ according to (13), and let Assumptions 1 and 2 be valid. Then, any $\alpha^\star \in \Upsilon$ and $\theta^\star \in \Theta$ such that*

$$\sum_{i=1}^{K} \alpha_i^\star\, \theta_{ij}^\star = \frac{c_j}{N}, \quad j = 1, 2, \ldots, M, \tag{23}$$

is a solution to the MLE problem (16).

Proof. Assumption 1 implies that the joint distribution of $\mathcal{R}$ is given by (12). With c_j defined according to (13), the likelihood (11) can be written as (14) and by exploiting the structure of the parameter spaces (6) and (9), it trivially follows that the MLE problem (16) can be equivalently reformulated as the maximization problem (17). From Lemma 1, and the fact that $\log[\cdot]$ is a strictly increasing function, it follows that any $\alpha^\star \in \Upsilon$ and $\theta^\star \in \Theta$ that satisfies $\frac{c_j}{N} = \sum_{i=1}^K \alpha_i^\star\, \theta_{ij}^\star$, $j = 1, 2, \ldots, M$, is a solution to the maximization problem

$$\begin{aligned} \max_{\alpha\in\mathbb{R}^K,\ \theta\in\mathbb{R}^{K\times M}} \quad & \sum_{j=1}^{M} c_j \log\left[\sum_{i=1}^{K} \alpha_i\, \theta_{ij}\right] \\ \text{subject to} \quad & \sum_{i=1}^{K} \alpha_i\, \theta_{ij} \geq 0, \quad j = 1, 2, \ldots, M, \\ & \sum_{j=1}^{M}\sum_{i=1}^{K} \alpha_i\, \theta_{ij} = 1. \end{aligned} \tag{24}$$

Now note that (24) has the same objective function as (17) and that the feasible set of (17) is contained in the feasible set of (24), since $\alpha_i \geq 0$ and $\theta_{ij} \geq 0$ implies $\sum_{j=1}^M \sum_{i=1}^K \alpha_i\, \theta_{ij} \geq 0$ and $\sum_{i=1}^K \alpha_i = 1$ and $\sum_{j=1}^M \theta_{ij} = 1$ implies that $\sum_{j=1}^M \sum_{i=1}^K \alpha_i\, \theta_{ij} = \sum_{i=1}^K \alpha_i \sum_{j=1}^M \theta_{ij} = \sum_{i=1}^K \alpha_i \cdot 1 = 1$, since $\theta \in \Theta$. Clearly, $(\alpha^\star, \theta^\star)$ is contained in the feasible set of problem (17) and it follows that $(\alpha^\star, \theta^\star)$ is a solution also to (17). It now remains to show that $(\alpha^\star, \theta^\star)$ is a global solution to (17). Since $\log[\cdot]$ is a non-decreasing concave function, and $\sum_{i=1}^K \alpha_i\, \theta_{ij}$ is a linear function, it holds that $\log\left[\sum_{i=1}^K \alpha_i\, \theta_{ij}\right]$ is a concave function. Therefore, the objective function in (17) is a convex sum of concave functions, since $c_j > 0$ due to Assumption 2, and hence a concave function. Since all constraints in (17) are linear, it follows that (17) is a concave optimization problem. Thus, the solution $(\alpha^\star, \theta^\star)$ is a global maximizer to (17) and hence a solution to the MLE problem (16). □

Numerator MLE Problem

Consider now the MLE problem

$$\max_{\alpha\in\Upsilon} \quad \mathcal{L}\left(\alpha, \theta^{\mathrm{NF}} | \mathcal{R}\right), \tag{25}$$

in the numerator of the GLR (8).

Note that (25) and (16) differs by that θ is fixed to θ^{NF} in (25).

With the notion of Section 2, the parameter θ^{NF} characterizes the set of distributions of the no-fault residual for all operating modes of the system. In this sense, the MLE problem (25) corresponds to finding a no-fault distribution that is most likely to fit the residual samples in $\mathcal{R}$.

By again using Assumption 1, the log-likelihood function (15), and exploiting the structure of the space (6) of the parameter α, the MLE problem (25) can be equivalently stated as the non-linear constrained maximization problem

$$\begin{aligned} \max_{\alpha\in\mathbb{R}^K} \quad & \sum_{j=1}^{M} c_j \log\left[\sum_{i=1}^{K} \alpha_i\, \theta_{ij}^{\mathrm{NF}}\right] \\ \text{subject to} \quad & \alpha_i \geq 0, \quad i = 1, 2, \ldots, K, \\ & \sum_{i=1}^{K} \alpha_i = 1. \end{aligned} \tag{26}$$

In the general case, it is unfortunately not possible to find an explicit expression for a solution to the maximization problem (26), or equivalently the MLE problem (25), as was the case with the MLE problem (16). There are however several efficient numerical approaches, see, e.g., Nocedal and Wright (2006).

By using similar arguments as in the proof of Theorem 1, it can be shown that also (26) is a concave maximization problem. The concavity property facilitates the numerical solving since it implies that if a local maximum can be found, then it is also a global maximum.

4 Online Residual Evaluation Algorithm

Typically, residual evaluation is to be done in an online environment subject to real-time constraints, i.e., computational times in order of micro- or milliseconds with strict deadlines. Unfortunately, it is in general not feasible to solve the non-linear MLE problem (25), or equivalently (26), under such conditions. In this section, a relaxed version of the MLE problem (25) is proposed. The relaxed problem requires less computational effort and results in a residual evaluation test that under certain conditions performs better than the residual evaluation test based on the original MLE problem.

4.1 Relaxed Problem

In light of Theorem 1, and since the problems (26) and (17) exhibit significant similarities, an intuitive solution to problem (26) is to, if possible, choose $\alpha \in \Upsilon$ so that

$$\sum_{i=1}^{K} \alpha_i \, \theta_{ij}^{\mathrm{NF}} = \frac{c_j}{N}, \qquad j = 1, 2, \ldots, M. \tag{27}$$

However, since $K < M$, see Section 2.2, (27) corresponds to an overdetermined set of equations which in general has no solution. Motivated by this discussion, it makes sense to chose α so that each $\sum_{i=1}^{K} \alpha_i \, \theta_{ij}^{\mathrm{NF}}$ is as close as possible to $\frac{c_j}{N}$ for $j = 1, 2, \ldots, M$. Thus, the following relaxation of the problem (26) is considered

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^K} \quad & \frac{1}{2} \| \sum_{i=1}^{K} \alpha_i \theta_i^{\mathrm{NF}} - \phi^\star \|_2^2 \\ \text{subject to} \quad & \alpha_i \geq 0, \quad i = 1, 2, \ldots, K, \\ & \sum_{i=1}^{K} \alpha_i = 1, \end{aligned} \tag{28}$$

where $\phi^\star$ is defined by (18).

The relaxed problem (28) is equivalent to a linear least squares problem with equality and non-negative constraints. Solving (28) therefore typically requires less computational effort than solving the original general non-linear maximization problem (26). Solving of (28) will be further discussed in Section 4.3.

In order to compare the fault detection properties of the residual evaluation tests based on the relaxed problem (28) and the original MLE problem (26), the following result is given.

Lemma 2. *Let $c_1, c_2, \dots, c_M$ fulfill Assumption 2, let $\theta^{NF} \in \Theta$, and*

$$\Phi^{NF} = \left\{ \phi \;:\; \phi = \sum_{i=1}^{K} \alpha_i \; \theta_i^{NF}, \; \forall \alpha \in \Upsilon \right\}. \tag{29}$$

Further, let $\phi^\star \in \Phi^{NF}$, and let α^O and α^R be solutions to the original problem (26) and relaxed problem (28), respectively. Then, it holds that

$$\sum_{i=1}^{K} \alpha_i^O \; \theta_i^{NF} = \sum_{i=1}^{K} \alpha_i^R \; \theta_i^{NF} = \phi^\star. \tag{30}$$

Proof. First note that $\phi^\star \in \Phi^{\mathrm{NF}}$ is equivalent to that the set

$$\Upsilon^{\mathrm{NF}} = \left\{ \alpha \in \Upsilon \;:\; \phi^\star = \sum_{i=1}^{K} \alpha_i \; \theta_i^{\mathrm{NF}} \right\}, \tag{31}$$

is non-empty. Assume that $\Upsilon^{\mathrm{NF}} \neq \varnothing$ and consider first the optimization problem (26). Since $\Upsilon^{\mathrm{NF}} \neq \varnothing$, it follows from Lemma 1, and the fact that $\log[\cdot]$ is an increasing function, that any optimal solution to (26) is contained in Υ^{NF}. In particular, this holds for α^{O} and thus $\phi^\star = \sum_{i=1}^{K} \alpha_i^{\mathrm{O}} \; \theta_i^{\mathrm{NF}}$. Consider next the optimization problem (28). Again $\Upsilon^{\mathrm{NF}} \neq \varnothing$ implies that any optimal solution to (28), in particular α^{O}, is contained in Υ^{NF}. Hence, $\phi^\star = \sum_{i=1}^{K} \alpha_i^{\mathrm{R}} \; \theta_i^{\mathrm{NF}}$ and the proof is complete. □

Consider the hypotheses in (7) and the GLR test statistic $\lambda(\mathcal{R})$ defined by (10) and (8). Define the test statistic

$$\lambda_{\mathrm{R}}(\mathcal{R}) = -2 \log \frac{\mathcal{L}\left(\alpha^{\mathrm{R}}, \theta^{\mathrm{NF}} | \mathcal{R}\right)}{\mathcal{L}\left(\alpha^\star, \theta^\star | \mathcal{R}\right)}, \tag{32}$$

where $(\alpha^\star, \theta^\star)$ is a solution to the original MLE problem (16) as present in (8), but where α^{R} is a solution to the relaxed numerator MLE problem (28).

The power of the residual evaluation test $\lambda(\mathcal{R}) > J$ can be quantified by the *power function* (Casella and Berger, 2001)

$$\beta_\lambda(\alpha, \theta) = \Pr\left(\text{reject } H_0 | \alpha, \theta\right) = \Pr\left(\lambda(\mathcal{R}) > J | \alpha, \theta\right), \tag{33}$$

where J is a fixed threshold. If $\alpha \in \Upsilon$ and $\theta = \theta^{\mathrm{NF}}$ in (33), i.e., under H_0, the power function gives the probability of false detection, or *Type I error*. Otherwise, the power function gives the probability of detection for fixed α and θ, or equivalently the probability of missed detection or *Type II error*, by $1 - \beta_\lambda(\alpha, \theta)$.

Consider now the power function

$$\beta_{\lambda_{\mathrm{R}}}(\alpha, \theta) = \Pr\left(\lambda_{\mathrm{R}}(\mathcal{R}) > J | \alpha, \theta\right), \tag{34}$$

for the residual evaluation test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$, based on the relaxed problem (28). The relation between the power functions (33) and (34) is given by the following result.

Theorem 2. *It holds that*

$$\beta_{\lambda_R}(\alpha, \theta) \geq \beta_\lambda(\alpha, \theta). \tag{35}$$

Proof. It is first noted that according to Theorem 1, it holds that $\phi_j^\star = \frac{c_j}{N}$, $j = 1, 2, \ldots, M$, and thus Lemma 2 is applicable. According to Lemma 2, it holds that $\phi^\star = \sum_{i=1}^K \alpha_i^{\mathrm{O}}\ \theta_i^{\mathrm{NF}} = \sum_{i=1}^K \alpha_i^{\mathrm{R}}\ \theta_i^{\mathrm{NF}}$ if $\phi^\star \in \Phi^{\mathrm{NF}}$. This implies, due to (14), that $\mathcal{L}\left(\alpha^{\mathrm{O}}, \theta^{\mathrm{NF}}|\mathcal{R}\right) = \mathcal{L}\left(\alpha^{\mathrm{R}}, \theta^{\mathrm{NF}}|\mathcal{R}\right)$ if $\phi^\star \in \Phi^{\mathrm{NF}}$. Due to the concavity property of the likelihood function $\mathcal{L}(\alpha, \theta|\mathcal{R})$, and the fact that α^{O} is a solution to the MLE problem (25), it follows that

$$\mathcal{L}\left(\alpha^{\mathrm{R}}, \theta^{\mathrm{NF}}|\mathcal{R}\right) \leq \mathcal{L}\left(\alpha^{\mathrm{O}}, \theta^{\mathrm{NF}}|\mathcal{R}\right),$$

with equality if $\phi^\star \in \Phi^{\mathrm{NF}}$. Thus, it holds that

$$\frac{\mathcal{L}\left(\alpha^{\mathrm{R}}, \theta^{\mathrm{NF}}|\mathcal{R}\right)}{\mathcal{L}\left(\alpha^\star, \theta^\star|\mathcal{R}\right)} \leq \frac{\mathcal{L}\left(\alpha^{\mathrm{O}}, \theta^{\mathrm{NF}}|\mathcal{R}\right)}{\mathcal{L}\left(\alpha^\star, \theta^\star|\mathcal{R}\right)}, \tag{36}$$

and equivalent that $\lambda_{\mathrm{R}}(\mathcal{R}) \geq \lambda(\mathcal{R})$, due to (32) and (10), again with equality if $\phi^\star \in \Phi^{\mathrm{NF}}$. The claim (35) then follows directly by definitions (33) and (34). □

The implication of Theorem 2 is that the residual evaluation test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$, based on the relaxed problem (28), gives greater or equal probability for detection than the test $\lambda(\mathcal{R}) > J$, based on the original problem (26). Or equivalently, that the Type II error, i.e., the probability for missed detection, for the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$ always is smaller than, or equal to, the Type II error for the test $\lambda(\mathcal{R}) > J$.

In general, unfortunately, the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$ gives larger probability for false detection, i.e., Type I error, than the test $\lambda(\mathcal{R}) > J$. This is a direct consequence of Theorem 2. However, asymptotically the condition $\phi^\star \in \Phi^{\mathrm{NF}}$ holds under hypothesis H_0, i.e., in the no-fault case, which implies that also the probabilities for false detection becomes equal for the two tests. This fact is formalized in the following result.

Theorem 3. *Let N denote the number of residual samples in $\mathcal{R}$, and let H_0 in (7) be valid. Then, it holds that*

$$\lim_{N\to\infty} \beta_{\lambda_R}(\alpha, \theta) - \beta_\lambda(\alpha, \theta) = 0. \tag{37}$$

Proof. Define $\phi = \sum_{i=1}^K \alpha_i\ \theta_i$ and note that from (7), it can be deduced that $\phi \in \Phi^{\mathrm{NF}}$ is equivalent to that $\alpha \in \Upsilon$ and $\theta = \theta^{\mathrm{NF}}$, i.e., that H_0 in (7) is valid. Thus, by assumption, it holds that $\phi \in \Phi^{\mathrm{NF}}$. Consider now $\phi^\star$ and note that due to the invariance property (Casella and Berger, 2001) of maximum likelihood estimates it holds that if $(\alpha^\star, \theta^\star)$ are the MLE of (α, θ), which indeed is true by assumption, then $\phi^\star = \sum_{i=1}^K \alpha_i^\star\ \theta_i^\star$ is the MLE of ϕ. Lemma 5 (found in Appendix A) then implies that

$$\lim_{N\to\infty} \Pr\left(|\phi^\star - \phi| \geq \varepsilon\right) = 0,$$

for all $\varepsilon > 0$ and $\phi \in \Phi'$, with Φ' defined by (70). Since it holds that $\phi \in \Phi^{\mathrm{NF}}$ by assumption, it therefore holds that $\phi^\star \in \Phi^{\mathrm{NF}}$ when $N \to \infty$. Since $\phi^\star \in \Phi^{\mathrm{NF}}$ holds, (36) holds with equality which is equivalent to that $\lambda_{\mathrm{R}}(\mathcal{R}) = \lambda(\mathcal{R})$. By (33) and (34) this is equivalent to $\beta_{\lambda_{\mathrm{R}}}(\alpha, \theta) = \beta_\lambda(\alpha, \theta)$, and thus (37) holds. □

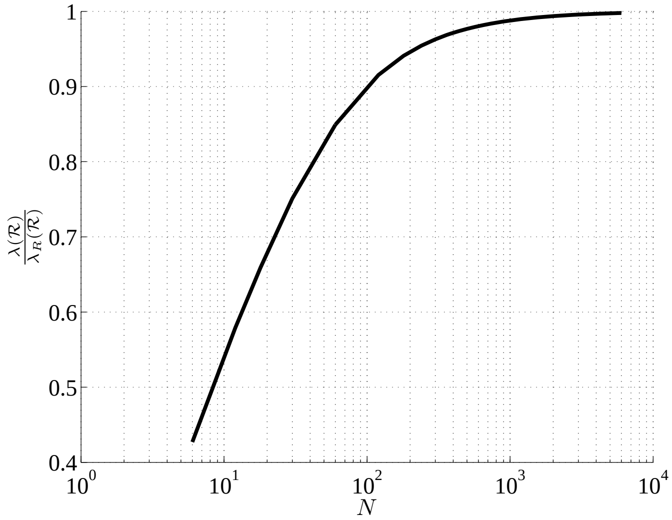


Figure 4: Comparison of test quantities $\lambda_{\mathrm{R}}(\mathcal{R})$ and $\lambda(\mathcal{R})$ under hypothesis H_0, by means of the quantity $\frac{\lambda(\mathcal{R})}{\lambda_{\mathrm{R}}(\mathcal{R})}$, for different values of the size N of the residual sample $\mathcal{R}$.

Theorem 3 is empirically illustrated in Figure 4, which shows a comparison of the test statistics $\lambda_{\mathrm{R}}(\mathcal{R})$ and $\lambda(\mathcal{R})$, under hypothesis H_0, as the size N of the residual sample $\mathcal{R}$ grows. In this particular case, the parameters $M = 80$ and $K = 25$ was used. The comparison is done by means of the quantity $\frac{\lambda(\mathcal{R})}{\lambda_{\mathrm{R}}(\mathcal{R})}$, and Figure 4 shows the average of 10,000 Monte Carlo simulations using synthetic data. It is clear that the test quantities $\lambda_{\mathrm{R}}(\mathcal{R})$ and $\lambda(\mathcal{R})$ are almost equal when N is large, in this case for $N > 1000$. Since both test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$ and $\lambda(\mathcal{R}) > J$ are based on the same threshold J, the situation in Figure 4 implies that the power functions $\beta_{\lambda_{\mathrm{R}}}(\alpha, \theta)$ and $\beta_{\lambda}(\alpha, \theta)$ are almost identical under H_0 when N is sufficiently large.

To summarize, Theorem 2 implies that the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$, based on the relaxed problem (28), will result in greater or equal probability for detection than the GLR test $\lambda(\mathcal{R}) > J$, based on the original MLE problem (26). Moreover, according to Theorem 3, if N is sufficiently large, then also the probabilities for false detection will be almost equal for two tests.

In an application where computational effort is crucial, and when implementation matters limit usage of a “sufficiently large” N, a switch from the original MLE problem (26) to the relaxed problem (28), means trading probability of false detection against computational feasibility.

4.2 Residual Evaluation Algorithm

The proposed method for residual evaluation is summarized as an algorithm below. Input to the algorithm is a set of residual samples $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$, a no-fault residual distribution parameter θ^{NF}, and a detection threshold J. Output is a decision whether to reject hypothesis H_0 in (7) or not, i.e., whether a fault is present in the system or not.

Step 1: Compute $c_1, c_2, \ldots, c_M$ according to (13).

Step 2: Obtain α^{R} by solving (28).

Step 3: Obtain (32) by computing

$$\lambda_{\mathrm{R}} = -2 \log \frac{\prod_{j=1}^{M} \left(\sum_{i=1}^{K} \alpha_i^{\mathrm{R}} \, \theta_{ij}^{\mathrm{NF}} \right)^{c_j}}{\prod_{j=1}^{M} \left(\frac{c_j}{N} \right)^{c_j}}. \tag{38}$$

Step 5: Reject H_0 if $\lambda_{\mathrm{R}} > J$.

Note that for use with sequential residual data, the samples in $\mathcal{R}$ may be collected by using a sliding window, i.e., at sampling instant t the set of residual samples

$$\mathcal{R}_t = \left\{ r_{t-N+1}, r_{t-N+2}, \ldots, r_t \right\},$$

is used, where r_t denotes the residual sample collected at instant t.

Parameter Choices

The parameters involved in the residual evaluation are the number N of residual samples in $\mathcal{R}$, the detection threshold J, and the no-fault distribution parameter θ^{NF}. The first two parameters, N and J, are discussed below. The parameter θ^{NF} is the topic of Section 5.

According to Theorem 3, the relaxation (28) of the MLE problem (25) is justified in terms of the probability for false detection if N is sufficiently large. The actual meaning of "sufficiently large" is application dependent and must be evaluated from case to case. This can for example be done by comparing the test quantities $\lambda_{\mathrm{R}}(\mathcal{R})$ and $\lambda(\mathcal{R})$, under hypothesis H_0, for different values of N in the same manner as in Figure 4.

In general, given that N is large enough to justify the relaxation, the choice of N is a trade-off between detection performance and complexity. A large N will give the test statistic smoothed, low-pass, characteristics. This makes it possible to detect small changes in the residual, but on the other hand a large N may increase the detection time. Computational and memory aspects will be discussed in Section 4.3.

The choice of detection threshold J is a trade-off between detection time, and test power, in terms of probability of false detection and probability of missed detection. The higher the threshold, the longer the detection time, the lower the probability of false detections but the higher the probability of missed detection. The actual selection of threshold may be aided by the fact that the test statistic based on the GLR, ideally, is Chi-squared distributed (Willsky and Jones, 1976).

4.3 Implementation Issues and Computational Complexity

Typically, the residual evaluation algorithm outlined in Section 4.2 is implemented and executed in real-time in an online environment. This poses strict restrictions on the computational complexity of the algorithm, in terms of requirements of computing time and storage.

The main potential computational pitfalls of the algorithm are related to Step 1 and Step 2, i.e., computing the bin counts $c_1, c_2, \dots, c_M$ according to (13) and solving the equality and inequality constrained linear least square problem (28). These two issues will now be considered, starting with the former.

Computing the Bin Counts

Computing the bin counts $c_1, c_2, \dots, c_M$ given a set of residual samples $\mathcal{R}$ corresponds to, for each $x_j \in \mathcal{X}$, counting how many samples in $\mathcal{R}$ that takes value x_j, where $\mathcal{X}$ denotes the range space of the residual, see Section 2.2.

As said in Section 3.1, the quantities $c_1, c_2, \dots, c_M$ can be obtained from a regular histogram, with M bins, computed from $\mathcal{R}$ and the computational complexity for this problem depends on the parameters M and N, i.e., the number of bins in the histogram, and the number of samples in $\mathcal{R}$, respectively.

The number of required computations for computing a regular histogram with M bins from a set of N samples, is $M \times N$ and grows linearly with both M and N. Considering the memory requirements, the N residual values and the M bin counts need to be stored, and these requirements also grow linearly. The conclusion is that if only there is enough memory available, the histogram calculations, and thus the computation of the bin counts, can easily be performed in real-time in an online environment.

Solving the Constrained Linear Least Square Problem

A variety of numerical methods have been developed for solving linear least square problems with inequality and equality constraints, see, e.g., Haskell and Hanson (1981); Lawson and Hanson (1974); Bjorck (1996); Zhu and Rong Li (2007). Most methods are based on convex optimization (Boyd and Vandenberghe, 2004), where primal-dual methods, including interior point methods (Wright, 1997) and the active set method (Bjorck, 1996), are of particular interest.

Convex optimization problems can be efficiently solved (Boyd and Vandenberghe, 2004; Wright, 1997), using for example algorithms with worst-case polynomial complexity (Nesterov and Nemirovskii, 1994). State-of-the-art algorithms often exploits code-generation, where solvers are customized to a specific problem class. One such example is CVXGEN (Mattingley and Boyd, 2012), which enables real-time, i.e., solving time scales in microseconds or milliseconds with strict deadlines, solving of modest-sized quadratic optimization problems (Mattingley and Boyd, 2010).

The absolute requirements on memory and computation time for solving the linear least square problem (28) by using any of the above methods, depends on the dimension and structure of the $K \times M$ matrix θ^{NF}, where K denotes the number of considered operating modes of the system, and M the number of bins in the above mentioned histogram. The most crucial parameter of these two is K, which in this sense should be kept as low as possible. Implications of the value of this quantity, in the context of residual evaluation performance, is further discussed in Section 5.

It is worth noting that the complexity of the problem (28) does not depend on the number N of residual samples in $\mathcal{R}$. This is favorable since it is only justified to consider the relaxed problem (28) instead of the MLE problem (25) if N is sufficiently large, see Section 4.1.

5 Learning No-Fault Distribution Parameters

In previous sections, it was assumed that the distribution of the residual was known, by means of the parameter θ^{NF}, for K operating modes of the no-fault system. Given a set of residual samples, the problem was to determine if the set of samples originated from the distribution (3) with θ fixed to θ^{NF}. In the context of this section, however, the parameter θ^{NF}, as well as K, are considered to be unknown and the task at hand is to *learn*, i.e., estimate, these using a large set of residual samples, denoted training data.

It is important to stress that the learning is done in an off-line environment with less restrictions on computational complexity, while the actual residual evaluation, as considered in Section 4, typically, is performed online.

5.1 Problem Characterization

With the notion of Section 2, the distribution parameter θ_i^{NF}, i.e., the i-th row of the $K \times M$ matrix which constitutes the parameter θ^{NF}, characterizes the distribution of the no-fault residual when the considered system is in operating mode i. Thus, the value of K determines the number of considered operating modes of the system and θ^{NF} the set of no-fault residual distributions associated with these operating modes.

Note that if training data partitioned according to operating mode is given, the parameter θ^{NF} can be directly obtained by means of Lemma 1. Specifically, the distribution parameter θ_i^{NF} is obtained by computing a normalized histogram with M bins for the part of the data corresponding to operating mode i.

If the total number of operating modes of the system is known, this knowledge can be exploited and K set accordingly. In general, however, K is unknown and must be learned from the training data. The importance and meaning of the value of K is discussed next.

A large K allows for a complete description of the set of no-fault residual distributions as specified by θ^{NF}, which may be desirable. However, if K is too large, the set of distributions may become too large in the sense that any distribution in the form (3) can be characterized by θ^{NF}. This may reduce the fault detection performance of the residual evaluation test developed in previous sections, since almost any set of residual samples will be considered as generated from a no-fault system, which means no alarm, even if there is a fault present. In addition, a large K results in a θ^{NF} of large dimension, which affects the computational issues addressed in Section 4.3. So in this sense, K should be kept as low as possible. A too small K, on the other hand, may give an insufficient description of the set of all no-fault distributions. This typically also leads to decreased fault detection performance, either in the form of missed detections or false alarms, depending on the strategy used when setting the alarm threshold.

In conclusion, the choice of K and θ^{NF} is a trade-off between fault detection performance and computational effort. However, in order to take the fault detection performance into account, training data from a set of representative fault cases is needed. In the context of this work it is however assumed that only no-fault training data is available due to a number of reasons. First of all, the amount of available no-fault data is typically substantially larger than the available amount of fault data, since faults are rare. To create fault data, one alternative is to inject faults in the real system. This is however considered to be expensive, both in terms of time and money, since it typically require hardware modifications and active usage of the system. Another alternative is to create fault data by simulation. To give realistic results, this on the other hand requires models capable of describing the faulty system, which in turn require detailed knowledge regarding the behavior of the faulty system and possibly also its environment. This kind of information is seldom available for real applications.

Motivated by this discussion, fault detection performance will not be explicitly considered in the learning of K and θ^{NF}. Instead, the learning problem will be formulated as a trade-off between the ability of K and θ^{NF} to characterize the set of all no-fault residual distributions, i.e., model fit, and computational effort. The main motivation for this choice is that a good characterization of the no-fault case will hopefully make it possible to detect deviations from the no-fault case, meaning good fault detection performance. The resulting fault detection performance is however empirically studied in Section 6.

5.2 Problem Formulation

Consider a set $\mathcal{D} = (r_1, r_2, \ldots, r_{N_{\mathcal{D}}})$ of $N_{\mathcal{D}}$ residual samples ordered according to time. The residual samples in $\mathcal{D}$ will now be split into residual sample sets

$$\mathcal{R}_k = \left\{ r_{(k-1)n+1}, r_{(k-1)n+2}, \ldots, r_{k-n} \right\}, \tag{39}$$

containing n consecutive residual samples from $\mathcal{D}$. To this end, let $n < N_{\mathcal{D}}$, and define

$$\mathcal{T} = (\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{N_{\mathcal{T}}}), \tag{40}$$

where $N_{\mathcal{T}} = \left\lfloor \frac{N_{\mathcal{D}}}{n} \right\rfloor$, and $\mathcal{R}_k$ is given by (39) for $k = 1, 2, \ldots, N_{\mathcal{T}}$. The collection $\mathcal{T}$ of residual sample sets $\mathcal{R}_k$, will henceforth be referred to as the training data.

In the following, it is assumed that each $\mathcal{R}_k \in \mathcal{T}$ contains residual samples from only one operating mode. In practice, this can be achieved by choosing n such that the time it takes to collect a set of n residual samples is shorter than the time the system spends in one operating mode, as well as longer than the transition time between any two operating modes.

Formalization of Learning Problem

Let $V(\mathcal{T}, \theta)$ denote a metric that quantifies model fit, i.e., how well the set of distributions characterized by a given parameter θ is able to describe a data set $\mathcal{T}$ in the form (40).

A general approach for enabling a trade-off between goodness of model fit and model complexity when identifying parameters in a model is to combine the model fit metric, in the present case $V(\mathcal{T}, \theta)$, with some metric that reflects the model complexity (Ljung, 1999; Söderström and Stoica, 1989).

In the context of this work, required computational effort rather than model complexity is of direct interest. As said in Section 4.2, the required computational effort for the residual evaluation algorithm presented in Section 4.2 is strongly dependent on the dimension $K \times M$ of θ^{NF}, and in particular the value of K. Since the larger the value of K, the higher the computational requirements, a function $C(K)$ that increases with K is suitable for quantification of the computational effort. Typically, the actual choice of $C(K)$ is implementation dependent. In general, there are many options, see, e.g., Ljung (1999); Söderström and Stoica (1989). One alternative is to exploit the information criteria due to Akaike (Akaike, 1974).

Given $V(\mathcal{T}, \theta)$ and $C(K)$, the learning problem as stated in Section 5.1 can be formulated as the problem

$$\left(K^\star, \theta^{\mathrm{NF}}\right) = \arg \max_{K,\ \theta \in \Theta^{(K)}} \left(V(\theta, \mathcal{T}) - C(K)\right), \tag{41}$$

where the notation $\Theta^{(K)}$ for the space defined in (9) is introduced to stress the dependency between the space and K. The topic of the remaining of this section is to derive a suitable metric $V(\mathcal{T}, \theta)$ for quantification of model fit.

Quantification of Model Fit

To be able to exploit the developments in previous sections, a likelihood-based framework is adopted for quantification of model fit, and an expression for the (log)-likelihood $l(\theta|\mathcal{T})$ is sought.

To this end, recall from Section 3.1 that under Assumption 1, the joint pmf for a set of residual samples, in this case $\mathcal{R}_k \in \mathcal{T}$, can be written as

$$\begin{aligned} p(\mathcal{R}_k|\alpha_k, \theta) &= \prod_{r_p \in \mathcal{R}_k} p\left(r_p|\alpha_k, \theta\right) \\ &= \prod_{r_p \in \mathcal{R}_k} \sum_{i=1}^{K} \alpha_{ki}\, p\left(r_p|\theta_i\right), \end{aligned} \tag{42}$$

where $\alpha_k = (\alpha_{k1}, \alpha_{k2}, \ldots, \alpha_{kK})$ contains the mixture weights associated with $\mathcal{R}_k$. By the construction in (39), it holds that $\mathcal{R}_i \cap \mathcal{R}_j = \varnothing$ for any pair $\mathcal{R}_i \in \mathcal{T}$ and $\mathcal{R}_j \in \mathcal{T}$, where $i \neq j$. This, Assumption 1 and (42), implies that

$$\begin{aligned} p(\mathcal{T}|\theta, \alpha_1, \alpha_2, \ldots, \alpha_{N_\mathcal{T}}) &= \prod_{k=1}^{N_\mathcal{T}} p(\mathcal{R}_k|\alpha_k, \theta) \\ &= \prod_{k=1}^{N_\mathcal{T}} \prod_{r_p \in \mathcal{R}_k} \sum_{i=1}^{K} \alpha_{ki}\, p\left(r_p|\theta_i\right) \end{aligned} \tag{43}$$

Let c_{kj} denote the total number of residual samples in $\mathcal{R}_k$ that takes value x_j, c.f. (13). The log-likelihood of θ and α_k, $k = 1, 2, \ldots, N_{\mathcal{T}}$, given $\mathcal{T}$, can then be written as

$$\begin{aligned} l\left(\theta, \alpha_1, \alpha_2, \ldots, \alpha_{N_{\mathcal{T}}} | \mathcal{T}\right) &= \log p\left(\mathcal{T} | \theta, \alpha_1, \alpha_2, \ldots, \alpha_{N_{\mathcal{T}}}\right) \\ &= \log \prod_{k=1}^{N_{\mathcal{T}}} \prod_{r_p \in \mathcal{R}_k} \sum_{i=1}^{K} \alpha_{ki}\; p\left(r_p | \theta_i\right) \\ &= \log \prod_{k=1}^{N_{\mathcal{T}}} \prod_{j=1}^{M} \left(\sum_{i=1}^{K} \alpha_{ki}\; p\left(x_j | \theta_i\right) \right)^{c_{kj}} \\ &= \log \prod_{k=1}^{N_{\mathcal{T}}} \prod_{j=1}^{M} \left(\sum_{i=1}^{K} \alpha_{ki}\; \theta_{ij} \right)^{c_{kj}} \\ &= \sum_{k=1}^{N_{\mathcal{T}}} \sum_{j=1}^{M} c_{kj} \log \left[\sum_{i=1}^{K} \alpha_{ki} \theta_{ij} \right] \end{aligned} \tag{44}$$

The likelihood function $l\left(\theta, \alpha_1, \alpha_2, \ldots, \alpha_{N_{\mathcal{T}}} | \mathcal{T}\right)$ in (44) contains both the parameter of interest θ, and the nuisance parameters α_k, $k = 1, 2, \ldots, N_{\mathcal{T}}$. Thus, the nuisance parameters α_k must be *eliminated* from (44). There are mainly two standard approaches (Basu, 1977) for doing this. The first approach is to fix a prior probability distribution for the nuisance parameters, compute the posterior, and then integrate out the nuisance parameter from the posterior to arrive at the posterior marginal distribution of the parameter of interest, see for example Berger et al. (1999). The second approach is to replace the nuisance parameters in the original likelihood function with their conditional maximum likelihood estimates. The resulting function, which not indeed is a pure likelihood function anymore, is referred to as a *profile likelihood* or *maximized likelihood*, see, e.g., Patefield (1977); Murphy and Vaart (2000).

In the context of this section, the mixture weight α_{ki} specifies the probability that the samples in $\mathcal{R}_k$ were collected when operating mode i was present. As said Section 2.1, this probability, and all other probabilities related to the nuisance parameters α_k are assumed to be unknown, which complicates the usage of the first approach mentioned above.

Motivated by this discussion, the second approach is adopted for elimination of α_k, $k = 1, 2, \ldots, N_{\mathcal{T}}$, from (44). The resulting profile likelihood of θ, given $\mathcal{T}$, takes the form

$$\begin{aligned} \hat{l}\left(\theta | \mathcal{T}\right) &= \max_{\alpha_1, \alpha_2, \ldots, \alpha_{N_{\mathcal{T}}} \in \Upsilon} l\left(\theta, \alpha_1, \alpha_2, \ldots, \alpha_{N_{\mathcal{T}}} | \mathcal{T}\right) \\ &= \max_{\alpha_1, \alpha_2, \ldots, \alpha_{N_{\mathcal{T}}} \in \Upsilon} \sum_{k=1}^{N_{\mathcal{T}}} \sum_{j=1}^{M} c_{kj} \log \left[\sum_{i=1}^{K} \alpha_{ki} \theta_{ij} \right] \\ &= \sum_{k=1}^{N_{\mathcal{T}}} \max_{\alpha_k \in \Upsilon} \sum_{j=1}^{M} c_{kj} \log \left[\sum_{i=1}^{K} \alpha_{ki} \theta_{ij} \right] \end{aligned} \tag{45}$$

Under the assumption that each $\mathcal{R}_k \in \mathcal{T}$ contains residual samples from only one operating mode, it holds that each α_k, $k = 1, 2, \ldots, N_{\mathcal{T}}$, contains one and only one non-zero element, equal to one. In this case,

$$\max_{\alpha_k \in \Upsilon} \sum_{j=1}^{M} c_{kj} \log \left[\sum_{i=1}^{K} \alpha_{ki} \theta_{ij} \right] = \max_{i \in \{1,2,\dots,K\}} \sum_{j=1}^{M} c_{kj} \log \theta_{ij},$$

and thus the (profile) likelihood (45) of θ, given $\mathcal{T}$, can be written as

$$\hat{l}\left(\theta|\mathcal{T}\right) = \sum_{k=1}^{N_{\mathcal{T}}} \max_{i \in \{1,2,\dots,K\}} \sum_{j=1}^{M} c_{kj} \log \theta_{ij}. \tag{46}$$

Motivated by these developments, the metric

$$V\left(\mathcal{T},\theta\right) = \hat{l}\left(\theta|\mathcal{T}\right) = \sum_{k=1}^{N_{\mathcal{T}}} \max_{i \in \{1,2,\dots,K\}} \sum_{j=1}^{M} c_{kj} \log \theta_{ij}, \tag{47}$$

will be used to quantify how well the set of distributions characterized by a given parameter θ is able to describe a data set $\mathcal{T}$.

5.3 Learning Algorithm

Consider now the learning problem as formulated in (41). According to Section 2.2 and the fact that it is required that $K < M$, the feasible set of $K^\star$ is bounded. Moreover, the quantity $C\left(K\right)$ is not dependent on θ. Thus, given that the problem $\max_{\theta \in \Theta^{(K)}} V\left(\mathcal{T},\theta\right)$ can be solved for a given K, the learning problem (41) can be solved by an exhaustive search over the feasible set of $K^\star$.

The key step when searching for $K^\star$ and θ^{NF} that solve (41), is therefore to find, for a given K, a $\theta^{(K)}$ that satisfies

$$\theta^{(K)} = \arg\max_{\theta \in \Theta^{(K)}} V\left(\mathcal{T},\theta\right). \tag{48}$$

This is the topic of the remainder of this section.

Method Outline

The basic idea of the proposed approach for finding $\theta^{(K)}$ is to first calculate a distribution parameter $\theta_k \in \Theta^{(1)}$ for each $\mathcal{R}_k \in \mathcal{T}$ by exploiting Theorem 1 and form the set

$$\Psi = \left(\theta_1, \theta_2, \dots, \theta_{N_{\mathcal{T}}}\right), \tag{49}$$

where

$$\theta_k = \arg\max_{\theta \in \Theta^{(1)}} l\left(\theta|\mathcal{R}_k\right), \tag{50}$$

for $k = 1,2,\dots,N_{\mathcal{T}}$. Then group the distribution parameters in Ψ into K clusters $\mathrm{P}_1, \mathrm{P}_2, \dots, \mathrm{P}_K$ according to their similarity, and finally calculate the distribution parameter $\theta_i^\star$, which constitute the i-th row of $\theta^{(K)}$, from the distribution parameters in cluster P_i.

For an illustration of the approach, consider the residual sample sets

$$\mathcal{T} = (\mathcal{R}_1, \mathcal{R}_2, \dots \mathcal{R}_9),$$

defined according to Figure 5a. Note that the sets $\mathcal{R}_k$ in Figure 5a have been generated in an ideal way for the purpose of illustration. The set of corresponding distribution parameters $\Psi = (\theta_1, \theta_2, \dots \theta_9)$ is illustrated in Figure 5b, and the sought clusters are $\mathrm{P}_1 = \{\theta_1, \theta_2, \theta_3\}$, $\mathrm{P}_2 = \{\theta_4, \theta_5, \theta_6\}$, and $\mathrm{P}_3 = \{\theta_7, \theta_8, \theta_9\}$. The resulting distribution parameters $\theta_1^\star$, $\theta_2^\star$, and $\theta_3^\star$, calculated as the mean of the parameters in the clusters P_1, P_2, and P_3, respectively, are shown in Figure 6. Note the similarity between Figure 6 and Figure 3b, where the latter in fact shows the true distribution parameters.

Algorithm

The general algorithm for finding a solution to (48) is given below. The input to the algorithm is a set of residual samples $\mathcal{D}$ and constants n and K. The output is a distribution parameter $\theta^{(K)}$.

In the algorithm, $D\left(p\left(r|\theta_k\right) \| p\left(r|\theta_i^\star\right)\right)$ denotes the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) between the probability distributions characterized by $p\left(r|\theta_k\right)$ and $p\left(r|\theta_i^\star\right)$. The KL-divergence is one way to quantify the similarity of probability distributions and is properly defined in Section 5.4.

Step 1: Let $\mathcal{T}$ be defined by (40).

Step 2: Let Ψ be defined by (49).

Step 3: Partition Ψ into $\mathrm{P}^\star = (\mathrm{P}_1, \mathrm{P}_2, \dots, \mathrm{P}_K)$ such that

$$\mathrm{P}^\star = \arg\min_{\mathrm{P}} \sum_{i=1}^{K} \sum_{\theta_k \in \mathrm{P}_i} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta_i^\star\right)\right), \tag{51}$$

where

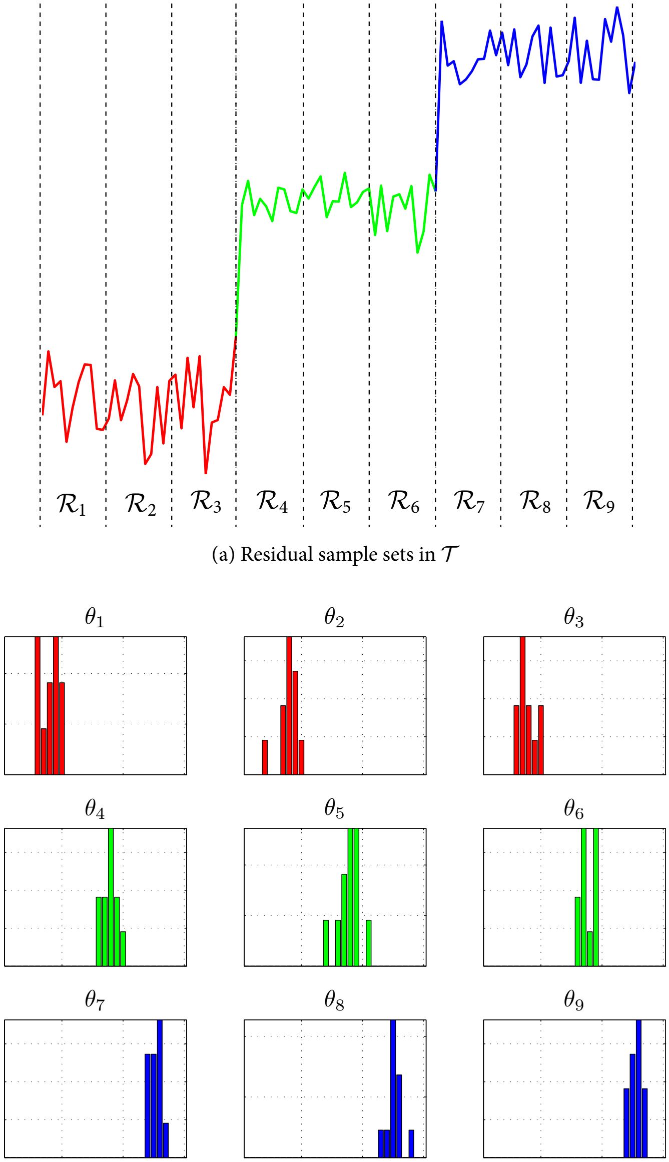
$$\theta_i^\star = \frac{1}{|\mathrm{P}_i|} \sum_{\theta_k \in \mathrm{P}_i} \theta_k, \quad i = 1, 2, \dots, K. \tag{52}$$

Step 4: Let

$$\theta^{(K)} = \begin{pmatrix} \theta_1^\star \\ \theta_2^\star \\ \vdots \\ \theta_K^\star \end{pmatrix} = \begin{pmatrix} \theta_{11}^\star & \theta_{12}^\star & \cdots & \theta_{1M}^\star \\ \theta_{21}^\star & \theta_{22}^\star & \cdots & \theta_{2M}^\star \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{K1}^\star & \theta_{K2}^\star & \cdots & \theta_{KM}^\star \end{pmatrix}. \tag{53}$$

The most crucial part of the above algorithm is Step 3, in which a particular partition of the set Ψ should be computed. This problem in fact corresponds to a hard K-means clustering problem (Bishop, 2006), for which efficient heuristic methods exists (Lloyd, 1982). Implementation issues are discussed in Section 5.5.

The justification of the algorithm, in terms of its ability to provide a solution to the problem (48), is given in next section.



(a) Residual sample sets in $\mathcal{T}$

(b) Distribution parameters in Ψ

Figure 5: Illustration of the proposed learning algorithm. Figure 5a shows the residual sample sets in $\mathcal{T}$ and Figure 5b the corresponding distribution parameters in Ψ.

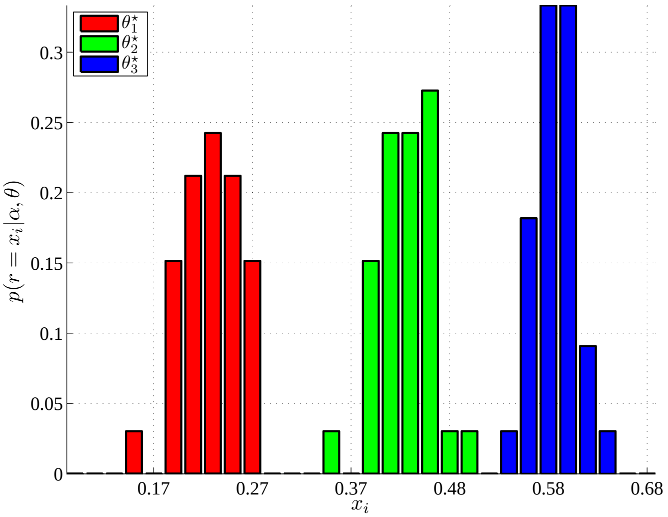


Figure 6: The distribution parameters learned from the training data in Figure 5a.

5.4 Justification of Learning Algorithm

This section contains technical developments necessary for proving that the algorithm defined by Steps 1-4 in Section 5.3 indeed gives a solution to the problem (48) as output. This is done in the following manner. First, a sufficient condition for a solution to the problem (48) is given. The condition is given in terms of properties of a partition of the set $\mathcal{T}$, computed in Step 1 of the algorithm. Next, the sufficient condition is transformed into a condition on a partition of the set Ψ, defined in Step 2. Finally, it is verified that the partition of Ψ computed by means of K-means clustering in Step 3 satisfies this condition.

A sufficient condition for a solution to the problem (48) is given below.

Theorem 4. *Let $\mathcal{D}$ be a set of $N_\mathcal{D}$ residual samples fulfilling Assumption 1, let $n < N_\mathcal{D}$, and let $\mathcal{T}$ be defined by (40). For a given positive integer K, if $\mathrm{T} = (\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_K)$ is a partition of $\mathcal{T}$ such that for each block $\mathrm{T}_i \in \mathrm{T}$ and for each element $\mathcal{R}_k \in \mathrm{T}_i$, it holds that*

$$l\left(\theta_i^\star|\mathcal{R}_k\right) \geq l\left(\theta_p^\star|\mathcal{R}_k\right), \quad p = 1, 2, \ldots, K, \tag{54}$$

where

$$\theta_i^\star = \arg\max_{\theta \in \Theta^{(1)}} \sum_{\mathcal{R}_k \in \mathrm{T}_i} l\left(\theta|\mathcal{R}_k\right), \quad i = 1, 2, \ldots, K, \tag{55}$$

then

$$V\left(\mathcal{T}, \theta^{(K)}\right) = \max_{\theta \in \Theta^{(K)}} V\left(\mathcal{T}, \theta\right), \tag{56}$$

with $V\left(\mathcal{T}, \theta\right)$ and $\theta^{(K)}$ defined by (47) and (53), respectively.

Proof. It is first noted that by Assumption 1, the joint pmf for the samples in $\mathcal{R}_k \in \mathcal{T}$ is given by (42), which is equivalent to (12). From (15), and the fact that $\theta \in \Theta^{(1)}$ due to (55), which implies that $K = 1$ and $\alpha_1 = 1$ in (15), it holds that

$$l(\theta|\mathcal{R}_k) = \sum_{j=1}^{M} c_{kj} \log \theta_j, \tag{57}$$

where c_{kj}, $j = 1, 2, \ldots, M$, denotes the total number of samples in $\mathcal{R}_k$ that takes value x_j. Given is that $\mathrm{T} = (\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_K)$ is a partition of $\mathcal{T}$, such that (54) is satisfied for each block $\mathrm{T}_i \in \mathrm{T}$ and for each element $\mathcal{R}_k \in \mathrm{T}_i$, with $\theta_i^\star$, $i = 1, 2, \ldots, K$, defined according to (55). From (54) and (57) it follows that for each $\mathrm{T}_i \in \mathrm{T}$ and for each $\mathcal{R}_k \in \mathrm{T}_i$, it holds that

$$\sum_{j=1}^{M} c_{kj} \log \theta_{ij}^\star \geq \sum_{j=1}^{M} c_{kj} \log \theta_{pj}^\star, \tag{58}$$

for $p = 1, 2, \ldots, K$. Due to (58) it holds that for each $\mathrm{T}_i \in \mathrm{T}$ and for each $\mathcal{R}_k \in \mathrm{T}_i$

$$\max_{p \in \{1,2,\ldots,K\}} \sum_{j=1}^{M} c_{kj} \log \theta_{pj}^\star = \sum_{j=1}^{M} c_{kj} \log \theta_{ij}^\star. \tag{59}$$

Due to (59) and the fact that $\mathrm{T} = (\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_K)$ is a partition of $\mathcal{T} = (\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{N_\mathcal{T}})$, it holds that

$$\begin{aligned} V(\mathcal{T}, \theta^\star) &= \sum_{k=1}^{N_\mathcal{T}} \max_{p \in \{1,2,\ldots,K\}} \sum_{j=1}^{M} c_{kj} \log \theta_{pj}^\star \\ &= \sum_{i=1}^{K} \sum_{\mathcal{R}_k \in \mathrm{T}_i} \max_{p \in \{1,2,\ldots,K\}} \sum_{j=1}^{M} c_{kj} \log \theta_{pj}^\star \\ &= \sum_{i=1}^{K} \sum_{\mathcal{R}_k \in \mathrm{T}_i} \sum_{j=1}^{M} c_{kj} \log \theta_{ij}^\star \end{aligned} \tag{60}$$

By definition (55), it holds that $\theta_i^\star \in \Theta^{(1)}$, $i = 1, 2, \ldots, K$, and therefore that $\theta^{(K)} \in \Theta^{(K)}$ with $\theta^{(K)}$ defined by (56). To show that (56) is satisfied, it is sufficient to show that $V(\mathcal{T}, \theta^\star)$ is a maximum value. Since $\theta_i^\star = (\theta_{i1}^\star, \theta_{i2}^\star, \ldots, \theta_{iM}^\star)$ only is present in the term

$$\sum_{\mathcal{R}_k \in \mathrm{T}_i} \sum_{j=1}^{M} c_{kj} \log \theta_{ij}^\star, \tag{61}$$

in (60), it follows that $V(\mathcal{T}, \theta^\star)$ as given by (60) is a maximum if (61) is a maximum, for each $i = 1, 2, \ldots, M$. It is now noted that, due to (57), (55) is equivalent to

$$\theta_i^\star = \arg\max_{\theta \in \Theta^{(1)}} \sum_{\mathcal{R}_k \in \mathrm{T}_i} \sum_{j=1}^{M} c_{kj} \log \theta_j, \quad i = 1, 2, \ldots, K,$$

which completes the proof. □

The implication of Theorem 4, is that the solving of (48) can be reduced to finding a partition $\mathrm{T} = (\mathrm{T}_1, \mathrm{T}_2, \dots, \mathrm{T}_K)$ of the set $\mathcal{T}$, defined according to (40), that fulfills (54). Next result, establishes a relation between the sought partition T of $\mathcal{T}$ and a partition P of the set Ψ computed in Step 2 of the algorithm.

To this end, KL-divergence needs to be properly defined. In general, for two distributions of a discrete random variable R with range $\mathcal{X}$ that are characterized by the pmf's $f_1(r)$ and $f_2(r)$, the KL-divergence between $f_1(r)$ and $f_2(r)$ is defined as

$$D\left(f_1(r) \| f_2(r)\right) = \sum_{x_k \in \mathcal{X}} f_1(x_k) \log \frac{f_1(x_k)}{f_2(x_k)}. \tag{62}$$

It follows that $D\left(f_1(r) \| f_2(r)\right) \geq 0$, with equality if and only if $f_1(r) \equiv f_2(r)$.

A transformation of the sufficient condition in Theorem 4 on a partition T of $\mathcal{T}$ to a partition P of the set Ψ is given by the following lemma.

Lemma 3. *Let* $\mathrm{P}_i \subseteq \Psi$, *let*

$$\mathrm{T}_i = \{\mathcal{R}_k \in \mathcal{T} \ : \ \theta_k \in \mathrm{P}_i\} \tag{63}$$

and let all residual samples in all $\mathcal{R}_k \in \mathrm{T}_i$ *fulfill Assumption 1. Then, for any* $\theta_p, \theta_q \in \Theta^{(1)}$ *and for each* $\mathcal{R}_k \in \mathrm{T}_i$ *it holds that*

$$l\left(\theta_p | \mathcal{R}_k\right) \geq l\left(\theta_q | \mathcal{R}_k\right), \tag{64}$$

if and only if for each $\theta_k \in \mathrm{P}_i$ *it holds that*

$$D\left(p\left(r|\theta_k\right) || p\left(r|\theta_q\right)\right) \leq D\left(p\left(r|\theta_k\right) || p\left(r|\theta_q\right)\right). \tag{65}$$

Moreover, it holds that

$$\arg\max_{\theta \in \Theta^{(1)}} \sum_{\mathcal{R}_k \in \mathrm{T}_i} l\left(\theta | \mathcal{R}_k\right) = \arg\min_{\theta \in \Theta^{(1)}} \sum_{\theta_k \in \mathrm{P}_i} D\left(p\left(r|\theta_k\right) || p\left(r|\theta\right)\right). \tag{66}$$

Proof. Given in Appendix A. □

The problem of finding a partition T of $\mathcal{T}$ fulfilling the sufficient condition in Theorem 4 can with aid of Lemma 3 be equivalently stated as the problem of finding a partition P of Ψ fulfilling the condition (65). Next result verifies that a partition of Ψ computed in Step 3 of the algorithm indeed satisfies (65).

Lemma 4. *Let* $\mathcal{D}$ *be a set of* $N_{\mathcal{D}}$ *residual samples fulfilling Assumption 1, let* $n < N_{\mathcal{D}}$, *let* $\mathcal{T}$ *be defined by (40), let* Ψ *be defined by (49), and let* K *be a positive integer. Further, let* $\mathrm{P}^\star = (\mathrm{P}_1, \mathrm{P}_2, \dots, \mathrm{P}_K)$ *be a partition of* Ψ *such that (51) holds and* $\theta_i^\star$, $i = 1, 2, \dots, K$, *satisfies (52). Then, it holds that*

$$\theta_i^\star = \arg\min_{\theta \in \Theta^{(1)}} \sum_{\theta_k \in \mathrm{P}_i} D\left(p\left(r|\theta_k\right) || p\left(r|\theta\right)\right), \tag{67}$$

for $i = 1, 2, \dots, K$. *Moreover, for each block* $\mathrm{P}_i \in \mathrm{P}^\star$ *and for each element* $\theta_k \in \mathrm{P}_i$ *it holds that*

$$D\left(p\left(r|\theta_k\right) || p\left(r|\theta_i^\star\right)\right) \leq D\left(p\left(r|\theta_k\right) || p\left(r|\theta_j^\star\right)\right), \tag{68}$$

for $j = 1, 2, \dots, K$.

Proof. Given in Appendix A. □

With help of Theorem 4, Lemma 3, and Lemma 4, it can be proved that the output from the algorithm in Section 5.3 indeed is a solution to the problem (48).

Theorem 5. *Let $\mathcal{D}$ be a set of $N_{\mathcal{D}}$ residual samples fulfilling Assumption 1, let $n < N_{\mathcal{D}}$, and let K be a positive integer. Further, let $\mathcal{D}$, n, and K, be input to the algorithm defined by Steps 1-4 in Section 5.3 and let $\theta^{(K)}$ be the output. Then, $\theta^{(K)}$ is a solution to (48).*

Proof. Due to Step 3 in the algorithm, it is clear that the partition $\mathrm{P}^\star = (\mathrm{P}_1, \mathrm{P}_2, \dots, \mathrm{P}_K)$ fulfills (51) and that $\theta_i^\star$, $i = 1, 2, \dots, M$, fulfills (52). Lemma 4 then implies that (68) holds for each block $\mathrm{P}_i \in \mathrm{P}^\star$ and for each element $\theta_k \in \mathrm{P}_i$, and that $\theta_i^\star$, $i = 1, 2, \dots, M$, fulfills (67). Now define $\mathrm{T} = (\mathrm{T}_1, \mathrm{T}_2, \dots, \mathrm{T}_K)$ with T_i according to (63) for $i = 1, 2, \dots, K$. Note that due to (49) and (63), it follows that there is block $\mathrm{T}_i \in \mathrm{T}$ and an element $\mathcal{R}_k \in \mathrm{T}_i$ for each element $\theta_k \in \mathrm{P}_i$ and for each block $\mathrm{P}_i \in \mathrm{P}^\star$, and vice versa. The fact that $\mathrm{P}^\star$ is a partition of Ψ, then implies that T is a partition of $\mathcal{T}$. Appliance of Lemma 3 to each block $\mathrm{P}_i \in \mathrm{P}^\star$ then asserts that the partition T satisfies $l\left(\theta_i^\star|\mathcal{R}_k\right) \geq l\left(\theta_j^\star|\mathcal{R}_k\right)$ for each block $\mathrm{T}_i \in \mathrm{T}$ and for each element $\mathcal{R}_k \in \mathrm{T}_i$, for all $j = 1, 2, \dots, K$. Further, since (67) holds for $\theta_i^\star$ and due to (66) in Lemma 3, it follows that $\theta_i^\star = \arg\max_{\theta \in \Theta^{(1)}} \sum_{\mathcal{R}_k \in \mathrm{T}_i} l\left(\theta|\mathcal{R}_k\right)$, $i = 1, 2, \dots, M$. The claim then follows directly from Theorem 4. □

5.5 Implementation Issues

As said in Section 5.3, the most crucial part of the learning algorithm is Step 3, i.e., to find a partition P of Ψ by means of hard K-means clustering (Bishop, 2006).

The complexity properties of the general K-means clustering problem depends on which similarity measure, in the present case the KL-divergence, that is used in (51). For instance, the problem is NP-hard (Aloise et al., 2009) when the (squared) Euclidean distance is used, but can be solved in a polynomial time if a variance-based measure is used (Inaba et al., 1994).

There are however a variety of heuristic algorithms available for solving the general clustering problem approximately. One widely used (Berkhin, 2002) and in practice often successful alternative, is the local search based K-Means algorithm (MacQueen, 1967; Lloyd, 1982), which also is referred to as Lloyd's algorithm. For the particular, and present, case when the KL-divergence is used as similarity measure, am approximate solution to the clustering problem can be computed with the K-means algorithm in polynomial time (Manthey and Röglin, 2009). For a general treatment of clustering problems with similarity measures based on Bregman divergences, including the KL-divergence, see Banerjee et al. (2005).

The K-means algorithm solves (51) by alternating two steps: i) given a set of distribution parameters, assign each $\theta_k \in \Psi$ to the most similar, in a KL-divergence sense, distribution parameter, ii) update the distribution parameters according to the new assignments. These two steps are iterated until no assignments change, which eventually will be the case after a finite number of iterations (Selim and Ismail, 1984; Bottou and Bengio, 1995).

As a remark, it is noted that the assignment and update steps in fact (Bishop, 2006) correspond to the Expectation and Maximization steps, respectively, in the EM-algorithm (Dempster et al., 1977). Thus, when the K-means algorithm is employed for solving the clustering problem in Step 3, the learning algorithm in a sense resembles the EM-algorithm.

It is also noted that in a practical implementation of the learning algorithm, the training data set $\mathcal{T}$ is preferably split into an estimation data set $\mathcal{E}$ and a validation data set $\mathcal{V}$, in order to avoid over-fitting, see, e.g., Ljung (1999). In this setting, the estimation data set $\mathcal{E}$ is used when solving (48) to obtain $\theta^{(K)}$, for a fixed K, and then the validation data set $\mathcal{V}$ is used to evaluate if the obtained solution $\theta^{(K)}$ and K satisfies (41).

Parameter Choices

The only parameter involved in the learning problem (41) is n, the number of residual samples used in each $\mathcal{R}_k$ when calculating the set $\mathcal{T}$ according to (40), which is done in Step 1 of the algorithm.

The choice of n is determined by the properties of the considered system. As said in Section 5.2, n should be chosen so that each $\mathcal{R}_k \in \mathcal{T}$ contains residual samples from only one operating mode of the system. In order to achieve this, n should be chosen so that the time it takes to collect a set of n residual samples is less than the average time that the system spends in one operating mode.

Before learning the parameter θ^{NF}, the quantization M of the residual, i.e., the size of the residual range space and thereby the resolution of the residual distribution (1), must be determined and the training data in $\mathcal{D}$ formated accordingly. Choosing M, in fact, corresponds to the well-studied, but nevertheless difficult, problem of choosing the number of bins in a regular histogram given a sample of data. Numerous approaches for solving this problem exist, see for example Davies et al. (2009) and references therein.

Regardless of the method used to solve the problem, the choice of M is a trade-off between accuracy and computational complexity, in terms of time and storage. A larger M results in a more accurate discretization of the residual and higher resolution of the probability distributions. On the other hand, a large M requires more memory and involves more computations. The choice of M is also related to the choice of n and N, since a small n, or N, together with a large M will result in an inadequate estimation of the distribution, i.e., a sparse histogram.

The resolution of the residual also affects the fault detection performance in the sense that if the resolution is high, small deviations of the residual can be perceived and thereby small faults can be detected. As a guideline, the resolution of the residual can be matched to the size of the smallest fault that should be possible to detect.

6 Application Example

The proposed residual evaluation approach has been applied to the problem of fault detection in the gas-flow system of a Scania 6 cylinder, 13 liter, truck diesel engine equipped with Exhaust Gas Recirculation (EGR), Variable Geometry Turbine (VGT), and intake throttle. The overall purpose of the study was to evaluate and demonstrate the proposed on-line residual evaluation algorithm, as well as the off-line algorithm for learning no-fault residual distributions, using measurement data. In addition, it is also illustrated how the fault detection performance of the residual evaluation test is influenced by different values of the involved parameters, in particular the size N of the residual sample set $\mathcal{R}$, and the number K of no-fault distribution parameters in θ^{NF}.

6.1 Automotive Gas-Flow Diagnosis

The automotive gas-flow system, or rather the truck diesel engine itself, is a complex system that operates in a variety of different operating modes characterized by for instance ambient pressure and temperature, engine torque, engine speed, etc. Fault diagnosis of the gas-flow system consists of detecting and isolating faults in sensors that measure pressure, temperature, and mass-flow, actuators that control the EGR, VGT and intake throttle, as well as faults related to, e.g., manifold leakages and clogged air filters. The main incentives for gas-flow diagnosis are fault management by means of fault tolerant control, On-Board Diagnosis (OBD) regulations, and repair and maintenance.

The model of the gas-flow system, which is described in Wahlström and Eriksson (2011), relies on both fundamental first principle physics and gray-box modeling. For diagnosis of the gas-flow system, a set of model-based residual generators were designed with the sequential residual generation method described in Svärd and Nyberg (2010). Naturally, the model does not describe all aspects of the system, leading to that all residuals exhibit properties similar to those illustrated in Figure 1.

The particular residual considered in this study is sensitive to 10 faults: 3 leakages, 6 sensor faults, and 1 actuator fault. The value of the residual is based on a comparison of two modeled values of the temperature before the cylinders.

6.2 Learning of No-Fault Distribution Parameters

The data set used for the learning contains measurements from parts of a test drive, including both city and high-way driving, from Södertälje to Arvidsjaur in Sweden. The data set contains in total 156,912 measurements sampled at a rate of 0.1 s, which corresponds to more than 4 hours of driving. The measurements in the data set were used as input to the considered residual generator and the residual samples used in the study were computed off-line. In order to minimize the risk of over-fitting the no-fault distribution parameters to the training data, the set of residual samples was divided into an estimation data set, $\mathcal{E}$, and a validation data set, $\mathcal{V}$, of equal size.

Parameter Values

The value of the parameter M, i.e., the quantization of the residual samples, was chosen to be $M = 80$. This makes it theoretically possible to detect faults that cause deviations of the residual of about 3 kelvin. For this application, this is a good trade-off between complexity, in terms of required memory and computational effort, and accuracy.

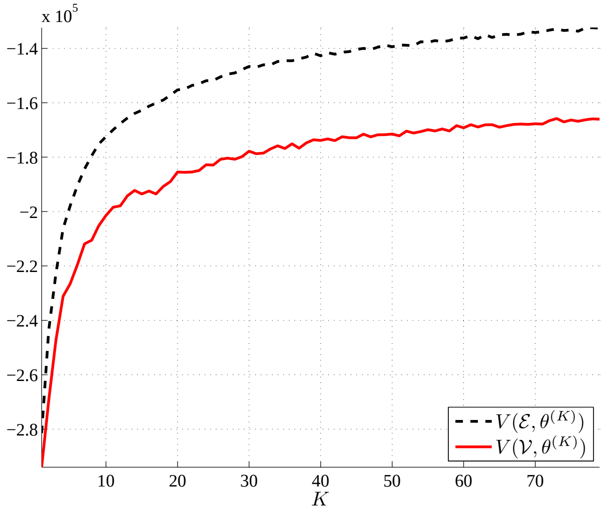


Figure 7: Evaluation of model fit metrics $V\left(\mathcal{E},\theta^{(K)}\right)$ (dashed, black) and $V\left(\mathcal{V},\theta^{(K)}\right)$ (solid, red) for different of values of K.

By a brief analysis of the residual samples, it seems that the minimum time that the gas-flow system spends in one operating mode is approximately 4 s. This can be seen in Figure 1, which in fact shows a subset of the residual samples used in this study. Since the sample rate is 0.1 s, the parameter n, which specifies the number of residual samples in each $\mathcal{R}_k$ in the set $\mathcal{T}$ calculated in Step 1 of the algorithm, should be chosen to satisfy $n < 40$, see Section 5.5. Based on this, the parameter was chosen to be $n = 32$.

Results

The algorithm for learning no-fault distribution parameters described in Section 5.3, was implemented in Matlab. To solve the involved clustering problem, the K-means algorithm (MacQueen, 1967; Lloyd, 1982) was employed. The algorithm was run with $K \in \{1, 2, \ldots, 79\}$.

Figure 7 shows the model fit metric (47) evaluated for the estimation data set $\mathcal{E}$ and validation data set $\mathcal{V}$, and with the parameters $\theta^{(K)}$, $K \in \{1, 2, \ldots, 79\}$, obtained as output from the algorithm. In Figure 7 it can first of all be seen that the quantitative behaviors of $V\left(\mathcal{E},\theta^{(K)}\right)$ and $V\left(\mathcal{V},\theta^{(K)}\right)$ are similar, but that $V\left(\mathcal{E},\theta^{(K)}\right)$ always is larger than $V\left(\mathcal{V},\theta^{(K)}\right)$. The latter seems natural since the data set $\mathcal{E}$ indeed was used as input to the learning algorithm. Second, it can also be noticed that the improvement in model fit as a function of K is larger for smaller K.

Based on the above observations, and with respect to the trade-off between model fit and required computational effort stated by (41), $K = 10$ was chosen. The 10 no-fault distribution parameters, i.e., the rows of $\theta^{(10)}$, are shown in Figure 8. Note that the characteristics of the learned distribution parameters are quite different, some are multi-

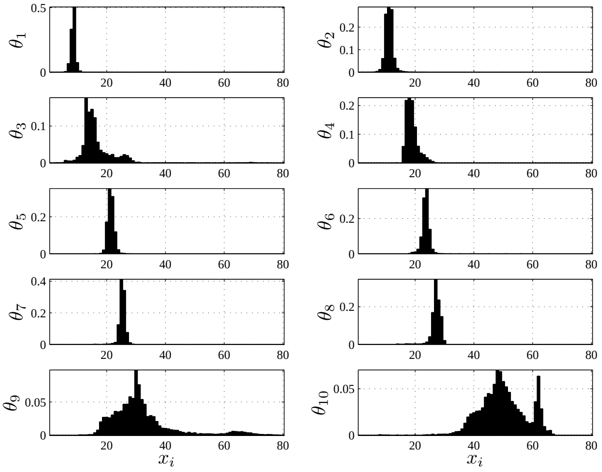


Figure 8: The no-fault distribution parameters contained in $\theta^{\mathrm{NF}} = \theta^{10}$.

modal and some have only one single mode. In addition, the distribution parameters are overlapping.

6.3 Evaluation Setup

The set of residual samples used in the evaluation is based on the validation data set $\mathcal{V}$, which contains in total 78,456 residual samples. Note that this data set is different than the estimation data set used to learn the no-fault distribution parameters as described above.

Considered Fault

The fault considered in the evaluation is a fault in the boost pressure sensor. The relation between the boost pressure sensor signal $y_{p_{\mathrm{im}}}$ and the considered residual is dynamic, and the residual value r depends on the derivative of the boost pressure sensor signal, as well as the actual sensor signal, i.e., $r = F(y_{p_{\mathrm{im}}}, \dot{y}_{p_{\mathrm{im}}}, \ldots)$, where $F(\cdot)$ is a non-linear function. The considered fault scenario is a gain fault in the boost pressure sensor, that is, the sensor signal $y_{p_{\mathrm{im}}}$ fed to the residual generator is $y_{p_{\mathrm{im}}} = \delta \cdot p_{\mathrm{im}}$, where p_{im} is the actual boost pressure, and $\delta \neq 1$ indicates a gain fault. Gain faults in the range $\delta \in [0.2, 1.8]$ were implemented off-line by modification of the sensor signal.

Fault Detection Performance Metrics

The main metric considered in the evaluation is the power function, in this context defined as

$$\beta_{\lambda_{\mathrm{R}}}(\delta) = \Pr(\text{detection}|\delta) = \Pr(\lambda_{\mathrm{R}}(\mathcal{R}) > J|\delta), \qquad (69)$$

for the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$, defined in Section 4. Note that $\delta = 1$ in the power function (69) corresponds to that $\alpha \in \Upsilon$ and $\theta = \theta^{\mathrm{NF}}$ in the power function (34).

To study another important aspect of the detection performance, the *Mean Time to Detection* (MTD) will also be considered. Note that the choices of the values of the parameters N and J, i.e., the size of the residual sample set $\mathcal{R}$ and the detection threshold, respectively, are a trade-off between the metrics measured by the power function and the MTD, see Section 4.2.

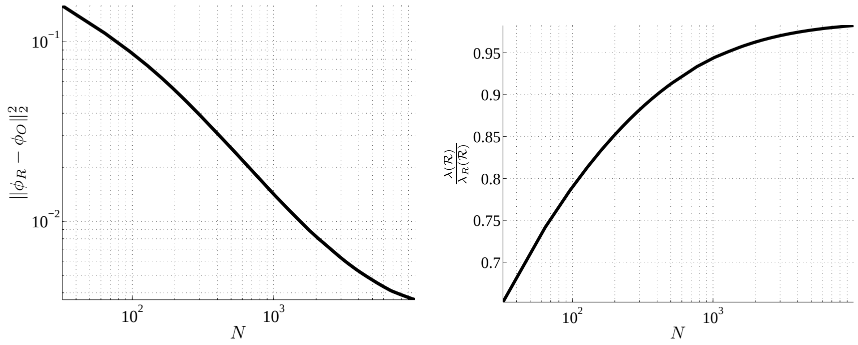
In order to be able to say something about the relative performance of the proposed residual evaluation approach, it will be compared to the often in practice used norm-based residual evaluation approach built upon the test statistic $s(\mathcal{R}) = \frac{1}{N}\sum_{r_k \in \bar{\mathcal{R}}} \bar{r}_k^2$ where $\bar{\mathcal{R}} = (\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_N)$ is a low-passed filtered version of the sample $\mathcal{R}$. Note that the purpose of this comparison merely is to give a feeling of the relative performance of the proposed residual evaluation approach, and the comparison is not claimed to be exhaustive. The low-pass filtering was in this study performed with a first-order Butterworth filter and for comparison, four different cut-off frequencies, $f_1 = 0.005$ Hz, $f_2 = 0.05$ Hz, $f_3 = 0.5$ Hz, and $f_4 = 4.5$ Hz, were used. The corresponding test statistics are denoted s_1, s_2, s_3, and s_4. Recall that the residual is sampled at a rate of 0.1 s, corresponding to a frequency of $f_s = 10$ Hz.

Implementation Details

The residual evaluation algorithm described in Section 4.2, was implemented in Matlab. To solve the optimization problem (28), a tailored solver was generated using the software tool CVXGEN (Mattingley and Boyd, 2012), see Section 4.3. With this solver, the optimization problem (28) in the setting of this study, could be solved in the time scale of 10^{-4} s. Solving the corresponding problem using the Matlab optimization toolbox results in solving times of the magnitude of 10^{-3} s. Solving the original numerator MLE problem (25) using the Matlab optimization toolbox however renders solving times of magnitude 10^{-1} s.

As said in Section 4, it is only justified, in terms of the probability of false detection, to consider the relaxed problem (28) instead of the original MLE problem (25) if the size N of the set of residual samples $\mathcal{R}$ is sufficiently large. To investigate the meaning of sufficiently large in the context of this study, Figure 9 shows a comparison of the solutions to the respective problems, as well as a comparison of the corresponding test statistics, for different values of N in the no-fault case. Figure 9a shows a comparison of the solution α^{R} to the relaxed problem (28) and the solution α^{O} to the original MLE problem (25), by means of the quantity $\|\phi_{\mathrm{R}} - \phi_{\mathrm{O}}\|_2^2$, where $\phi_{\mathrm{R}} = \sum_{i=1}^{K} \alpha_i^{\mathrm{R}} \theta_i^{\mathrm{NF}}$ and $\phi_{\mathrm{O}} = \sum_{i=1}^{K} \alpha_i^{\mathrm{O}} \theta_i^{\mathrm{NF}}$. Figure 9b shows a comparison of the test statistics $\lambda_{\mathrm{R}}(\mathcal{R})$, based on the relaxed problem and $\lambda(\mathcal{R})$, based on the original MLE problem, by means of the quantity $\frac{\lambda(\mathcal{R})}{\lambda_{\mathrm{R}}(\mathcal{R})}$. The results shown in Figure 9 are the average of 150,000 runs. Based on Figure 9, it was concluded that in the context of this study, $N > 1000$ is good enough to justify the switch to the relaxed problem. Recall from Section 4.3 that the complexity of the relaxed problem, in terms of computational time and memory, is independent of N.

The threshold J for the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$, as well as the thresholds for the norm-based



(a) Comparison of α^{R} and α^{O}.

(b) Comparison of $\lambda_{\mathrm{R}}(\mathcal{R})$ and $\lambda(\mathcal{R})$.

Figure 9: Investigation of how the relation between the solutions α^{R} and α^{O} to the relaxed (28) and original (25) MLE problems, respectively, as well as the corresponding test quantities, $\lambda_{\mathrm{R}}(\mathcal{R})$ and $\lambda(\mathcal{R})$, changes with the size N of the residual sample $\mathcal{R}$.

tests, was computed based on the estimation data set used in the learning of the no-fault distribution parameters. All thresholds were computed in order to give a probability of false detection of 5 %. All residual sample sets were taken from the validation data set by using a sliding window, see Section 4.2.

6.4 Evaluation Results

Figure 10 shows the residual and the test statistics $\lambda_{\mathrm{R}}(\mathcal{R})$ and $s_1(\mathcal{R})$, for size $N = 1024$ of the set of residual samples $\mathcal{R}$, in a test case when an abrupt fault occurs at time $t = 450$ s. The fault is a 10 % gain fault in the boost pressure sensor, which correspond to $\delta = 1.1$. For the test statistic $\lambda_{\mathrm{R}}(\mathcal{R})$, the parameter $\theta^{\mathrm{NF}} = \theta^{(10)}$ illustrated in Figure 8 was used.

It can be noted that, as in Figure 1, the residual in Figure 10 is non-zero in the no-fault case, i.e., for $t < 450$ s, and its distribution exhibit non-stationary features in both the no-fault and fault cases. Further, it can also be seen that the difference between the residual in the no-fault and fault cases are small, but that there is a significant difference between the test statistic $\lambda_{\mathrm{R}}(\mathcal{R})$ in the no-fault and fault cases. Since $\lambda_{\mathrm{R}}(\mathcal{R})$ is above the threshold in the fault case, the present fault can be detected. The fault can however not be detected in a reliable way with the test statistic $s_1(\mathcal{R})$, which in this case performed better than each of the test statistics $s_2(\mathcal{R})$, $s_3(\mathcal{R})$, and $s_4(\mathcal{R})$.

Power as Function of N

To illustrate how the power of the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$ varies with the number N of residual samples in $\mathcal{R}$, Figure 11 shows the power function for the test for different values of N and parameter $\theta^{\mathrm{NF}} = \theta^{(10)}$. Figure 11 clearly shows that the power of the test increases with N.

In Figure 11, it can be seen that as small faults as $\delta \approx 0.95$ and $\delta \approx 1.05$, corresponding to gain faults in the boost pressure sensor of about ± 5 %, may be possible to detect if N

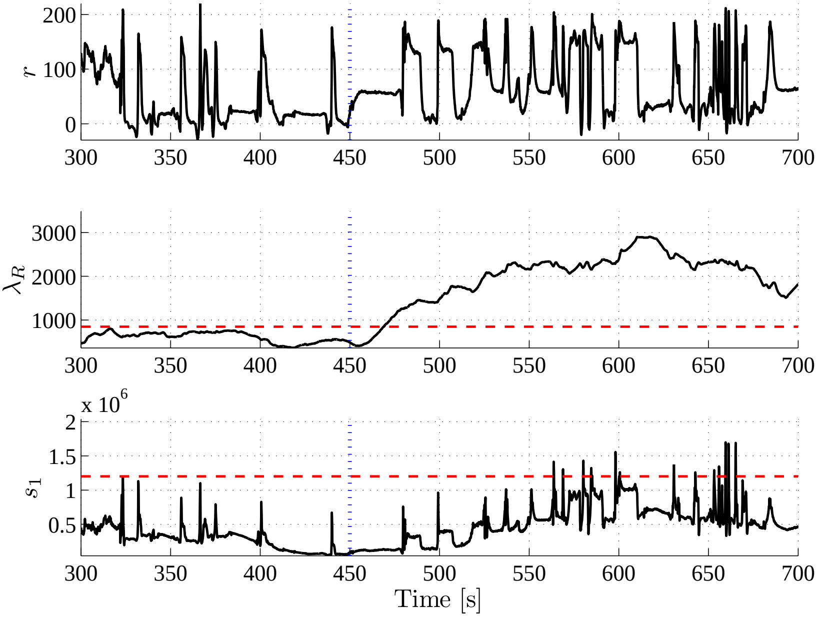


Figure 10: Residual r (top), test statistic $\lambda_R(\mathcal{R})$ (middle), and test statistic $s_1(\mathcal{R})$ (bottom), when an abrupt fault occurs at $t = 450$ s. The fault is a 10 % gain fault in the boost pressure sensor, which corresponds to $\delta = 1.1$.

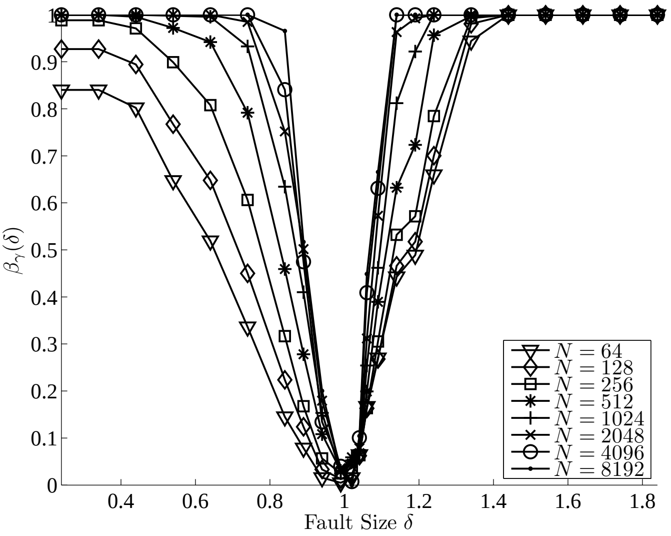


Figure 11: Power function $\beta_{\lambda_{\mathrm{R}}}(\delta)$ for the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$ for different sizes N of the sample $\mathcal{R}$. The power increases with N.

is sufficiently large. To further illustrate this, Figure 12 shows the Receiver Operating Characteristic (ROC) curve for different values of N, for a test case with $\delta = 1.05$. The ROC curve shows the relation between the True Positive Rate (TPR) of detection (y-axis), and the False Positive Rate (FPR) of detection (x-axis), i.e., the relation between correct detections and false detections, when the detection threshold J is varied. Figure 12 again shows that the detection performance increases with N, but also that the rate of false detections can be made lower than the rate of actual detections even for moderate values of N.

Power as Function of K

To analyze how the power of the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$ varies with different values of the parameter $\theta^{\mathrm{NF}} = \theta^{(K)}$, specifying the set of no-fault residual distributions, or more specifically with K, i.e., the number of operating modes of the system, Figure 13 shows the power function for the test for different values of K. All considered parameters $\theta^{(K)}$ were obtained by means of the algorithm described in Section 5. To also see how the power of the test depends on the relation between K and N, Figure 13 shows how the power function depends on K for different values of N.

The general conclusion from the evaluation shown in Figure 13, is that for a given $256 \le N \le 1024$, the power of the test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$ is almost equal for all considered K. For small N, e.g., $N = 64$, however, the power increases with K and for large N, e.g., $N = 4096$, the power increases as K decreases. The liable rationale behind this is that a small K results in a generic and averaged, in terms of operating modes, description of

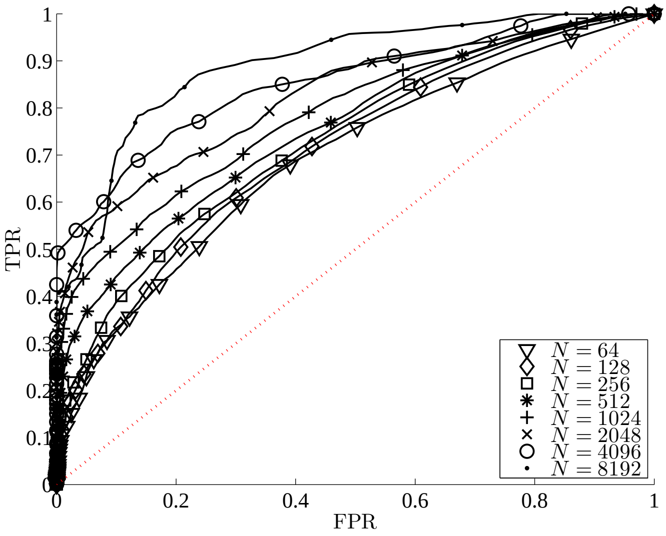


Figure 12: ROC for test $\lambda_{\mathrm{R}}(\mathcal{R}) > J$ when $\delta = 1.05$ for different sizes N of the sample $\mathcal{R}$.

the set of no-fault residual distributions. A large set of residual samples typically means residual samples from a variety of operating modes, while a small set of residual samples on the other hand means residual samples from only a few operating modes. This means that a parameter θ^{NF} corresponding to a small K, typically can describe the distribution of a large set of no-fault residual samples, i.e., a large N, better than the distribution of a small set of no-fault residual samples, i.e., a small N. An accurate description of the no-fault residual distribution makes it possible to distinguish such from a faulty residual distribution, which indeed means good detection power.

Comparison of Tests

Figure 14 shows a comparison of the power functions for the tests based on the test statistics $\lambda_{\mathrm{R}}(\mathcal{R})$, $s_1(\mathcal{R})$, $s_2(\mathcal{R})$, $s_3(\mathcal{R})$, and $s_4(\mathcal{R})$, for different values of the parameter N, which specifies the number of residual samples in $\mathcal{R}$. For the test statistic $\lambda_{\mathrm{R}}(\mathcal{R})$, the parameter $\theta^{\mathrm{NF}} = \theta^{(10)}$ illustrated in Figure 8 was used.

Figure 14 shows that the powers of all tests increases with N and that the differences between the power of the tests seem to decrease with an increasing N. It can also be seen that the power function for the test based on $\lambda_{\mathrm{R}}(\mathcal{R})$ is near symmetric for all N, while the power functions for the other tests are asymmetric and tend to be less powerful for faults sizes $\delta < 1$. The difference in power for $\delta < 1$ is for example significant for $N = 64$.

The mean time to detection (MTD) for each of the tests based on $\lambda_{\mathrm{R}}(\mathcal{R})$, $s_1(\mathcal{R})$, $s_2(\mathcal{R})$, $s_3(\mathcal{R})$, and $s_4(\mathcal{R})$, is shown in Figure 15, for different sizes N of the sample $\mathcal{R}$.

In order to get comparable results, the MTD was computed as the mean of the detection time for the two largest faults, corresponding to $\delta = 0.2$ and $\delta = 1.8$, since all

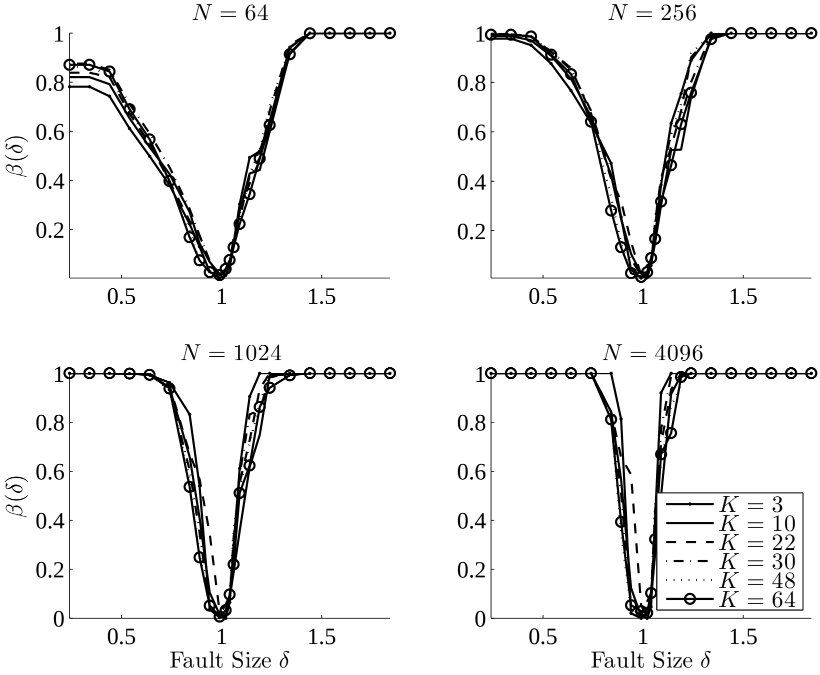


Figure 13: Comparison of power functions for the test based on $\lambda_{\mathrm{R}}(\mathcal{R})$ for a set of no-fault distribution parameters $\theta^{(K)}$ with different values of K.

considered test statistics are able to detect these faults to some extent, see Figure 14. Each fault was injected in the test sequence at 10 time instances.

In Figure 15, it can be seen that the MTD's for all tests increase for $N > 256$. For $N < 256$, however, the MTD decreases with N for the norm-based tests and increases with N for the test based on $\lambda_{\mathrm{R}}(\mathcal{R})$. It is worth noting that the MTD for the test based on $\lambda_{\mathrm{R}}(\mathcal{R})$ is smaller for all N than the MTD's for all other tests.

7 Conclusions

As illustrated by Figure 1, residuals in practice often deviate from zero even in the no-fault case due to uncertainties and disturbances caused by for example modeling errors, measurement noise, and unmodeled phenomena. In addition, due to changes in the operating mode of the underlying system, the magnitude of uncertainties and disturbances is time-varying, causing the behavior of residuals to be non-stationary. To handle these issues, a novel statistical residual evaluation approach has been proposed.

The main contribution is to base the residual evaluation on an explicit comparison of the probability distribution of the residual, estimated on-line using current data, with a no-fault residual distribution. The no-fault distribution is based on a set of a-priori known

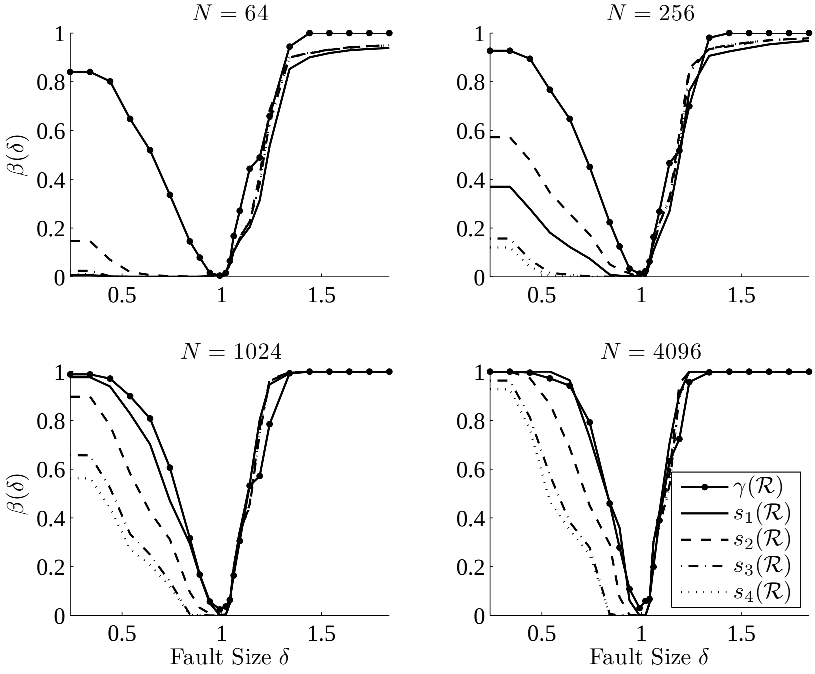


Figure 14: Comparison of power functions for the tests based on $\lambda_{\mathrm{R}}(\mathcal{R})$ (solid with dot markers), $s_1(\mathcal{R})$ (solid), $s_2(\mathcal{R})$ (dashed), $s_3(\mathcal{R})$ (dash-dotted), and $s_4(\mathcal{R})$ (dotted), for different sizes N of the sample $\mathcal{R}$.

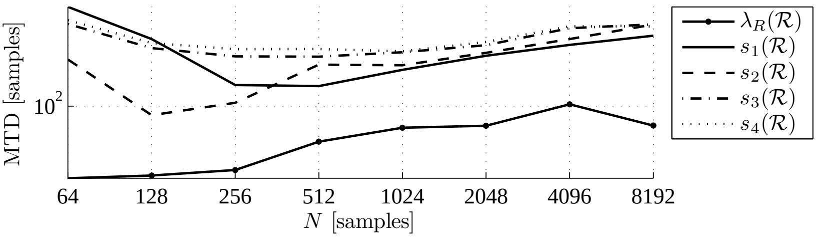


Figure 15: Comparison of the Mean Time to Detection (MTD) for the tests based on $\lambda_{\mathrm{R}}(\mathcal{R})$ (solid with dot markers), $s_1(\mathcal{R})$ (solid), $s_2(\mathcal{R})$ (dashed), $s_3(\mathcal{R})$ (dash-dotted), and $s_4(\mathcal{R})$ (dotted), for different sizes N of the sample $\mathcal{R}$.

no-fault distributions, and is continuously adapted to the current operating mode of the system by means of the likelihood maximization problem (26). A computational efficient version of the residual evaluation test statistic suitable for online implementation has been derived by considering a properly chosen approximation (28) to the maximization problem (26). The fault detection properties of the resulting residual evaluation test have been analyzed by means of Theorems 2 and 3.

As a second contribution, a method has been proposed for learning the required set of no-fault residual distributions off-line from training data. Thus, by using this method, the overall residual evaluation method is data-driven and no assumptions regarding the properties of the probability distribution of the residual, nor the properties of the faults to be detected, are needed. The method was given by means of an algorithm based on K-means clustering, and was theoretically justified in Theorem 5.

The proposed residual evaluation method has been evaluated with measurement data on a residual for fault detection in the gas-flow system of a Scania truck diesel engine. The proposed test statistic performs well despite non-conventional properties of the considered residual. For instance, the method outperforms regular norm-based methods using constant thresholding in the sense that small faults can be detected in cases where these methods fail. It has been empirically investigated how the fault detection performance of the proposed method is influenced by different values of the involved parameters.

Acknowledgment

This work was sponsored by Scania and VINNOVA (Swedish Governmental Agency for Innovation Systems).

A Proofs of Theorems and Lemmas

Lemma 5. *Let $\{r_1, r_2, \dots, r_N\}$ be a set of iid samples from the pmf $p(r|\phi)$ described by (1), let $\phi^\star_N$ be the MLE of ϕ based on $\{r_1, r_2, \dots, r_N\}$, and let*

$$\Phi' = \left\{ \phi \in \mathbb{R}^M \ : \ \phi_j > 0, \quad \sum_{j=1}^{M} \phi_j = 1 \right\}. \tag{70}$$

Then, for every $\varepsilon > 0$ and $\phi \in \Phi'$, it holds that

$$\lim_{N\to\infty} \Pr\left(|\phi^\star_N - \phi| \geq \varepsilon\right) = 0. \tag{71}$$

Proof. According to (Casella and Berger, 2001, Theorem 10.1.6), (71) holds if the following regularity conditions on $p(r|\phi)$ are satisfied: i) $r_1, r_2, \dots, r_N$ are iid samples from $p(r|\phi)$; ii) the parameter ϕ is identifiable, i.e., if $\phi \neq \phi'$, then $p(r|\phi) \neq p(r|\phi')$; iii) the densities $p(r|\phi)$, for all $\phi \in \Phi'$, have common support, and $p(r|\phi)$ is differentiable in ϕ; iv) the parameter space Φ' contains an open set φ of which the true parameter ϕ is an interior point. It is first noted condition i) is trivially satisfied by assumption. For condition ii), assume that $\phi \neq \phi'$. This implies that there exists $k \in \{1, 2, \ldots, M\}$ such that $\phi_k \neq \phi'_k$, and it holds that

$$p\left(r = x_k|\phi\right) = \phi_k \neq \phi'_k = p\left(r = x_k|\phi\right),$$

and hence $p\left(r|\phi\right) \neq p\left(r|\phi'\right)$. Regarding condition iii), it is recalled that the support of a function is the set of points where the function is non-zero zero. Thus, the first part of condition iii) is trivially satisfied due to the form of the pmf $p\left(r|\phi\right)$ in (1) and the properties of the parameter space Φ' defined by (70). Considering next the differentiability, it holds that

$$\frac{\partial}{\partial \phi_k} p\left(x_j|\phi\right) = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases}$$

for $j = 1, 2, \ldots, M$, and hence condition iii) is satisfied. For condition iv), it is noted that the parameter space Φ' is an open set. Therefore every $\phi \in \Phi'$ is an interior point of an open set and condition iv) is satisfied. This completes the proof. □

Lemma 6. *Let $\mathcal{R}$ be a set of residual samples, $c_1, c_2, \ldots, c_M$ be defined according to (13), and let Assumptions 1 and 2 be valid. Further, let $\alpha^\star \in \Upsilon$ and $\theta^\star \in \Theta^{(K)}$ fulfill*

$$\sum_{i=1}^{K} \alpha_i^\star \, \theta_{ij}^\star = \frac{c_j}{N}, \tag{72}$$

where $N = \sum_{j=1}^{M} c_j$. Then, for each $\alpha \in \Upsilon$ and $\theta \in \Theta^{(K)}$ it holds that

$$D\left(p\left(r|\alpha^\star, \theta^\star\right) \| p\left(r|\alpha, \theta\right)\right) = \frac{1}{N} \log \frac{\mathcal{L}\left(\alpha^\star, \theta^\star|\mathcal{R}\right)}{\mathcal{L}\left(\alpha, \theta|\mathcal{R}\right)}, \tag{73}$$

where $p\left(r|\cdot\right)$ is given by (3) and $\mathcal{L}\left(\cdot, \cdot|\mathcal{R}\right)$ by (14).

Proof. It is first noted that $p\left(x_j|\alpha, \theta\right) = p\left(r = x_j|\alpha, \theta\right) = \sum_{i=1}^{K} \alpha_i \theta_{ij}$ according to (3) and (1). By using this, (62), and (72), the left hand side of (73) can be written as

$$\begin{aligned} D\left(p\left(r|\alpha^\star, \theta^\star\right) \| p\left(r|\alpha, \theta\right)\right) &= \sum_{j=1}^{M} p\left(x_j|\alpha^\star, \theta^\star\right) \log \frac{p\left(x_j|\alpha^\star, \theta^\star\right)}{p\left(x_j|\alpha, \theta\right)} \\ &= \sum_{j=1}^{M} \left(\sum_{i=1}^{K} \alpha_i^\star \theta_{ij}^\star\right) \log \frac{\sum_{i=1}^{K} \alpha_i^\star \theta_{ij}^\star}{\sum_{i=1}^{K} \alpha_i \theta_{ij}} \\ &= \sum_{j=1}^{M} \frac{c_j}{N} \log \frac{\frac{c_j}{N}}{\sum_{i=1}^{K} \alpha_i \theta_{ij}}. \end{aligned} \tag{74}$$

Consider next the right hand side of (73). Due to the prerequisites of the lemma, the likelihood $l\left(\cdot, \cdot|\mathcal{R}\right)$ is given by (15). With this, the right hand side of (73) can be written as

$$
\begin{aligned}
\frac{1}{N}\log\frac{\mathcal{L}\left(\alpha^{\star},\theta^{\star}|\mathcal{R}\right)}{\mathcal{L}\left(\alpha,\theta|\mathcal{R}\right)} &= \frac{1}{N}\left(l\left(\alpha^{\star},\theta^{\star}|\mathcal{R}\right)-l\left(\alpha,\theta|\mathcal{R}\right)\right)\\
&= \frac{1}{N}\left(\sum_{j=1}^{M}c_j\log\left[\sum_{i=1}^{K}\alpha_i^{\star}\,\theta_{ij}^{\star}\right]-\sum_{j=1}^{M}c_j\log\left[\sum_{i=1}^{K}\alpha_i\,\theta_{ij}\right]\right)\\
&= \frac{1}{N}\sum_{j=1}^{M}c_j\log\frac{\sum_{i=1}^{K}\alpha_i^{\star}\,\theta_{ij}^{\star}}{\sum_{i=1}^{K}\alpha_i\,\theta_{ij}}\\
&= \frac{1}{N}\sum_{j=1}^{M}c_j\log\frac{\frac{c_j}{N}}{\sum_{i=1}^{K}\alpha_i\,\theta_{ij}},
\end{aligned}
$$

which equals (74). □

Proof of Lemma 3. First note that (63) implies that for each $\theta_k \in \mathrm{P}_i$ there is an element $\mathcal{R}_k \in \mathrm{T}_i$, and vice versa. By using the same arguments as in the proof of Theorem 4, it holds that the log-likelihood $l\left(\theta_k|\mathcal{R}_k\right)$ is given by (57). Thus, each MLE problem in (49) is equivalent to (16) if $K = 1$ and $\alpha_1 = 1$, or equivalently (17), and Theorem 1 is applicable. From Theorem 1 it then follows that $\theta_{kj} = \frac{c_{kj}}{n}$, $j = 1, 2, \ldots, M$, for each $\theta_k \in \Psi$. From Lemma 6, again with $K = 1$ and $\alpha_1 = 1$, it follows that

$$
\begin{aligned}
D\left(p\left(r|\theta_k\right)\|p\left(r|\theta\right)\right) &= \frac{1}{n}\log\frac{\mathcal{L}\left(\theta_k|\mathcal{R}_k\right)}{\mathcal{L}\left(\theta|\mathcal{R}_k\right)}\\
&= \frac{1}{n}\left(l\left(\theta_k|\mathcal{R}_k\right)-l\left(\theta|\mathcal{R}_k\right)\right),
\end{aligned}
\tag{75}
$$

for any $\theta \in \Theta^{(1)}$. Consider now the inequality (65). By exploiting (75), the inequality (65) can be written as

$$
\begin{gathered}
D\left(p\left(r|\theta_k\right)\|p\left(r|\theta_p\right)\right) \le D\left(p\left(r|\theta_k\right)\|p\left(r|\theta_q\right)\right)\\
\Longleftrightarrow\\
\frac{1}{n}\left(l\left(\theta_k|\mathcal{R}_k\right)-l\left(\theta_p|\mathcal{R}_k\right)\right) \le \frac{1}{n}\left(l\left(\theta_k|\mathcal{R}_k\right)-l\left(\theta_q|\mathcal{R}_k\right)\right)\\
\Longleftrightarrow\\
l\left(\theta_p|\mathcal{R}_k\right) \ge l\left(\theta_q|\mathcal{R}_k\right),
\end{gathered}
$$

and equivalence between (65) and (64) has been established. Consider now (66). By again using (75) and (63), it follows that

$$
\arg\min_{\theta\in\Theta^{(1)}}\sum_{\theta_k\in\mathrm{P}_i}D\left(p\left(r|\theta_k\right)\|p\left(r|\theta\right)\right) = \arg\min_{\theta\in\Theta^{(1)}}\sum_{\mathcal{R}_k\in\mathrm{T}_i}\frac{1}{n}\left(l\left(\theta_k|\mathcal{R}_k\right)-l\left(\theta|\mathcal{R}_k\right)\right). \tag{76}
$$

Since θ only is present in the term $l\left(\theta|\mathcal{R}_k\right)$ in (76), and due to the minus sign in front of this term, (76) can be written as

$$
\arg\min_{\theta\in\Theta^{(1)}}\sum_{\mathcal{R}_k\in\mathrm{T}_i}\frac{1}{n}\left(l\left(\theta_k|\mathcal{R}_k\right)-l\left(\theta|\mathcal{R}_k\right)\right) = \arg\max_{\theta\in\Theta^{(1)}}\sum_{\mathcal{R}_k\in\mathrm{T}_i}l\left(\theta|\mathcal{R}_k\right),
$$

and the proof is complete. □

Proof of Lemma 4. First note that by using the same arguments as in Theorem 4 and Lemma 3 it holds that the likelihood function $l\left(\theta_k|\mathcal{R}_k\right)$ is given by (57) and that $\theta_{kj} = \frac{c_{kj}}{n}$, $j = 1, 2, \dots, M$, for each $\theta_k \in \Psi$. Consider now the claim (67). Define T_i according to (63) for $i = 1, 2, \dots, K$. Due to (63), (52), and since $\theta_k \in \mathrm{P}_i \in \mathrm{P}^\star$ and $\mathrm{P}^\star = \bigcup_{\mathrm{P}_i \in \mathrm{P}^\star} \bigcup_{\theta_k \in \mathrm{P}_i} \theta_k = \Psi$, it follows that

$$\begin{aligned} \theta^\star_{ij} = \frac{1}{|\mathrm{P}_i|} \sum_{\theta_k \in \mathrm{P}_i} \theta_{kj} &= \frac{1}{|\mathrm{T}_i|} \sum_{\mathcal{R}_k \in \mathrm{T}_i} \frac{c_{kj}}{n} \\ &= \frac{\sum_{\mathcal{R}_k \in \mathrm{T}_i} c_{kj}}{|\mathrm{T}_i| \cdot n}, \end{aligned} \tag{77}$$

for $i = 1, 2, \dots, K$ and $j = 1, 2, \dots, M$. It is now noted that $\sum_{\mathcal{R}_k \in \mathrm{T}_i} c_{kj}$ denotes the number of samples in all $\mathcal{R}_k \in \mathrm{T}_i$ that takes value x_j, and that $|\mathrm{T}_i| \cdot n$ denotes the total number of samples in all $\mathcal{R}_k \in \mathrm{T}_i$, which indeed is equal to $\sum_{j=1}^{M} \sum_{\mathcal{R}_k \in \mathrm{T}_i} c_{kj}$. From (77), it can thus be deduced that $\theta^\star_{ij} = \frac{c_j}{\sum_{j=1}^{M} c_j}$, where $c_j = \sum_{\mathcal{R}_k \in \mathrm{T}_i} c_{kj}$. Theorem 1 then implies that

$$\theta^\star_i = \arg\max_{\theta \in \Theta^{(1)}} l\left(\theta| \cup_{\mathcal{R}_k \in \mathrm{T}_i} \mathcal{R}_k\right) \tag{78}$$

for $i = 1, 2, \dots, K$. Now note that due to the properties of the log-likelihood function (57) it holds that

$$\begin{aligned} l\left(\theta| \cup_{\mathcal{R}_k \in \mathrm{T}_i} \mathcal{R}_k\right) &= \sum_{j=1}^{M} \sum_{\mathcal{R}_k \in \mathrm{T}_i} c_{kj} \log \theta_j \\ &= \sum_{\mathcal{R}_k \in \mathrm{T}_i} \sum_{j=1}^{M} c_{kj} \log \theta_j \\ &= \sum_{\mathcal{R}_k \in \mathrm{T}_i} l\left(\theta|\mathcal{R}_k\right) \end{aligned} \tag{79}$$

and thus (78) turns into

$$\theta^\star_i = \arg\max_{\theta \in \Theta^{(1)}} \sum_{\mathcal{R}_k \in \mathrm{T}_i} l\left(\theta|\mathcal{R}_k\right), \tag{80}$$

for $i = 1, 2, \dots, K$. From (80), the claim (67) follows directly via (66) in Lemma 3. Now turn to the claim (68) and denote

$$M(\mathrm{P}^\star) = \sum_{i=1}^{K} \sum_{\theta_k \in \mathrm{P}_i} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta^\star_i\right)\right), \tag{81}$$

where, due to (67), it holds that

$$\theta^\star_i = \arg\min_{\theta \in \Theta^{(1)}} \sum_{\theta_k \in \mathrm{P}_i} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta\right)\right), \tag{82}$$

for $i = 1, 2, \ldots, K$. To show that (68) holds by contradiction, assume that there exists $\theta_p \in \mathrm{P}_i$, for some $\mathrm{P}_i \in \mathrm{P}^\star$, such that

$$D\left(p\left(r|\theta_p\right) \| p\left(r|\theta_i^\star\right)\right) > D\left(p\left(r|\theta_p\right) \| p\left(r|\theta_j^\star\right)\right), \tag{83}$$

for some $j = 1, 2, \ldots, K$. Now define

$$\bar{M} = M(\mathrm{P}^\star) + D\left(p\left(r|\theta_p\right) \| p\left(r|\theta_j^\star\right)\right) - D\left(p\left(r|\theta_p\right) \| p\left(r|\theta_i^\star\right)\right), \tag{84}$$

and note that, due to (83), it holds that $\bar{M} < M(\mathrm{P}^\star)$. Define a new partition P' of Ψ by moving θ_p from block $\mathrm{P}_i \in \mathrm{P}^\star$ to block $\mathrm{P}_j \in \mathrm{P}^\star$, i.e., let $\mathrm{P}' = (\mathrm{P}'_1, \mathrm{P}'_2, \ldots, \mathrm{P}'_K)$ where

$$\mathrm{P}'_l = \begin{cases} \mathrm{P}_i \setminus \left\{\theta_p\right\}, & l = i \\ \mathrm{P}_j \cup \left\{\theta_p\right\}, & l = j \\ \mathrm{P}_l, & \text{else.} \end{cases} \tag{85}$$

Form

$$M(\mathrm{P}') = \sum_{l=1}^{K} \sum_{\theta_k \in \mathrm{P}'_l} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta'_l\right)\right) \tag{86}$$

where

$$\theta'_l = \arg\min_{\theta \in \Theta^{(1)}} \sum_{\theta_k \in \mathrm{P}'_l} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta\right)\right), \tag{87}$$

for $l = 1, 2, \ldots, K$. It is first noted that due to Lemma 3, and a similar argument as above including (79), (78), and (77), the distribution parameters θ'_l, $l = 1, 2, \ldots, K$, satisfy (52). Consider now the quantity $\bar{M} - M(\mathrm{P}')$, which by using (84) and (86) can be written as

$$\begin{aligned} \bar{M} - M(\mathrm{P}') = M(\mathrm{P}^\star) &+ D\left(p\left(r|\theta_p\right) \| p\left(r|\theta_j^\star\right)\right) - D\left(p\left(r|\theta_p\right) \| p\left(r|\theta_i^\star\right)\right) \\ &- \sum_{l=1}^{K} \sum_{\theta_k \in \mathrm{P}'_l} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta'_l\right)\right). \end{aligned} \tag{88}$$

Due to (81) and the properties of the partition P' as given by (85), it holds that

$$\begin{aligned} &M(\mathrm{P}^\star) + D\left(p\left(r|\theta_p\right) \| p\left(r|\theta_j^\star\right)\right) - D\left(p\left(r|\theta_p\right) \| p\left(r|\theta_i^\star\right)\right) \\ &= \sum_{l=1}^{K} \sum_{\theta_k \in \mathrm{P}'_l} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta_l^\star\right)\right), \end{aligned}$$

and therefore (88) can be written as

$$\bar{M} - M(\mathrm{P}') = \sum_{l=1}^{K} \sum_{\theta_k \in \mathrm{P}'_l} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta_l^\star\right)\right) - \sum_{l=1}^{K} \sum_{\theta_k \in \mathrm{P}'_l} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta'_l\right)\right). \tag{89}$$

It is now noted that due to (87) it holds that

$$\sum_{l=1}^{K} \sum_{\theta_k \in \mathrm{P}'_l} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta'_l\right)\right) \leq \sum_{l=1}^{K} \sum_{\theta_k \in \mathrm{P}'_l} D\left(p\left(r|\theta_k\right) \| p\left(r|\theta^\star_l\right)\right)$$

and therefore (89) implies that $\bar{M} - M(\mathrm{P}') \geq 0$, or equivalently that $M(\mathrm{P}') \leq \bar{M}$. Thus, it holds that $M(\mathrm{P}') \leq \bar{M} < M(\mathrm{P}^\star)$, which contradicts the statement (51). Hence, (83) cannot hold and consequently (68) holds and the proof is complete. □

References

M. Abid, W. Chen, S. X. Ding, and A.Q. Khan. Optimal residual evaluation for nonlinear systems using post-filter and threshold. *International Journal of Control*, 84(3):526 – 39, 2011.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716 – 723, 1974.

I. M. Al-Salami, S. X. Ding, and P. Zhang. Statistical based residual evaluation for fault detection in networked control systems. In *Proceedings of Workshop on Advances Control and Diagnosis*, Nancy, France, 2006. Nancy University.

I. M. Al-Salami, K. Chabir, D. Sauter, and C. Aubrun. Adaptive thresholding for fault detection in networked control systems. In *Proceedings of the IEEE International Conference on Control Applications*, pages 446 – 451, Yokohama, Japan, 2010.

D. Aloise, A. Deshpande, P. Hansen, and P. Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75:245–248, 2009.

A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6(10):1705 – 1749, 2005.

M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, 1993.

D. Basu. On the elimination of nuisance parameters. *Journal of the American Statistical Association*, 72(358):355–366, 1977.

J. O. Berger, B. Liseo, and R. L. Wolpert. Integrated likelihood methods for eliminating nuisance parameters. *Statistical Science*, 14(1):1–22, 1999.

P. Berkhin. Survey of clustering data mining techniques. *Techniques*, 10(c):1–56, 2002.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

A. Bjorck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, PA, 1996.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, second edition, 2006.

M. R. Blas and M. Blanke. Stereo vision with texture learning for fault-tolerant automatic baling. *Computers and Electronics in Agriculture*, 75(1):159 – 68, 2011.

L. Bottou and Y. Bengio. Convergence properties of the K-Means algorithm. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, Denver, 1995.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, Cambridge, U.K, 2004.

G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, second edition, 2001.

J. Chen and R. J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. MA: Kluwer, Boston, 1999.

R. N. Clark. State estimation schemes for instrument fault detection. In R. J. Patton, P. M. Frank, and R. N. Clark, editors, *Fault Diagnosis in Dynamic Systems: Theory and Application*, chapter 2, pages 21–45. Prentice Hall, 1989.

L. Davies, U. Gather, D. Nordman, and H. Weinert. A comparison of automatic histogram constructions. *ESAIM: Probability and Statistics*, 13:181–196, 2009.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

S. X. Ding, P. Zhang, and E. L. Ding. Fault detection system design for a class of stochastically uncertain systems. In Hong-Yue Zhang, editor, *Fault Detection, Supervision and Safety of Technical Processes 2006*, pages 705 – 710. Elsevier Science Ltd, 2007.

A. Emami-Naeini, M. M. Akhter, and S. M. Rock. Effect of model uncertainty on failure detection: the threshold selector. *IEEE Transactions on Automatic Control*, 33(12):1106 –1115, 1988.

P. M. Frank. Enhancement of robustness in observer-based fault-detection. *International Journal of Control*, 59(4):955–981, 1994.

P. M. Frank. Residual evaluation for fault diagnosis based on adaptive fuzzy thresholds. In *IEE Colloquium on Qualitative and Quantitative Modelling Methods for Fault Diagnosis*, pages 401 –411, 1995.

P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6):403 – 424, 1997.

J. J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, 1998.

F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, 2000.

G. H. Hardy, J. E. Littlewood, and G. Polya. *Inequalities*. Cambridge, 1934.

K.H. Haskell and R.J. Hanson. An algorithm for linear least squares problems with equality and nonnegativity constraints. *Mathematical Programming*, 21(1):98–118, 1981.

T. Höfling and R. Isermann. Fault detection based on adaptive parity equations and single-parameter tracking. *Control Engineering Practice*, 4(10):1361 – 1369, 1996.

M. Inaba, N. Katoh, and H. Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract). In *Proceedings of the tenth annual symposium on Computational geometry*, SCG '94, pages 332–339, New York, NY, USA, 1994. ACM.

A. Ingimundarson, A. G. Stefanopoulou, and D. A. McKay. Model-based detection of hydrogen leaks in a fuel cell stack. *IEEE Transactions on Control Systems Technology*, 16 (5):1004 –1012, 2008.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

W. Li, Z. Zhu, and S. X. Ding. Fault detection design of networked control systems. *IET Control Theory and Applications*, 5(12):1439 – 49, 2011.

L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2 edition, 1999.

S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.

B. Manthey and H. Röglin. Worst-case and smoothed analysis of k-means clustering with bregman divergences. In Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra, editors, *Algorithms and Computation*, volume 5878 of *Lecture Notes in Computer Science*, pages 1024–1033. Springer Berlin, Heidelberg, 2009.

J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, 2010.

J. Mattingley and S. Boyd. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

S. A. Murphy and A. W. van der Vaart. On profile likelihood. *Journal of the American Statistical Association*, 95(450):449–465, 2000.

Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, PA, 1994.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

M. Nyberg and T. Stutte. Model based diagnosis of the air path of an automotive diesel engine. *Control Engineering Practice*, 12(5):513 – 525, 2004.

W. M. Patefield. On the maximized likelihood function. *The Indian Journal of Statistics, Series B (1960-2002)*, 39(1):92–96, 1977.

Y. Peng, A. Youssouf, P. Arte, and M. Kinnaert. A complete procedure for residual generation and evaluation with application to a heat exchanger. *IEEE Transactions on Control Systems Technology*, 5(6):542 – 555, 1997.

S. Z. Selim and M. A. Ismail. K-Means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):81–87, 1984.

H. Sneider and P. M. Frank. Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation. *IEEE Transactions on Control Systems Technology*, 4(3):274 –282, 1996.

T. Söderström and P. Stoica. *System Identification*. Prentice-Hall Int., London, UK, 1989.

C. Svärd and M. Nyberg. Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(6):1310–1328, 2010.

C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Residual evaluation for fault diagnosis by data-driven analysis of non-stationary probability distributions. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011)*, 2011.

J. Wahlström and L. Eriksson. Modeling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 225(7), 2011.

X. Wei, H. Liu, and Y. Qin. Fault diagnosis of rail vehicle suspension systems by using glrt. In *Control and Decision Conference (CCDC), 2011 Chinese*, pages 1932 –1936, 2011.

A. Willsky and H. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control*, 21(1): 108 – 112, 1976.

S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, PA, 1997.

X. Zhang, M. M. Polycarpou, and T. Parisini. A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE Transactions on Automatic Control*, 47(4):576 –593, 2002.

M. Zhong, H. Ye, S.X. Ding, and G. Wang. Observer-based fast rate fault detection for a class of multirate sampled-data systems. *IEEE Transactions on Automatic Control*, 52 (3):520 – 525, 2007/03/.

Y. Zhu and X. Rong Li. Recursice least squares with linear constraints. *Communications in Information and Systems*, 7(3):287–312, 2007.

Paper D

Automotive Engine FDI by Application of an Automated Model-Based and Data-Driven Design Methodology☆

D

☆ Submitted to *Control Engineering Practice*, 2012.

Automotive Engine FDI by Application of an Automated Model-Based and Data-Driven Design Methodology

Carl Svärd, Mattias Nyberg, Erik Frisk, and Mattias Krysander

Vehicular Systems, Department of Electrical Engineering,
Linköping University, SE-581 83 Linköping, Sweden.

Abstract

Fault detection and isolation (FDI) in automotive diesel engines is important in order to achieve and guarantee low exhaust emissions, high vehicle uptime, and efficient repair and maintenance. This paper illustrates how a set of general methods for model-based sequential residual generation and data-driven statistical residual evaluation can be combined into an automated design methodology. The automated design methodology is then utilized to create a complete FDI-system for an automotive diesel engine. The performance of the obtained FDI-system is evaluated using measurements from road drives and engine test-bed experiments. The overall performance of the FDI-system is good in relation to the required design effort, in particular since no specific tuning of the FDI-system, nor any adaption of the design methodology, were needed. It is illustrated how estimations of the statistical powers of the fault detection tests in the FDI-system can be used to further increase the performance, specifically in terms of fault isolability.

1 Introduction

Emission related legislations (United Nations, 2008; European Parliament, 2009; California EPA, 2010; United States EPA, 2009) require on-board diagnosis (OBD) of all faults in automotive engines that may lead to increased exhaust emissions. In addition, fault accommodation by, e.g., fault-tolerant control (FTC) (Blanke et al., 2006), and off-board diagnosis, are means in order to meet dependability requirements in the form of high vehicle uptime, high safety, and efficient repair. A necessity for both diagnosis and fault accommodation is fault detection and isolation (FDI).

Automotive engines pose several challenges and difficulties when it comes to design of FDI-systems. Typically, engines are optimized for low-cost and high functionality, and not for FDI, which means that there is no hardware redundancy in the form of multiple sensors. To obtain good detection and isolation of faults it is therefore necessary to employ analytical redundancy and model-based FDI. Due to the inherent complexity of automotive engines, as well as their multi-domain features due to chemical, mechanical, and thermodynamic subsystems, modeling results in large-scale, dynamic, and highly non-linear systems (Wahlström and Eriksson, 2011). Thus, such models must be handled by the methods used in the design of the FDI-system.

As a consequence of the complexity of automotive engines, in combination with their wide operating range, models are typically not fully capable of capturing their behavior in all operating modes. This results in model errors, and in particular stationary model errors (Höckerdal et al., 2011a,b), regardless of substantial modeling work. In addition, a model may be more accurate in one operating mode than another and since the operating mode of the engine varies in time, so does the magnitude and nature of the model errors. These aspects must be taken into account in the design of the FDI-system.

It is clear that design of a complete model-based FDI-system for an automotive engine, and for large-scale real-world systems in general, is an intricate task that demands a substantial engineering effort. An optimal solution in general requires detailed knowledge of the behavior of the system and well-defined requirements, which typically not is available during early design stages. In order to make the overall design process more systematic and efficient, and in this way enable re-design or re-configuration, and eventually higher quality, a generic automated methodology for design of FDI-systems has been developed.

The design methodology relies on previously developed methods for sequential residual generation (Svärd and Nyberg (2010), Paper B), and statistical residual evaluation (Paper C). The residual generation methods described in Svärd and Nyberg (2010) and Paper B are together able to design residual generators for fault detection and isolation in systems described by complex large-scale models. This was demonstrated in Svärd and Nyberg (2012), where they were combined with a residual evaluation approach based on the Kullback-Leibler divergence (Kullback and Leibler, 1951) and applied to the Wind Turbine Benchmark (Fogh Odgaard et al., 2009). The residual evaluation approach employed in Svärd and Nyberg (2012) was however not able to fully handle the issue concerning time-varying uncertainties related to model errors and operating modes discussed above. In this work, the automated design methodology is refined by means of the data-driven statistical residual evaluation approach described in Paper C, which indeed is able to handle this issue.

This paper illustrates how an FDI-system for an automotive diesel engine can be designed by application of this automated design methodology. The overall aim, and the main contribution, is to demonstrate how a set of general methods may be combined into a complete methodology in order to solve a real industrial problem, in this case the indeed challenging problem of automotive diesel engine FDI (Nyberg and Stutte, 2004). In this sense, this work serves as an illustration of the state-of-practice in model-based FDI, and in particular sequential residual generation, e.g., Staroswiecki and Declerck (1989); Cassar and Staroswiecki (1997); Staroswiecki (2002); Pulido and Alonso-González (2004); Ploix et al. (2005); Travé-Massuyès et al. (2006); Blanke et al. (2006); Svärd and Nyberg (2010), and statistical residual evaluation, e.g., Willsky and Jones (1976); Gertler (1998); Basseville and Nikiforov (1993); Peng et al. (1997); Al-Salami et al. (2006); Blas and Blanke (2011); Wei et al. (2011). Moreover, as a secondary contribution, the usefulness and properties of the specific methods described in Svärd and Nyberg (2010), Paper B, and Paper C, are illustrated and discussed. For instance, it is empirically shown how the usage of residual generators utilizing both integral and derivative causality, i.e., mixed causality, increases the fault isolability, and how time-varying model errors can be handled in the framework of statistical likelihood-based residual evaluation.

The paper is structured as follows. Section 2 presents the considered automotive diesel engine system and the model of the system used in the design of the FDI-system. Section 3 gives an overview of the different stages in the automated design methodology from a user perspective. The different methods and their key properties are briefly discussed but technical details are kept at a minimum. Full details can be found in Svärd and Nyberg (2010), Paper B, and Paper C. Sections 4 and 5 describe how the automated methodology was applied to the diesel engine system and discuss details and different aspects of the resulting FDI-system. In Section 6, the FDI-system is experimentally evaluated and some final remarks are given in Section 7.

2 Automotive Diesel Engine System

The system considered in this work is a 13-liter six-cylinder Scania truck diesel engine equipped with Exhaust Gas Recirculation (EGR), Variable Geometry Turbochargers (VGT), and intake throttle. A schematic of the system is shown in Figure 1. This section describes the system and the model used in the design of the proposed FDI-system.

2.1 System Description

Consider Figure 1. Air of temperature T_{bc} and pressure p_{bc} enters the system and passes the compressor side of the VGT. The compressed air, with mass-flow W_{c}, then enters the intercooler after which the pressure of the air is denoted p_{ic}. The cooled air then passes the intake throttle, whose position is given by x_{th}, and which is used to control the amount of air entering the intake manifold.

The air mass-flow after the intake throttle is denoted W_{th}, and the pressure and temperature of the air in the intake manifold are denoted p_{im} and T_{im}, respectively. In

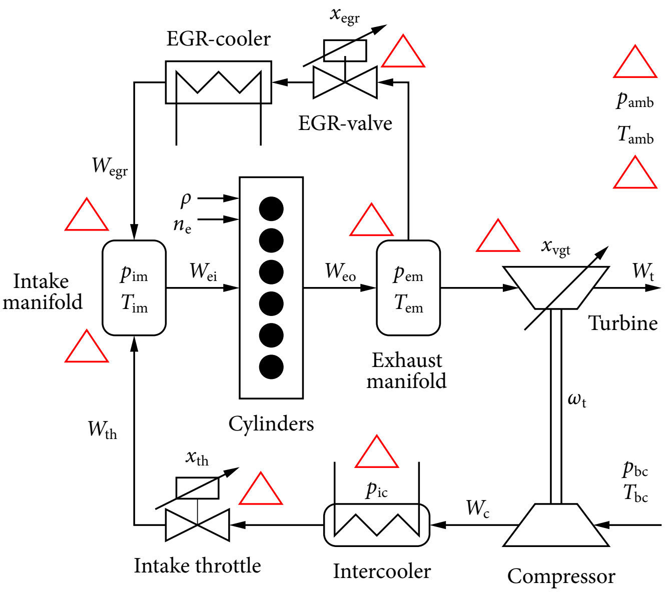


Figure 1: Schematic of the automotive diesel engine system. Locations of considered faults are illustrated with triangles.

the intake manifold, the air is mixed with recirculated exhaust gases, whose mass-flow is denoted W_{egr}, before it enters the cylinders. The amount of recirculated gas is controlled by the EGR-valve, whose position is denoted x_{egr}. The total mass-flow of the gas entering the cylinders is denoted W_{ei}.

In the cylinders, the gas is mixed with fuel and then combusted. The amount of fuel injected into the cylinders is given by ρ, and the rotational speed of the engine is denoted n_{e}. After the combustion, the gas enters the exhaust manifold. The mass-flow of the exhaust gas is denoted W_{eo}, and the pressure and temperature of the gas in the exhaust manifold p_{em} and T_{em}, respectively. The exhaust gas then passes the turbine side of the VGT, whose rotational speed is given by ω_{t}, and leaves the system with mass-flow W_{t}. The geometry of the VGT is controlled with the VGT-valve, whose position is denoted x_{vgt}.

2.2 Sensors and Actuators

The system is equipped with 4 actuators, $u_{x_{\mathrm{th}}}$, $u_{x_{\mathrm{egr}}}$, $u_{x_{\mathrm{vgt}}}$, u_{ρ}, and 7 sensors, $y_{p_{\mathrm{amb}}}$, $y_{T_{\mathrm{amb}}}$, $y_{p_{\mathrm{ic}}}$, $y_{p_{\mathrm{im}}}$, $y_{T_{\mathrm{im}}}$, $y_{p_{\mathrm{em}}}$, $y_{n_{\mathrm{e}}}$. See Table 1 for details.

2.3 Faults

Faults in all sensors and actuators in Table 1, except in actuator u_{ρ} and sensor $y_{n_{\mathrm{e}}}$, are considered. All faults along with their description can be found in Table 2. The

Table 1: Sensors and Actuators.

Signal	Description
$u_{x_{\mathrm{th}}}$	Throttle position actuator
$u_{x_{\mathrm{egr}}}$	EGR-valve position actuator
$u_{x_{\mathrm{vgt}}}$	VGT-valve position actuator
u_{ρ}	Injected fuel actuator
$y_{n_{\mathrm{e}}}$	Engine speed sensor
$y_{p_{\mathrm{amb}}}$	Ambient temperature sensor
$y_{T_{\mathrm{amb}}}$	Ambient pressure sensor
$y_{p_{\mathrm{ic}}}$	Inter-cooler pressure sensor
$y_{p_{\mathrm{im}}}$	Inlet manifold pressure sensor
$y_{T_{\mathrm{im}}}$	Inlet manifold temperature sensor
$y_{p_{\mathrm{em}}}$	Exhaust manifold pressure sensor

approximate locations of the faults are marked with triangles in Figure 1.

Modeling of Faults

The faults are modeled as additive signals in corresponding equations in the nominal model presented in next section. For example, fault $\Delta_{y_{p_{\mathrm{im}}}}$, representing a fault in the intake manifold pressure sensor $y_{p_{\mathrm{im}}}$, is modeled by simply adding $\Delta_{y_{p_{\mathrm{im}}}}$ to the equation describing the relation between the sensor value $y_{p_{\mathrm{im}}}$ and the actual intake manifold pressure p_{im}, i.e., $y_{p_{\mathrm{im}}} = p_{\mathrm{im}} + \Delta_{y_{p_{\mathrm{im}}}}$.

The main argument for using this fault modeling approach is that it is considered to be hard, or even impossible, to know how a faulty component behaves in reality and data for evaluation and validation of a more detailed fault model is seldom available. Moreover, modeling faults in this way also results in a minimum of fault modes, which gives a smaller model. This is beneficial since a smaller model simplifies several steps in model-based diagnosis, for example residual generation or fault isolation. The last but not least argument is simplicity, since extending the nominal model with additive fault signals is straightforward and easy. Nevertheless, the approach has shown to provide good results (Svärd and Nyberg, 2012).

The adopted approach is nonetheless general, and no assumptions are made regarding for example the time-behavior of faults. Note for example that the approach is able to handle multiplicative faults even though the fault signal is assumed to be additive. Consider for example a multiplicative fault in $y_{p_{\mathrm{im}}}$ given by $y_{p_{\mathrm{im}}} = \delta \cdot p_{\mathrm{im}}$, $\delta \neq 1$, which can be equivalently described by $\Delta_{y_{p_{\mathrm{im}}}} = p_{\mathrm{im}} \left(\delta - 1\right)$.

2.4 Model

The model of the automotive diesel engine can be found in Appendix A. The model contains in total 46 equations, 43 unknown variables, 11 known variables, of which 4 are actuators and 7 sensors, and 9 faults. Of the 46 equations, 5 are differential equations

Table 2: Considered Faults.

Fault	Description
$\Delta_{y_{p_{\text{amb}}}}$	Fault, ambient pressure sensor
$\Delta_{y_{T_{\text{amb}}}}$	Fault, ambient temperature sensor
$\Delta_{y_{p_{\text{ic}}}}$	Fault, intercooler pressure sensor
$\Delta_{y_{p_{\text{im}}}}$	Fault, intake manifold pressure sensor
$\Delta_{y_{T_{\text{im}}}}$	Fault, intake manifold temperature sensor
$\Delta_{y_{p_{\text{em}}}}$	Fault, exhaust manifold pressure sensor
$\Delta_{u_{x_{\text{th}}}}$	Fault, throttle position actuator
$\Delta_{u_{x_{\text{egr}}}}$	Fault, EGR-valve position actuator
$\Delta_{u_{x_{\text{vgt}}}}$	Fault, VGT-valve position actuator

and the rest algebraic equations.

The model describes the gas-exchange system of the engine and is described in Wahlström and Eriksson (2011). The model relies on both fundamental first principle physics and gray-box modeling.

Non-Linear Model Equations

Due to the non-linear characteristics of the considered engine system, the model in Appendix A contains several non-linear functions. For instance, the function $\Psi_{\text{th}}^{\gamma_{\text{th}}}(\Pi_{\text{th}})$ found in equation e_7 is given by

$$\Psi_{\text{th}}(\Pi_{\text{th}}) = \begin{cases} \Psi_{\text{th}}^*(\Pi_{\text{th}}) & \text{if } \Pi_{\text{th}} \leq \Pi_{\text{th,lin}} \\ \Psi_{\text{th}}^*(\Pi_{\text{th,lin}})\frac{1-\Pi_{\text{th}}}{1-\Pi_{\text{th,lin}}} & \text{if } \Pi_{\text{th}} > \Pi_{\text{th,lin}} \end{cases}, \tag{1}$$

where

$$\Psi_{\text{th}}^*(\Pi_{\text{th}}) = \sqrt{\frac{2\gamma_{\text{th}}}{\gamma_{\text{th}}-1}\left(\Pi_{\text{th}}^{2/\gamma_{\text{th}}} - \Pi_{\text{th}}^{1+1/\gamma_{\text{th}}}\right)},$$

and $\Pi_{\text{th,lin}}$ and γ_{th} are parameters.

For more details, see Wahlström and Eriksson (2011). For notational simplicity, complicated non-linearities like (1) have in the model given in Appendix A been denoted by functions named in analogy with $\Psi_{\text{th}}^{\gamma_{\text{th}}}(\Pi_{\text{th}})$. For instance, $f_{T_e W_f}(W_f)$ in e_{13} and $\eta_{\text{tm},\omega_t}(\omega_t)$ in e_{18}.

3 Overview of Design Methodology

This section presents an overview of the automated methodology used to design the FDI-system for the automotive diesel engine. The actual methods used in the different design stages are explained and discussed. First, however, a brief description of the structure of the FDI-system is given.

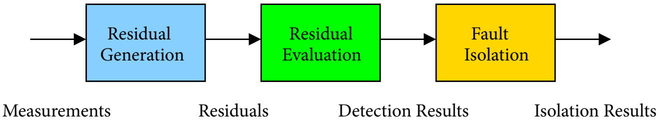


Figure 2: Overview of the FDI-system.

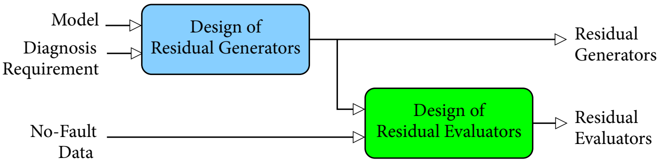


Figure 3: Overview of design methodology.

3.1 Structure of FDI-System

The proposed FDI-system for the engine contains the subsystems: residual generation, residual evaluation, and fault isolation, see Figure 2.

Measured signals, $\mathbf{y}$, in this case from the actuators and sensors listed in Table 1, are used as input to the residual generation block. This block contains a set of residual generators, $R_1, R_2, \ldots, R_n$, each used to monitor a part of the system. The output from the residual generation block is a set of residual signals, $r_1, r_2, \ldots, r_n$, with $r_i = R_i(\mathbf{y})$. The residual signals are used as input to the residual evaluation block, which contains a set of residual evaluators, $T_1, T_2, \ldots, T_n$. The aim of the residual evaluation is to detect changes in the residual signal behavior caused by faults in the system. The output from the residual evaluation block is a set of binary fault detection signals, $d_1, d_2, \ldots, d_n$, with $d_i = T_i(r_i)$. Each d_i indicates if a fault is present or not in the part of the system monitored by the corresponding residual generator R_i. The set of fault detection signals $d_1, d_2, \ldots, d_n$ is finally used as input to the fault isolation block, where they are used to isolate the detected fault(s).

3.2 Automated Design Methodology

An overview of the overall methodology used to design the residual generators and residual evaluators, is shown in Figure 3.

The design methodology depicted in Figure 3 have been developed with the aim to be automated to a high extent and requires limited human interaction. The methodology requires the following input:

- A *model* $M = (E, X, D, Y, F)$ of the system, where E is a set of differential-algebraic equations relating the unknown variables X, differentiated variables D, known variables Y, and fault variables F.

- A *diagnosis requirement* $\mathcal{F}$, given as a set of ordered fault pairs $(\Delta_i, \Delta_j) \in F \times F$. The interpretation of $(\Delta_i, \Delta_j) \in \mathcal{F}$ is that fault Δ_i should be isolable from fault Δ_j.

- No-fault data $\mathcal{Y}$, given in the form of measurements of the variables in Y.

The output is a set of residual generators $R_1, R_2, \ldots, R_n$, and a set of residual evaluators, $T_1, T_2, \ldots, T_n$. The specific methods used to design the residual generators and residual evaluators are described in subsequent sections. Design of the fault isolation subsystem is briefly discussed in Section 5.3.

3.3 Residual Generation

The method used to design the individual residual generators is described in Svärd and Nyberg (2010) and belongs to a class of methods referred to as *sequential residual generation*, based on ideas originally described in Staroswiecki and Declerck (1989). Similar approaches are described and exploited in for example Cassar and Staroswiecki (1997); Staroswiecki (2002); Pulido and Alonso-González (2004); Ploix et al. (2005); Travé-Massuyès et al. (2006); Blanke et al. (2006).

This class of methods has shown to be successful for real applications (Dustegor et al., 2004; Izadi-Zamanabadi, 2002; Cocquempot et al., 1998), and also has the potential to be automated to a high extent (Svärd and Nyberg, 2012). The key property of the specific method described in Svärd and Nyberg (2010) is its ability to handle *mixed causality*, which greatly increases the possibility to detect and isolate faults in large-scale complex models. This issue is discussed and illustrated in Section 4.

In general, it is possible to create thousands of residual generators with the method from (Svärd and Nyberg, 2010) for large models. Regarding implementation aspects such as complexity and computational load it is infeasible, or even impossible, to use all these residual generators in the FDI-system. In addition, it is often possible to meet the stated diagnosis requirement with a small subset of all residual generators. Therefore, the set of residual generators to be contained in the FDI-system is selected by means of a two-step approach, as also elaborated in Nyberg (1999); Krysander (2006); Nyberg and Krysander (2008), which is described next.

Two-Step Approach

Given the model M of the system and the diagnosis requirement $\mathcal{F}$, the two steps illustrated in Figure 4 are conducted. In the first step, a large set of *candidate residual generators*, in the form of subsets of the model equations, is found. This step is done in an exhaustive manner, in the sense that all model equation subsets that can be used as input to the sequential residual generation method (Svärd and Nyberg, 2010) are found. For this particular method, it can be shown (Svärd and Nyberg, 2010) that candidate residual generators by necessity should be based on Minimal Structural Overdetermined (MSO) sets of equations. There exists efficient algorithms for finding all MSO sets, given a model, see, e.g., Krysander et al. (2008).

In general, all candidate residual generators found in the first step are not *realizable*, i.e., it is not possible to create residual generators from all found candidate residual generators with the considered method. Therefore, in the second step, a set of realizable candidate residual generators that fulfills the diagnosis requirement $\mathcal{F}$ are selected and the final set of residual generators $R_1, R_2, \ldots, R_n$ is created.

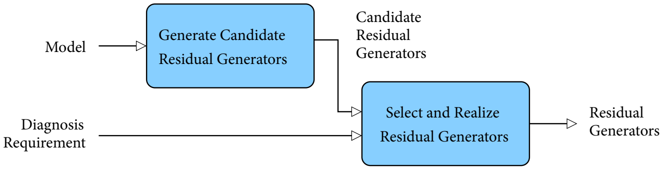

Model
Generate Candidate Residual Generators
Candidate Residual Generators
Diagnosis Requirement
Select and Realize Residual Generators
Residual Generators

Figure 4: Design of residual generators.

Realizability of Candidate Residual Generators

Realizability is a general property of a candidate residual generator, i.e., a set of equations, with respect to a given residual generation method, see Paper B. In the context of the method (Svärd and Nyberg, 2010), a set of of equations $S \subseteq E$ is said to be realizable if it can be written in the form

$$
\begin{aligned}
\dot{\mathbf{z}} &= f\left(\mathbf{z}, \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m, \mathbf{y}\right) && \text{(2a)}\\
\mathbf{w}_1 &= g_1\left(\mathbf{z}, \mathbf{y}\right) && \text{(2b)}\\
\mathbf{w}_2 &= g_2\left(\mathbf{z}, \mathbf{w}_1, \dot{\mathbf{w}}_1, \mathbf{y}\right) && \text{(2c)}\\
&\vdots\\
\mathbf{w}_m &= g_m\left(\mathbf{z}, \mathbf{w}_1, \dot{\mathbf{w}}_1, \mathbf{w}_1, \dot{\mathbf{w}}_2, \dots, \mathbf{w}_{m-1}, \dot{\mathbf{w}}_{m-1}, \mathbf{y}\right) && \text{(2d)}
\end{aligned}
$$

where $\mathbf{z}$ is a vector of differentiated variables, $\mathbf{w}_i$, $i = 1, 2, \dots, m$, vectors of algebraic variables, and $\mathbf{y}$ a vector of known variables. In addition, it is for realizability required that (2) is stable.

A sufficient condition for the ability to transform the equations in S into the form (2), is the existence of a *computation sequence* for the unknown variables contained in $\mathbf{z}$ and $\mathbf{w}_i$, $i = 1, 2, \dots, m$. The existence of a computation sequence depends naturally on the properties of the equations in S, but also on the *causality assumption*, i.e., regarding whether integral and/or derivative causality (Blanke et al., 2006) may be used to handle differential equations in the computation sequence, and a given set of *algebraic equation solving tools*. For further details, see Svärd and Nyberg (2010).

Selection of Residual Generators

Motivated by implementation aspects, it is in the second step desirable to find a minimal cardinality set of realizable residual generators that fulfills the diagnosis requirement $\mathcal{F}$. If the number of found candidate residual generators is large, which typically is the case for large-scale models such as the one considered in this work, the problem of finding such a minimal set of residual generators is hard, or even impossible, to solve optimally. However, by relaxing the minimal cardinality requirement, a near optimal solution to the selection problem can be efficiently computed by means of the greedy residual generator selection algorithm developed in Paper B.

In the greedy selection algorithm, in each iteration given the set of already selected candidate residual generators, the candidate residual generator able to isolate most of the

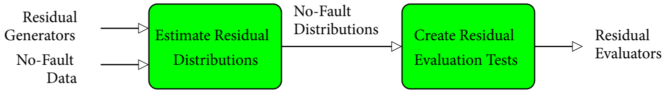


Figure 5: Design of residual evaluators.

not already isolable faults in the given diagnosis requirement $\mathcal{F}$ is selected, and added to the solution if it is realizable. This procedure is repeated until $\mathcal{F}$ is fulfilled, or no useful candidate residual generators remains.

In addition to make the selection problem tractable, the greedy selection algorithm has some additional properties. Specifically, it can be shown (Paper B that if, and only if, the given diagnosis requirement can be fulfilled for the given model with the method (Svärd and Nyberg, 2010), then the algorithm will provide a solution.

3.4 Residual Evaluation

The method used to design residual evaluators is described in Paper C. The key property of this statistical and data-driven method is its ability to handle residuals whose stochastic behavior vary with the current operating mode of the underlying system. The method is based on a comparison of the probability distribution of the residual, estimated online using current data, with a no-fault residual distribution. The no-fault distribution is based on a set of distributions estimated off-line using training data, and is continuously adapted to the current operating mode of the system.

The method used for design of residual evaluators is illustrated in Figure 5. Given a set of residual generators $R_1, R_2, \ldots, R_n$ and no-fault data $\mathcal{Y}$ in the form of measurements of the input to the residual generators, the residual generators are run and no-fault residual samples created. By application of the method developed in Paper C which utilizes K-Means clustering (MacQueen, 1967; Lloyd, 1982), the set of no-fault residual samples is then used to estimate a set θ_i^{NF} of K no-fault distributions for each of the residuals $r_1, r_2, \ldots, r_n$, obtained as output from the residual generators $R_1, R_2, \ldots, R_n$.

Test Statistic

The obtained no-fault residual distributions are then used to create a residual evaluator T_i for each of the residuals $r_1, r_2, \ldots, r_n$. The residual evaluator T_i, with the binary detection signal d_i as output, comprises a fault detection test

$$d_i = T_i\left(\mathcal{R}_i\right) = \begin{cases} 1 & \text{if} \quad \lambda_i\left(\mathcal{R}_i\right) > J_i, \\ 0 & \text{else,} \end{cases} \tag{3}$$

where λ_i is a test statistic, $\mathcal{R}_i$ is a set of discretized samples from residual r_i, and J_i is a constant detection threshold.

The test statistic λ_i in each fault detection test is designed with the method developed in Paper C and based on the Generalized Likelihood Ratio (GLR) test. Given a set

$\mathcal{R}_i$ of samples of the residual r_i, and the matrix θ_i^{NF} containing the estimated no-fault distributions of r_i, the test statistic is given by

$$\lambda_i\left(\mathcal{R}_i\right) = -2\log \frac{\max\limits_{\alpha}\ \mathcal{L}\left(\alpha, \theta_i^{\mathrm{NF}}|\mathcal{R}_i\right)}{\max\limits_{\alpha,\ \theta}\ \mathcal{L}\left(\alpha, \theta|\mathcal{R}_i\right)}, \tag{4}$$

where $\mathcal{L}\left(\alpha, \theta|\mathcal{R}_i\right)$ denotes the likelihood of the parameters α and θ, given the residual samples in $\mathcal{R}_i$. The parameters α and θ fully specify the probability distribution of the samples in $\mathcal{R}_i$. In this sense, the quantity in the denominator of (4) corresponds to the most likely distribution of the samples in $\mathcal{R}_i$, and the quantity in the numerator to the most likely no-fault residual distribution.

Maximum Likelihood Estimations

In Paper C, it is shown that an explicit solution to the maximum likelihood estimation (MLE) problem in the denominator of (4) can be obtained from the normalized histogram of the samples in $\mathcal{R}_i$. The MLE problem in the numerator however needs to be solved numerically. In order to enable implementation of the residual evaluators in an online environment subject to real-time constraints, this problem can be relaxed and posed as a constrained linear least square problem. This problem can be efficiently solved in real-time using methods based on convex optimization (Mattingley and Boyd, 2010). For technical details, see Paper C.

4 Design of Residual Generators

As said in Section 1 it is by OBD-legislations required that emission critical faults in an automotive engine are detected and isolated. For the considered engine, all faults found in Table 2 are emission critical. In addition, if not accommodated in time, the faults in Table 2 may also lead to decreased safety, increased fuel consumption, decreased driveability, or even engine breakdown. The latter indeed reduces vehicle uptime.

Motivated by this, it is required that all faults found in Table 2 can be detected and isolated from each other. Thus, the diagnosis requirement $\mathcal{F}$ for the diesel engine consists of all unique pairwise combinations of the 9 faults in Table 2, i.e.,

$$\mathcal{F} = \left\{\left(\Delta_{y_{p_{\mathrm{amb}}}}, \Delta_{y_{T_{\mathrm{amb}}}}\right), \left(\Delta_{y_{p_{\mathrm{amb}}}}, \Delta_{y_{p_{\mathrm{ic}}}}\right), \dots, \right\} \tag{5}$$

with $|\mathcal{F}| = 9 \times 9 - 9 = 72$.

4.1 Candidate Residual Generators

The model of the engine given in Appendix A together with the diagnosis requirement $\mathcal{F}$, were used as input to a Matlab implementation of the two-step residual generation methodology outlined in Section 3.3 and Figure 4.

In total 14, 242 candidate residual generators could be found for the engine model. These are based on 270 MSO sets, found using the algorithm described in Krysander et al.

(2008). An MSO set by definition contains one more equation than unknown variables. Given an MSO set, a sequential residual generator is created by removing one equation and then finding a computation sequence for the unknown variables in the remaining just-determined set of equations. The number of candidate residual generators that can be created from a single MSO set thus equals the number of equations in the MSO set. This is the rationale behind the number of 14, 242 candidate residual generators.

4.2 Residual Generator Selection and Realization

The algorithm (Svärd and Nyberg, 2010) for finding computation sequences for the candidate residual generators was configured to allow both integral and derivative causality, i.e., mixed causality, and also to use Maple as algebraic equation solving tool, see Section 3.3.

Using the greedy selection algorithm (Paper B) described in 3.3, 8 residual generators, $R_1, R_2, \ldots, R_8$, were selected and realized. For instance, the residual generator R_3 has the form

$$\begin{aligned}
\dot{\omega}_{\mathrm{t}} &= \frac{P_{\mathrm{t}}\eta_m - P_{\mathrm{c}}}{J_{\mathrm{t}}\omega_{\mathrm{t}}} && \text{(6a)}\\
\dot{T}_{\mathrm{em}} &= \frac{R_{\mathrm{e}} T_{\mathrm{em}}}{p_{\mathrm{em}} V_{\mathrm{em}} c_{\mathrm{ve}}} \left(W_{\mathrm{in}} c_{\mathrm{ve}} \left(T_{\mathrm{em,in}} - T_{\mathrm{em}} \right) + R_{\mathrm{e}} \left(T_{\mathrm{em,in}} W_{\mathrm{in}} - T_{\mathrm{em}} W_{\mathrm{out}} \right) \right) && \text{(6b)}\\
\dot{p}_{\mathrm{em}} &= \frac{R_{\mathrm{e}} T_{\mathrm{em}}}{V_{\mathrm{em}}} \left(W_{\mathrm{eo}} - W_{\mathrm{egr}} - W_{\mathrm{t}} + \Delta_{W_{\mathrm{em}}} \right) + \frac{R_{\mathrm{e}}}{V_{\mathrm{em}} c_{\mathrm{ve}}} (W_{\mathrm{in}} c_{\mathrm{ve}} \left(T_{\mathrm{em,in}} - T_{\mathrm{em}} \right) && \text{(6c)}\\
&\quad + R_{\mathrm{e}} \left(T_{\mathrm{em,in}} W_{\mathrm{in}} - T_{\mathrm{em}} W_{\mathrm{out}} \right))\\
p_{\mathrm{amb}} &= y_{p_{\mathrm{amb}}} && \text{(6d)}\\
p_{\mathrm{bc}} &= p_{\mathrm{amb}} && \text{(6e)}\\
x_{\mathrm{vgt}} &= u_{x_{\mathrm{vgt}}} && \text{(6f)}\\
&\vdots\\
T_{\mathrm{em,in}} &= T_{\mathrm{amb}} + \left(T_{\mathrm{e}} - T_{\mathrm{amb}} \right) \exp\left(-\frac{h_{tot}\pi d_{pipe} l_{pipe} n_{pipe}}{W_{\mathrm{eo}} c_{\mathrm{pe}}}\right) && \text{(6g)}\\
W_{\mathrm{egr}} &= \frac{\left(\dot{p}_{\mathrm{im}} V_{\mathrm{im}} - R_{\mathrm{a}} T_{\mathrm{im}} W_{\mathrm{th}} + W_{\mathrm{ei}} R_{\mathrm{a}} T_{\mathrm{im}} \right)}{R_{\mathrm{a}} T_{\mathrm{im}}} && \text{(6h)}\\
&\vdots\\
P_{\mathrm{c}} &= \frac{W_{\mathrm{c}} c_{pa} T_{\mathrm{bc}}}{\eta_{\mathrm{c}}} \left(\Pi_{\mathrm{c}}^{1-1/\gamma_{\mathrm{a}}} - 1 \right), && \text{(6i)}
\end{aligned}$$

with the *residual equation* $r = y_{p_{\mathrm{em}}} - p_{\mathrm{em}}$, corresponding to equation e_{43} in Appendix A.

Clearly, the structure of residual generator R_3 is in accordance with (2). Moreover, it is noted that residual generator R_3 exploits mixed causality. Integral causality is for example used in (6b) when variable T_{em} is computed. Derivative causality is employed when variable W_{egr} is computed in (6h), since $\dot{p}_{\mathrm{im}}$, the derivative of p_{im}, is used.

The use of derivative causality in general assumes that derivatives of known or previously computed variables can be computed or estimated. In this work, estimation of

Table 3: Fault Signature Matrix.

	$\Delta y_{p_{\mathrm{amb}}}$	$\Delta y_{T_{\mathrm{amb}}}$	$\Delta y_{p_{\mathrm{ic}}}$	$\Delta y_{p_{\mathrm{im}}}$	$\Delta y_{T_{\mathrm{im}}}$	$\Delta y_{p_{\mathrm{em}}}$	$\Delta u_{x_{\mathrm{th}}}$	$\Delta u_{x_{\mathrm{egr}}}$	$\Delta u_{x_{\mathrm{vgt}}}$
R_1	x	x		x	x	x		x	x
R_2	x	x	x	x		x	x	x	x
R_3	x	x	x	x	x	x	x		x
R_4	x	x	x	x	x		x	x	x
R_5	x		x	x	x	x	x	x	x
R_6	x	x	x		x	x	x	x	x
R_7	x	x	x	x	x	x		x	x
R_8	x	x		x	x	x	x	x	x

derivatives is done by appliance of a low-pass FIR-filter with coefficients calculated according to Vainio et al. (1997). This approach was used since it is simple and straightforward to implement, and gave good results.

The use of integral causality presupposes that ordinary differential equations can be solved, which in general assumes that consistent initial conditions for the state-variables are available. There are 5 different state variables present in the set of selected residual generators: the intake manifold pressure p_{im}, the exhaust manifold pressure p_{em}, intercooler pressure p_{ic}, the exhaust manifold temperature T_{em}, and the turbine speed ω_{t}. As seen in Table 1, the three pressures are measured directly. Thus, the values of the corresponding measured variables at the starting time instant are used as initial conditions for these variables, e.g., $p_{\mathrm{im}}(t_0) = y_{p_{\mathrm{im}}}(t_0)$. For the non-measured state-variable T_{em}, the initial condition is set to the value of the measured inlet air temperature $y_{T_{\mathrm{im}}}$ at the starting time instant. The initial condition for the state-variable ω_{t} is set to a constant nominal value.

Fault Detectability

Table 3 shows the *fault signature matrix* (FSM) for the 8 selected residual generators with respect to the faults in Table 2. In this context, the FSM contains an "x" in position (R_i, Δ_x) if the equation containing fault Δ_x is used in the computation sequence on which the residual generator R_i is based. This should be interpreted as that residual generator R_i may be *sensitive* to fault Δ_x, meaning that it may respond to the fault. The sensitivity of residual generator R_i to the fault Δ_x however strongly depends on the properties of R_i, the size and temporal properties of Δ_x, and also on for example the current operating mode of the system. In order to verify that R_i is indeed sensitive to Δ_x, it is necessary to implement and run R_i using representative data from relevant fault cases. This will be done in Section 6.

Clearly, assuming that Table 3 reflects the fault sensitivity, there is more than one residual generator that is sensitive to each of the 9 considered faults and thus all 9 faults can, in theory, be detected.

Table 4: Isolability Matrix.

	$\Delta_{y_{p_{\text{amb}}}}$	$\Delta_{y_{T_{\text{amb}}}}$	$\Delta_{y_{p_{\text{ic}}}}$	$\Delta_{y_{p_{\text{im}}}}$	$\Delta_{y_{T_{\text{im}}}}$	$\Delta_{y_{p_{\text{em}}}}$	$\Delta_{u_{x_{\text{th}}}}$	$\Delta_{u_{x_{\text{egr}}}}$	$\Delta_{u_{x_{\text{vgt}}}}$
$\Delta_{y_{p_{\text{amb}}}}$	x								x
$\Delta_{y_{T_{\text{amb}}}}$	x	x							x
$\Delta_{y_{p_{\text{ic}}}}$	x		x						x
$\Delta_{y_{p_{\text{im}}}}$	x			x					x
$\Delta_{y_{T_{\text{im}}}}$	x				x				x
$\Delta_{y_{p_{\text{em}}}}$	x					x			x
$\Delta_{u_{x_{\text{th}}}}$	x						x		x
$\Delta_{u_{x_{\text{egr}}}}$	x							x	x
$\Delta_{u_{x_{\text{vgt}}}}$	x								x

Fault Isolability

In general, given a set of residual generators, a fault Δ_x is said to be *isolable* from a fault Δ_y if the set contains a residual generator that is sensitive to fault Δ_x but not to fault Δ_y, see for example Paper B. As seen in Table 3, all 8 residual generators may be sensitive to the faults $\Delta_{y_{p_{\text{amb}}}}$ and $\Delta_{u_{x_{\text{vgt}}}}$. This is also indicated in Table 4, which shows the resulting *isolability matrix* for the 8 selected residual generators. In Table 4, for instance, the "x" in position $\left(\Delta_{y_{T_{\text{amb}}}}, \Delta_{y_{p_{\text{amb}}}}\right)$ denotes that fault $\Delta_{y_{T_{\text{amb}}}}$ is not isolable from fault $\Delta_{y_{p_{\text{amb}}}}$ using the residual generators $R_1, R_2, \ldots, R_8$.

Clearly, according to Table 4, the diagnosis requirement $\mathcal{F}$ in (5) not is met since, for example, $\Delta_{y_{T_{\text{amb}}}}$ not is isolable from $\Delta_{y_{p_{\text{im}}}}$. Nevertheless, due to the properties of the greedy selection algorithm discussed in Section 3.3, Table 4 shows the maximum attainable isolability for the engine model, given the method for residual generation considered in this work. The cardinality of the set of selected residual generators may however not be minimal. See Paper B for more details.

4.3 Properties of Selected Residual Generators

Some additional properties for the 8 selected residual generators can be found in Table 5. The first column in Table 5 shows which residual equation the corresponding residual generator uses, i.e., which model equation that is used to compute the residual in the corresponding residual generator. It can be noted that a majority of the 8 residual generators use either equation e_{39}, or equation e_{41}, as residual equation, corresponding to $r = y_{p_{\text{im}}} - p_{\text{im}}$ and $r = y_{p_{\text{em}}} - p_{\text{em}}$, respectively. This is a direct consequence of that the greedy selection algorithm was supplemented with an additional heuristic in order to make the final deployment of the residual generators as simple as possible. In those cases when the greedy heuristic described in Section 3.3 identified more than one candidate, the algorithm was configured to prefer small candidate residual generators, in terms of number of equations, before large candidate residual generators, and also to prefer candidate residual generators using sensor equations, i.e., $e_{36}, e_{37}, \ldots, e_{41}$, as residuals.

Table 5: Properties of the Selected Residual Generators.

	Residual	IC	DC	#Equations	#Inputs
R_1	e_{41}	x	x	42 (5)	9
R_2	e_7	x	x	43 (5)	10
R_3	e_{41}	x	x	43 (4)	10
R_4	e_{39}	x		44 (4)	10
R_5	e_{39}	x	x	44 (4)	10
R_6	e_{41}	x		44 (4)	10
R_7	e_{41}	x		41 (3)	10
R_8	e_{39}	x	x	43 (5)	10

Columns 2 and 3 in Table 5 show if the corresponding residual generator uses integral causality (IC) and/or derivative causality (DC), respectively. Clearly, 5 out of 8 residual generators employs mixed causality. Column 4 shows the number of equations contained in the computation sequence on which the corresponding residual generator is based, and the value in parenthesis how many of those equations that are differential equations. Recalling that the model contains in total 46 equations, of which 5 are differential equations, it can be concluded that all residual generators uses a substantial part of the complete model in spite of the above mentioned heuristic. This issue is further illustrated by column 5 in Table 5, which shows how many of the 11 available signals in Table 1 that each residual generator uses as input.

Columns 4 and 5 explain why most of the 8 selected residual generators may be sensitive to most of the 12 faults, as illustrated in Table 3. In fact, this property holds for all candidate residual generators which on average use about 40 equations, and is a direct consequence of the properties of the automotive engine system. Specifically, the system contains many physical interconnections, for example due to the shaft connecting the turbine and the compressor and thus the intake and the exhaust parts of the engine, see Figure 1. This leads to a model with coupled equations, in the sense that there are sets of equations containing the same set of unknown variables. This fact implies that a fault affecting one of these equations influences a large amount of the other model equations. This fact, in combination with the relatively small number of sensors, makes fault decoupling non-trivial and results in the situation shown in Table 3.

4.4 Comments on Realizability

The results presented above were obtained using mixed causality, i.e., computation sequences with both integral and derivative causality were allowed. For comparison, the algorithm (Svärd and Nyberg, 2010) for finding computation sequences was also configured to use solely integral and derivative causality. For the case with derivative causality, no realizable candidate residual generator were found. In the integral causality case, a set of 4 residual generators was selected. In fact, two of these residual generators were also found when adopting mixed causality and can be found as R_6 and R_7 in Table 5.

Before termination, the greedy selection algorithm discarded in total 4,739 of the

Table 6: Isolability Matrix when using Integral Causality.

	$\Delta_{y_{p_{\mathrm{amb}}}}$	$\Delta_{y_{T_{\mathrm{amb}}}}$	$\Delta_{y_{p_{\mathrm{ic}}}}$	$\Delta_{y_{p_{\mathrm{im}}}}$	$\Delta_{y_{T_{\mathrm{im}}}}$	$\Delta_{y_{p_{\mathrm{em}}}}$	$\Delta_{u_{x_{\mathrm{th}}}}$	$\Delta_{u_{x_{\mathrm{egr}}}}$	$\Delta_{u_{x_{\mathrm{vgt}}}}$
$\Delta_{y_{p_{\mathrm{amb}}}}$	X	X			X			X	X
$\Delta_{y_{T_{\mathrm{amb}}}}$	X	X			X			X	X
$\Delta_{y_{p_{\mathrm{ic}}}}$	X	X	X		X			X	X
$\Delta_{y_{p_{\mathrm{im}}}}$	X	X		X	X			X	X
$\Delta_{y_{T_{\mathrm{im}}}}$	X	X			X			X	X
$\Delta_{y_{p_{\mathrm{em}}}}$	X	X			X	X		X	X
$\Delta_{u_{x_{\mathrm{th}}}}$	X	X			X		X	X	X
$\Delta_{u_{x_{\mathrm{egr}}}}$	X	X			X			X	X
$\Delta_{u_{x_{\mathrm{vgt}}}}$	X	X			X			X	X

14,242 candidate residual generators for not being realizable in the mixed causality case, and 7,133 candidates in the integral causality case. The corresponding numbers in terms of MSO sets are 91 and 135, respectively, out of 270. In the derivative causality case, apparently, all candidate residual generators were discarded due to non-realizability. It can be concluded that mixed causality improves realizability, in the sense that considerably more candidate residual generators can be realized, which implies that more faults can be isolated. This can be seen by comparing Table 4 with Table 6, which shows the resulting isolability matrix when using only integral causality.

The large amount of discarded candidate residual generators, independent on the causality assumption, is due to that no computation sequence can be found for these candidate residual generators. This in turn is to a large extent caused by non-invertible non-linear functions in the model. To illustrate this aspect, consider the equation

$$e_7: \quad W_{\mathrm{th}} = \frac{p_{\mathrm{ic}} A_{\mathrm{th,max}}}{\sqrt{T_{\mathrm{im}} R_{\mathrm{a}}}} \Psi_{\mathrm{th}}^{\gamma_{\mathrm{th}}}(\Pi_{\mathrm{th}}) f_{\mathrm{th}}(x_{\mathrm{th}}),$$

where W_{th}, p_{ic}, T_{im}, Π_{th}, and x_{th} are unknown variables, $A_{\mathrm{th,max}}$ and R_{a} are parameters, and $\Psi_{\mathrm{th}}^{\gamma_{\mathrm{th}}}(\cdot)$ and $f_{\mathrm{th}}(\cdot)$ are non-linear functions, with $\Psi_{\mathrm{th}}^{\gamma_{\mathrm{th}}}(\Pi_{\mathrm{th}})$ given by (1). Clearly, the function $\Psi_{\mathrm{th}}^{\gamma_{\mathrm{th}}}(\Pi_{\mathrm{th}})$ is not invertible with respect to Π_{th} which implies that the variable Π_{th} can not be computed from the equation e_7. The same holds for the variable x_{th}, since the function $f_{\mathrm{th}}(\cdot)$ is non-invertible with respect to x_{th}. This implies that only the variables W_{th}, p_{ic}, and T_{im}, can be computed from equation e_7. Most of the equations in the diesel engine model exhibit this property, and this substantially limits how unknown variables in the model can be computed, which in turn explains the large amount of non-realizable, and thus discarded, candidate residual generators.

Stability Analysis

In comparison, only a fraction of the discarded candidate residual generators were discarded due to not being stable. Nevertheless, the stability analysis is an important part of the realization algorithm since stability is an important property in order to guarantee good dynamical behavior of residual generators. In fact, the considered diesel engine system exhibit a non-minimum phase behavior, see Wahlström and Eriksson (2011) for an analysis regarding this, which imply that there indeed are unstable candidate residual generators.

For sake of simplicity, combined with the urge to be able to conduct the stability analysis in an automated manner with a minimum of user input, the stability analysis is based on linearization. In each of 20 different equilibrium points, the non-linear residual generator obtained from the series of computations described by the corresponding computation sequence, is first linearized. If any of the eigenvalues of the linearized residual generator is greater or equal to zero in any of the 20 equilibrium points, the residual generator is discarded.

The 20 equilibrium points correspond to stationary operating points of the engine, parameterized by the injected fuel amount, u_δ, and engine speed, u_{n_e}. The linearization is done by finite difference approximation. Although the adopted stability analysis approach is simple, it is able to discard the residual generators that were observed to be unable to use due to instability. This has been verified through extensive experimental evaluations.

5 Design of Residual Evaluators

As said in Section 3.4, the first step in the residual evaluator design method is to estimate the probability distributions of the residuals $r_1, r_2, \ldots, r_8$ obtained as output from the residual generators $R_1, R_2, \ldots, R_8$, given the no-fault data set $\mathcal{Y}$.

5.1 Estimation of No-Fault Residual Distributions

To capture the behavior of the residuals in a variety of the operating modes of the diesel engine system, the no-fault data set $\mathcal{Y}$ was formed from two data sets of different characteristics. The first data set is about half an hour long and contains engine test-bed measurements from a World Harmonized Transient Cycle (WHTC) test cycle. The second data set is approximately 2 hours long and contains measurements from a part of a test drive in the south of Sweden, including both city and high-way driving. To reduce the risk of over-fit, the data sets were split into an estimation data set and a validation data set, of equal size. The data was sampled at a rate of 100 Hz, and consequently the estimation and validation data sets contain approximately 450,000 samples, each.

The 8 residual generators were run off-line using the measurements in $\mathcal{Y}$ as input to obtain no-fault residual samples. A set of samples from residuals r_5 is shown in Figure 6. Note the non-ideal behavior of the residual caused by uncertainties, mainly model errors of time-varying nature and magnitude, mentioned in Section 1.

Using a Matlab implementation of the algorithm in Paper C, a set θ^{NF} of $K = 20$ probability density functions were estimated for each residual, see Section 3.4. Figure 7 shows the 20 estimated no-fault residual distributions for the residual r_5 obtained as output from residual generator R_5.

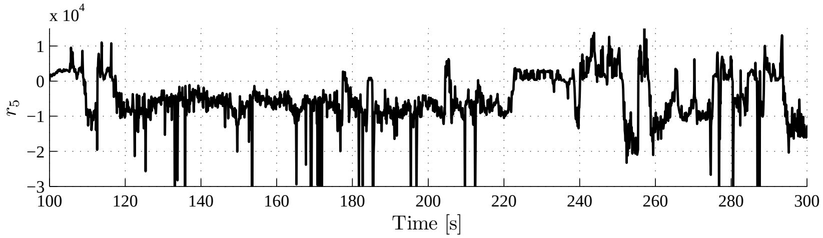


Figure 6: A subset of no-fault samples from residual r_5.

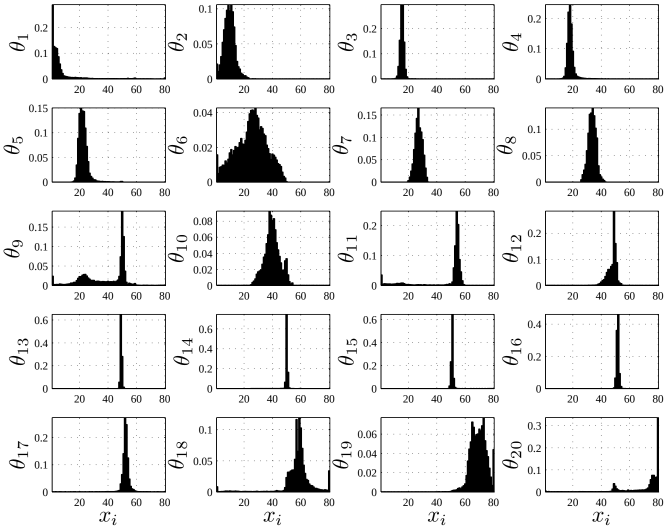


Figure 7: The set of 20 estimated no-fault distributions for residual r_5.

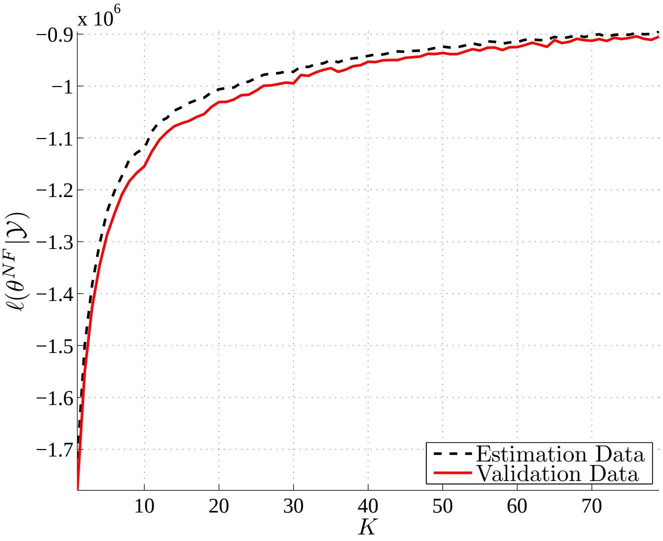


Figure 8: Fit of the set of estimated no-fault distributions for different values of K, i.e., for different number of distributions in the set, to the estimation and validation data sets. The figure shows the average of the fit for all 8 residuals.

For this application, 20 distributions per residual is a good trade-off between model fit and complexity since the gain in model fit obtained when choosing a higher number is marginal in comparison with the corresponding increase in computational effort. This is illustrated in Figure 8, which shows the model fit in the form of the log-likelihood $\ell\big(\theta^{\mathrm{NF}}|\mathcal{Y}\big)$ of the distributions in θ^{NF} given the no-fault data $\mathcal{Y}$. The quantity shown in Figure 8 is the averaged model fit for all 8 residuals, evaluated for different number of distributions and for both the estimation and validation data.

5.2 Residual Evaluators

For each of the residuals $r_1, r_2, \ldots, r_8$, a residual evaluator T_i in the form (3) was created. The sampling of residual values for the sets $\mathcal{R}_i$, $i = 1, 2, \ldots, 8$, was done by means of a sliding window. The number of samples in each sliding window was chosen to be 1024. The choice of this number is a trade-off between detection performance and computational complexity. For a thorough discussion of this issue, see Paper C.

To solve the relaxed version of MLE problem in the numerator of (4), see Section 3.4, a tailored solver was generated using the software tool CVXGEN (Mattingley and Boyd, 2012). The detection thresholds J_i, $i = 1, 2, \ldots, 8$, were computed in order to give a probability of false detection of 1%, by using the validation data set used in Section 5.1.

5.3 Fault Isolation Strategy

As illustrated in Figure 2, the binary fault detection signals based on the residual evaluators (3), are used as input to the fault isolation block. This section briefly describes the strategy used for fault isolation.

Due to the issue regarding fault sensitivity discussed in Section 4.2, and since the complete behavior of the no-fault residuals not are captured by the estimated no-fault distributions, the statistical power of the fault detection tests in (3) are not ideal. That is, the probability for detection is not one for all faults, in all situations, and the probability for false detections is not always zero. To take this into account, the fault isolation scheme is configured to interpret an "x" in a certain row of the FSM in Table 3 as if the test in the corresponding residual evaluator may respond, if the corresponding fault occurs. Consequently, no conclusion is drawn if a residual evaluator not alarms, see Nyberg (1999).

Given a set of alarming residual evaluators, i.e., non-zero detection signals d_i, the fault signatures of the corresponding residuals are matched using the FSM in Table 3. For an example, if only $d_1 = 1$, the row corresponding to R_1 in Table 3 is considered and it is concluded that either of the faults $\Delta_{y_{p_{\mathrm{amb}}}}$, $\Delta_{y_{T_{\mathrm{amb}}}}$, $\Delta_{y_{p_{\mathrm{im}}}}$, $\Delta_{y_{T_{\mathrm{im}}}}$, $\Delta_{y_{p_{\mathrm{em}}}}$, $\Delta_{u_{x_{\mathrm{egr}}}}$, and $\Delta_{u_{x_{\mathrm{vgt}}}}$, may be present in the system. If also the detection signals d_2, d_3, d_4, d_5, d_7, and d_8, are non-zero, it is concluded that either of the faults $\Delta_{y_{p_{\mathrm{im}}}}$, $\Delta_{y_{p_{\mathrm{amb}}}}$, and $\Delta_{u_{x_{\mathrm{vgt}}}}$, may be present. This is in accordance with standard consistency-based diagnosis, see, e.g., de Kleer and Williams (1987); Reiter (1987); Greiner et al. (1989).

6 Experimental Evaluation

This section presents an experimental evaluation of the designed FDI-system. The evaluation consists of two parts, with different purposes. The first part, presented in Section 6.1, focus on the fault detection performance of the individual residual generators and residual evaluators, whereas the second part, presented in Section 6.2, focus on the detection and isolation performance of the complete FDI-system.

6.1 Fault Detection Performance

The purpose of this part of the evaluation is to investigate the fault detection performance of the individual fault detection tests, comprised of the residual generators along with their corresponding residual evaluators.

Metrics

The fault detection performance is studied by means of the statistical power of the fault detection tests, for different sizes of the considered faults in Table 2. To quantify the power of a test, the *power function* (Casella and Berger, 2001) will be used. In this context, the power function for the fault detection test (3) for residual r_i is defined as

$$\beta_i(\delta) = \Pr(d_i = 1|\delta) = \Pr(\lambda_i(\mathcal{R}_i) > J_i|\delta), \tag{7}$$

where λ_i is the test statistic, $\mathcal{R}_i$ a set of samples from residual r_i, J_i is the detection threshold, and δ is a fixed fault size. In the no-fault case, i.e., when δ corresponds to a fault of size zero, the power function (7) gives the probability of false detection, or Type I error (Casella and Berger, 2001). Otherwise, the power function gives the probability of detection for fixed δ, or equivalently the probability of missed detection or Type II error, by $1 - \beta_i(\delta)$.

In order to obtain a scalar metric for the detection performance of a specific detection test with respect to a set D of different fault sizes, the quantity

$$\frac{1}{|D|} \sum_{\delta \in D} \beta_i(\delta), \tag{8}$$

will also be considered, where $\beta_i(\delta)$ is the power function for detection test i. The quantity (8) in some sense reflects the average detection performance of the detection test. It may be noted that for an ideal test, i.e., whose probability for detection is one for all fault sizes, the quantity (8) is equal to one.

Setup

In total 5 data sets were used in the evaluation. The data is not the same as the data described in Section 5. Each data set contains measurements collected during a drive on the Swedish west coast. The data sets contain measurements from in total approximately 2.5 hours of driving, and includes both high-way and city driving under different conditions.

The considered fault type is gain fault. In the case of for example sensor fault $\Delta_{y_{p_{\mathrm{amb}}}}$, this means that the sensor signal $y_{p_{\mathrm{amb}}}$ fed to the residual generators is $y_{p_{\mathrm{amb}}} = \delta \cdot p_{\mathrm{amb}}$ where $\delta \neq 1$ indicates a fault. The gain faults were implemented off-line by modification of the corresponding sensor or actuator measurement signals.

Behaviors of Residuals and Test Statistics

Before presenting quantitative results by means of the metrics (7) and (8) some qualitative results are presented in order to provide some insight of the properties of the residuals and test statistics on which the fault detection tests are based.

Figure 9 shows the residuals $r_1, r_2, \ldots, r_8$ and test statistics $\lambda_1, \lambda_2, \ldots, \lambda_8$ when fault $\Delta_{y_{p_{\mathrm{ic}}}}$ of size $\delta = 1.2$ is abruptly injected at time $t = 700$ s. Figure 10 shows the residuals and test statistics when fault $\Delta_{u_{x_{\mathrm{th}}}}$ of size $\delta = 0.3$ is injected at time $t = 700$ s.

First of all, it is noted that the residuals in Figures 9a and 10a are all non-zero in both the no-fault and fault cases. In addition, all residuals exhibit non-stationary behaviors. It is clear that a conventional residual evaluation approach by means of for example constant thresholding would not be sufficient for these residuals. Moreover, consider for instance residual r_5 in Figure 10a whose response to the fault is quite subtle, in the sense that the behavior of the residual before and after the fault injection is similar. Nevertheless, the test statistic λ_5 clearly indicates the presence of a fault.

According to the FSM in Table 3, residuals r_1 and r_8 may be sensitive to fault $\Delta_{y_{p_{\mathrm{ic}}}}$. This is hard to deduce from Figure 9a, but evident in Figure 9b since all test statics but λ_1

Table 7: Averaged Power for all Tests and all Faults.

	$\Delta y_{p_{\mathrm{amb}}}$	$\Delta y_{T_{\mathrm{amb}}}$	$\Delta y_{p_{\mathrm{ic}}}$	$\Delta y_{p_{\mathrm{im}}}$	$\Delta y_{T_{\mathrm{im}}}$	$\Delta y_{p_{\mathrm{em}}}$	$\Delta u_{x_{\mathrm{th}}}$	$\Delta u_{x_{\mathrm{egr}}}$	$\Delta u_{x_{\mathrm{vgt}}}$	
T_1	**.01**	.03	**0**	.17	.18	.74	**.01**	.05	.11	.22
T_2	.38	.06	.62	.32	**.01**	.38	.09	**.01**	.11	.28
T_3	.07	.05	.75	.40	.07	.25	.06	**.01**	.16	.23
T_4	**.01**	**0**	.83	.47	**.02**	**0**	.11	**0**	**.02**	.49
T_5	**.01**	**0**	.75	.53	.12	.44	.16	**.01**	**0**	.41
T_6	.09	.10	.81	**.01**	.40	.86	.11	.06	.34	.35
T_7	**.01**	.04	**0**	.19	.13	.39	**.01**	.07	.12	.16
T_8	.65	.41	**.02**	.45	.59	.84	.12	.05	.33	.43
	.31	.12	.76	.36	.26	.56	.11	.07	.20	

and λ_8 respond clearly to the injected fault. However, the test statistic λ_7 do not cross the detection threshold. It is noted that this indeed corresponds to a typical situation and is taken into account in the fault isolation scheme, see Section 5.3. It may be noted that a traditional column matching approach (Gertler, 1998) not is sufficient for this, typical, case.

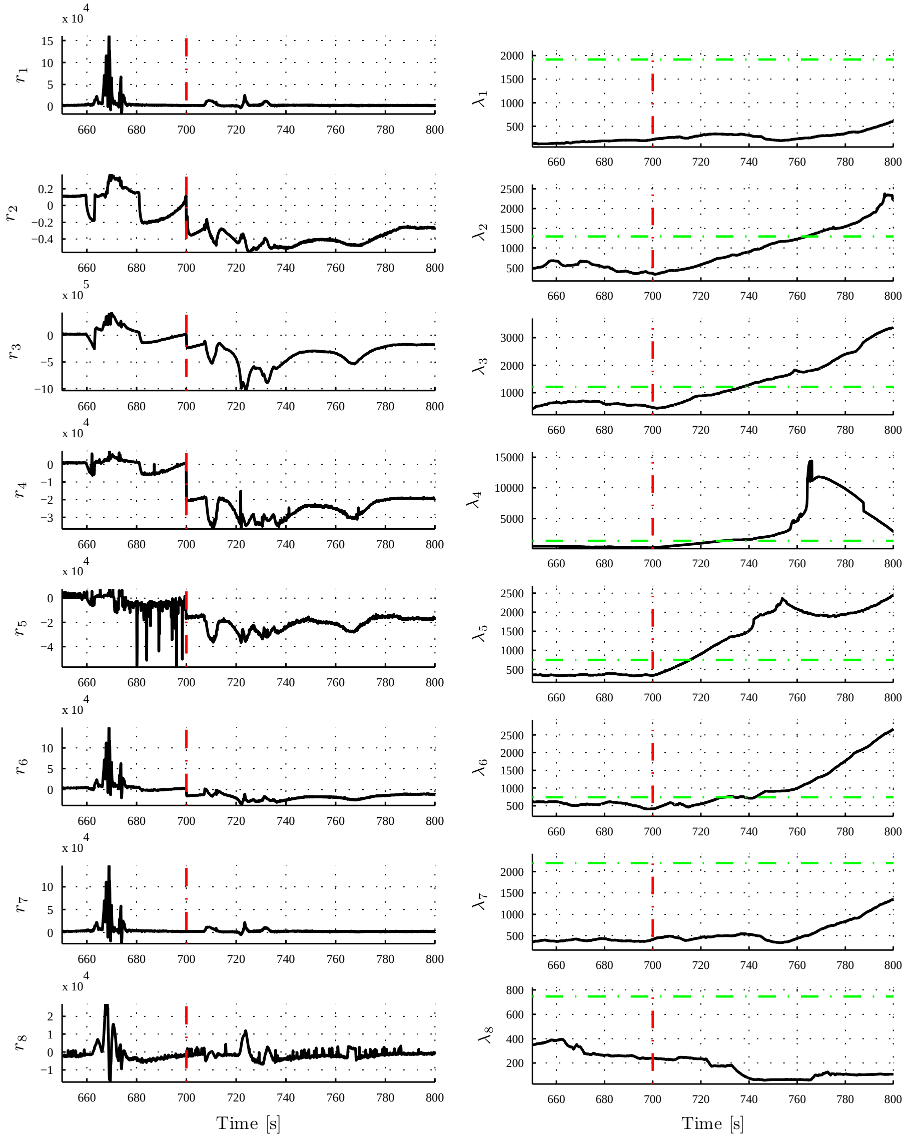
For the fault $\Delta_{u_{x_{\mathrm{th}}}}$, Table 3 states that residuals r_1 and r_7 should not be sensitive to the fault. Again, this is hard to tell from Figure 10a but Figure 10b clearly shows that test statistics λ_1 and λ_7 do not respond to the fault. As also seen in Figure 10b, the response from the test statistic λ_3 is weak and it only barely crosses the detection threshold. The responses from test statistics λ_2 and λ_8 are even weaker and they do not cross the detection thresholds at all. This issue will be further discussed in Sections 6.1 and 6.2.

Results and Comments

Table 7 shows the quantity (8) for the fault detection test based on the residual evaluators $T_1, T_2, \ldots, T_8$ and all faults in Table 2. Entries close to zero, specifically ≤ 0.02, have been marked bold. The right most column gives the average of each row, with the bold entries removed, and the same holds for the last row, but instead for the columns. Figure 11 explicitly shows the estimated power functions $\beta_1, \beta_2, \ldots, \beta_8$ for the faults $\Delta_{y_{p_{\mathrm{ic}}}}$ and $\Delta_{u_{x_{\mathrm{th}}}}$. The power functions were estimated by means of the fraction of samples for which the corresponding test alarmed, i.e., where $d_i = 1$.

As seen in both Figure 11 and Table 7, the powers of all tests are not ideal for all faults and all fault sizes. For example, some tests, e.g., T_2, respond only to sizes $\delta > 1$ for some faults, and only to sizes $\delta < 1$ for other faults. However, for instance fault $\Delta_{y_{p_{\mathrm{ic}}}}$ result in nice test power for almost all tests.

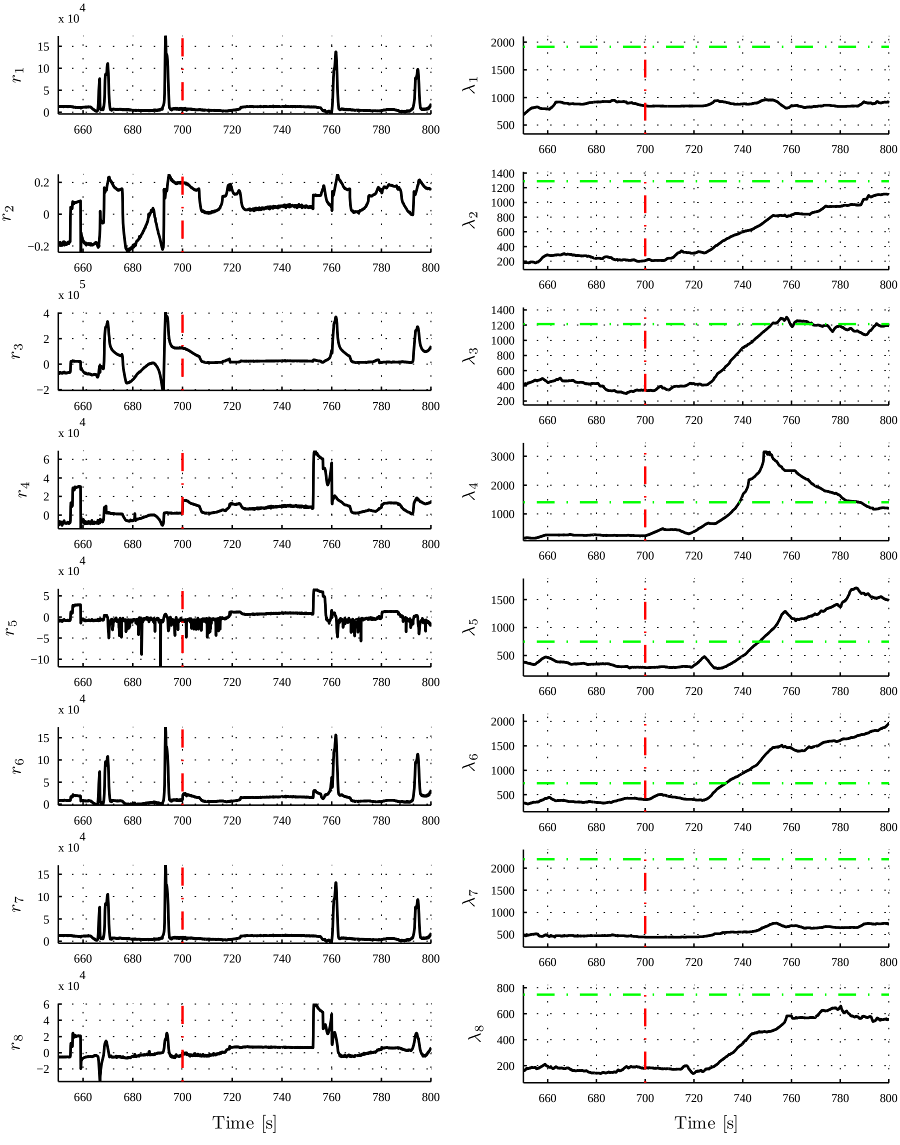
By considering the right most column in Table 7, it can be deduced that the average fault detection performances for all tests are comparable, but that tests T_4, T_5, T_6, and T_8, seem to be slightly better than the other tests. By considering the last row in Table 7, it can be deduced that the pressure sensor faults, $\Delta_{y_{p_{\mathrm{ic}}}}$, $\Delta_{y_{p_{\mathrm{im}}}}$, and $\Delta_{y_{p_{\mathrm{em}}}}$, seem to result



(a) Residuals

(b) Test Statistics

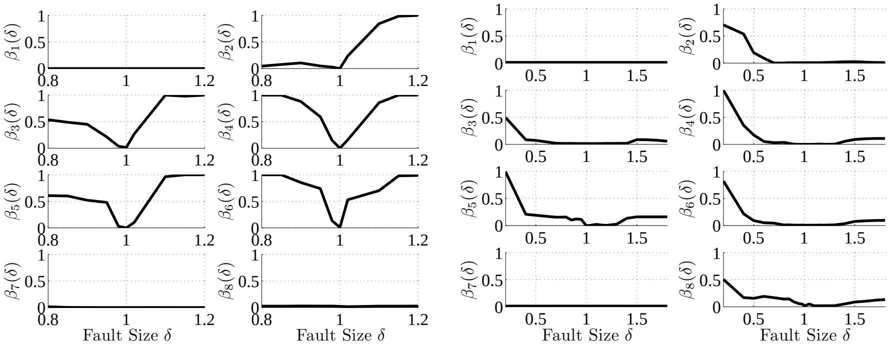
Figure 9: Residuals $r_1, r_2, \dots, r_8$ and test statistics $\lambda_1, \lambda_2, \dots, \lambda_8$ when fault $\Delta_{y_{p_{\text{ic}}}}$ is injected at time $t = 700$ s.



(a) Residuals

(b) Test Statistics

Figure 10: Residuals $r_1, r_2, \ldots, r_8$ and test statistics $\lambda_1, \lambda_2, \ldots, \lambda_8$ when fault $\Delta_{u_{x_{\text{th}}}}$ is injected at time $t = 700$ s.



(a) Fault $\Delta_{y_{p_{\text{ic}}}}$

(b) Fault $\Delta_{u_{x_{\text{th}}}}$

Figure 11: Power functions $\beta_i(\delta)$, $i = 1, 2, \ldots, 8$, for faults $\Delta_{y_{p_{\text{ic}}}}$ and $\Delta_{u_{x_{\text{th}}}}$.

in best overall averaged test power than all other faults. Faults $\Delta_{y_{T_{\text{amb}}}}$, $\Delta_{u_{x_{\text{th}}}}$, and $\Delta_{u_{x_{\text{egr}}}}$, result in quite poor test power in comparison. This can also be seen in Figure 11.

The correspondence between the FSM in Table 3 and the averaged test powers in Table 7 when it comes to non-sensitive residual generators is good, in the sense that an empty entry in Table 3 always corresponds to a zero, or almost zero, entry in Table 7. However, the converse is not always true, since there are zero, or almost zero, entries in Table 7 where there are an "x" in Table 3. In particular, this holds for faults $\Delta_{y_{p_{\text{amb}}}}$ and $\Delta_{u_{x_{\text{vgt}}}}$. According to Table 3, all residual generators may be sensitive to faults $\Delta_{y_{p_{\text{amb}}}}$ and $\Delta_{u_{x_{\text{vgt}}}}$. However, as indicated by Table 7, all tests do not respond to these faults.

6.2 Performance of FDI-System

The aim of this part of the evaluation is to investigate the detection and isolation performance of the complete FDI-system.

Metrics

To this end, the following metrics are considered.

Detection Time (DT): Time from fault injection to first detection by any test that may be sensitive to the fault.

Isolation Time (IT): Time from fault injection to first correct fault isolation statement.

Missed Detection Rate (MDR): The fraction of test runs for which the injected fault not is detected by any of the tests that may be sensitive to the fault.

Missed Isolation Rate (MIR): The fraction of test runs for which a correct fault isolation statement not is obtained.

Table 8: Fault Specifications.

Fault	Specification
$\Delta_{y_{p_{\text{amb}}}}$	$y_{p_{\text{amb}}} = 0.5 \cdot p_{\text{amb}}$
$\Delta_{y_{T_{\text{amb}}}}$	$y_{T_{\text{amb}}} = 1.3 \cdot T_{\text{amb}}$
$\Delta_{y_{p_{\text{ic}}}}$	$y_{p_{\text{ic}}} = 1.2 \cdot p_{\text{ic}}$
$\Delta_{y_{p_{\text{im}}}}$	$y_{p_{\text{im}}} = 0.9 \cdot p_{\text{im}}$
$\Delta_{y_{T_{\text{im}}}}$	$y_{T_{\text{im}}} = 0.7 \cdot T_{\text{im}}$
$\Delta_{y_{p_{\text{em}}}}$	$y_{p_{\text{em}}} = 0.8 \cdot p_{\text{em}}$
$\Delta_{u_{x_{\text{th}}}}$	$u_{x_{\text{th}}} = 0.3 \cdot u_{x_{\text{th}}}$
$\Delta_{u_{x_{\text{egr}}}}$	$u_{x_{\text{egr}}} = 0.4 \cdot u_{x_{\text{egr}}}$
$\Delta_{u_{x_{\text{vgt}}}}$	$u_{x_{\text{vgt}}} = 0.5 \cdot u_{x_{\text{vgt}}}$

False Detection Rate (FDR): The fraction of samples for which the injected fault is detected by a test that should not be sensitive to the fault, or a fault is detected by any test in a no-fault condition.

Note that all metrics are defined with respect to the complete FDI-system, and not in the context of the individual tests. This means, for instance, that a run in which where only one out of several sensitive tests responds, not will be regarded as a missed detection. A situation where only one out of several possible tests responds falsely, will on the other hand be counted as a false detection. Also note that missed detections and missed isolations are counted on test run basis, whereas false detections are counted on sample basis.

Moreover, note that with a *correct* fault isolation statement it is meant an isolation statement in accordance with the isolability matrix in Table 4. That is, when fault $\Delta_{y_{p_{\text{amb}}}}$ has occurred, the correct fault isolability statement is that either of the faults $\Delta_{y_{p_{\text{amb}}}}$ or $\Delta_{u_{x_{\text{vgt}}}}$ has occurred.

Setup

In total 12 different data sets were used in this part of the evaluation. As in the previous study, the data sets contain measurements from drives with both high-way and city parts under different conditions. Each fault specified in Table 8 was injected abruptly after a fixed time one at a time in each of the 12 data sets. This means that there were in total 12 test runs per fault. The sizes of the faults as specified in Table 8 were chosen in consultation with experienced engineers in order to be realistic for the considered diesel engine.

Results and Comments

Table 9 gives the mean, minimum, and maximum, detection time (DT), mean, minimum, and maximum, isolation time (IT), as well as the missed detection rate (MDR), missed isolation rate (MIR), and false detection rate (FDR), for all considered faults. The detection times and isolation times are given in seconds.

Table 9: Results

		$\Delta_{y_{p_{\text{amb}}}}$	$\Delta_{y_{T_{\text{amb}}}}$	$\Delta_{y_{p_{\text{ic}}}}$	$\Delta_{y_{p_{\text{im}}}}$	$\Delta_{y_{T_{\text{im}}}}$	$\Delta_{y_{p_{\text{em}}}}$	$\Delta_{u_{x_{\text{th}}}}$	$\Delta_{u_{x_{\text{egr}}}}$	$\Delta_{u_{x_{\text{vgt}}}}$
DT	Mean	49.1	78.4	33.2	41.1	86.5	39.2	66.5	75.0	90.9
	Min	5.0	2.3	18.7	18.7	4.8	11.9	9.4	2.9	6.1
	Max	83.6	35.9	72.5	115.0	290.5	61.3	166.8	116.9	144.3
IT	Mean	-	-	221.0	149.0	-	523.0	308.8	-	-
	Min	-	-	97.0	96.6	-	261.4	227.2	-	-
	Max	-	-	437.9	223.8	-	784.7	369.5	-	-
MDR		0	0	0	0	0	0	0	0	0
MIR		1	1	0.75	0.67	1	0.83	0.75	1	1
FDR		0.043	0.076	0.057	0.067	0.043	0.049	0.056	0.051	0.043

First of all, it can in Table 9 be noted that all faults can be detected within reasonable time, meaning that there were no missed detections. As seen, however, ideal isolation statements were not obtained for all faults. Nevertheless, the injected fault was contained in each of the obtained isolation statement. The occurrence of missed isolations can be explained by the fact that the FSM in Table 3 used in the isolation scheme, see Section 5.3, does not completely reflect the fault sensitivity of the tests in the FDI-system. This was illustrated in Figures 9b and 10b and will be further considered in next section.

It is evident from Table 9 that the conclusion in Section 6.1 regarding the ability to detect the pressure sensor faults $\Delta_{y_{p_{\text{ic}}}}$, $\Delta_{y_{p_{\text{im}}}}$, and $\Delta_{y_{p_{\text{em}}}}$ in a reliable way, is supported by Table 9. All of these faults result in comparatively short detection times, low rates of false detections, and can in addition be isolated to a higher extent than the other faults. The same holds for the conclusions in Section 6.1 regarding the faults $\Delta_{y_{T_{\text{amb}}}}$ and $\Delta_{u_{x_{\text{egr}}}}$, which according to Table 9 results in longer detection times, and higher rates of false detection.

The absolute values of the metrics in Table 9 depend mainly on the value of the detection thresholds. The higher the detection thresholds, the lower the rate of false detection, the higher the rate of missed detection, and the longer the detection and isolation times, and vice versa. In addition, as said in Section 5.2, the detection and isolation times is affected by the size of the sliding windows used to collect samples for the residual evaluation.

6.3 Final Tuning

Until now, no specific tuning of the FDI-system has been performed. In this section it is illustrated how the FDI-system can be tuned in order to give lower rates of missed isolation for all faults.

As said in Section 6.2, the missed isolations is a direct consequence of the mismatch between the fault sensitivity as specified by the FSM used in the isolation process, and the actual fault sensitivity. There are at least two approaches for solving this issue. The first approach is to lower the detection thresholds. This would obviously resolve situations

Table 10: Adjusted Fault Signature Matrix.

	$\Delta y_{p_{\text{amb}}}$	$\Delta y_{T_{\text{amb}}}$	$\Delta y_{p_{\text{ic}}}$	$\Delta y_{p_{\text{im}}}$	$\Delta y_{T_{\text{im}}}$	$\Delta y_{p_{\text{em}}}$	$\Delta u_{x_{\text{th}}}$	$\Delta u_{x_{\text{egr}}}$	$\Delta u_{x_{\text{vgt}}}$
T_1		x		x	x	x		x	x
T_2	x	x	x	x		x	x		x
T_3	x	x	x	x	x	x	x		x
T_4			x	x			x		
T_5			x	x	x	x	x		
T_6	x	x	x		x	x	x	x	x
T_7		x		x	x	x		x	x
T_8	x	x		x	x	x	x	x	x

similar as those depicted in Figures 9b and 10b, where a test responds but the response not is sufficient in order for the test statistic to cross the threshold. However, this would also increase the amount of false detections. In addition, the situation where a test do not respond at all to a fault, is not handled.

The second approach is to instead adjust the FSM so that it indeed represent the actual fault sensitivity of the tests. This can for example be done by exploiting the averaged test powers in Table 7. The benefit with this approach is that, in addition, the detection thresholds can be adjusted in order to achieve desired detection times, and desired rates of false and missed detections. The main drawback is that it may affect the overall detectability and isolability properties of the FDI-system, due to additional zeros in the adjusted FSM. See (Krysander, 2006, Chapter 11) for a more general treatment of this issue. Moreover, it should be noted that the adjustment of the FSM typically relies on estimated test power, which strongly depends on the features of the available data.

Results

Both approaches were applied. However, the first approach did not give satisfactory results. Despite detection thresholds resulting in fault detection rates in the magnitude of 30-40 %, the resulting missed isolation rates were not lower for all faults.

Using the second approach, the averaged powers of the residual evaluation tests as given in Table 7 were used in order to adjust the entries of the FSM in Table 3. Specifically, each "x" in the FSM in Table 3 was removed if the corresponding entry in Table 7 was lower than 0.02. The removed entries are marked with bold in Table 7. The adjusted FSM, now for residual evaluators instead of the residual generators, is given in Table 10.

The resulting isolability matrix is shown in Table 11, which should be compared with the original isolability matrix given in Table 4. It can be noted that the isolability in fact has increased in the sense that a larger fraction of the diagnosis requirement $\mathcal{F}$ in (5) is fulfilled. Specifically, 58 of the 72 fault pairs in $\mathcal{F}$ can now be isolated from each other, in comparison with 56 before.

Results in accordance with Table 9 are given for the FDI-system with the adjusted FSM in Table 12. The same detection thresholds and data were used as in the evaluation

Table 11: Isolability Matrix based on Adjusted Fault Signature Matrix.

	$\Delta_{y_{p_\text{amb}}}$	$\Delta_{y_{T_\text{amb}}}$	$\Delta_{y_{p_\text{ic}}}$	$\Delta_{y_{p_\text{im}}}$	$\Delta_{y_{T_\text{im}}}$	$\Delta_{y_{p_\text{em}}}$	$\Delta_{u_{x_\text{th}}}$	$\Delta_{u_{x_\text{egr}}}$	$\Delta_{u_{x_\text{vgt}}}$
$\Delta_{y_{p_\text{amb}}}$	x	x				x	x		x
$\Delta_{y_{T_\text{amb}}}$		x				x			x
$\Delta_{y_{p_\text{ic}}}$			x				x		
$\Delta_{y_{p_\text{im}}}$				x					
$\Delta_{y_{T_\text{im}}}$					x	x			
$\Delta_{y_{p_\text{em}}}$						x			
$\Delta_{u_{x_\text{th}}}$							x		
$\Delta_{u_{x_\text{egr}}}$		x			x	x		x	x
$\Delta_{u_{x_\text{vgt}}}$		x				x			x

Table 12: Results with Adjusted Fault Signature Matrix.

		$\Delta_{y_{p_\text{amb}}}$	$\Delta_{y_{T_\text{amb}}}$	$\Delta_{y_{p_\text{ic}}}$	$\Delta_{y_{p_\text{im}}}$	$\Delta_{y_{T_\text{im}}}$	$\Delta_{y_{p_\text{em}}}$	$\Delta_{u_{x_\text{th}}}$	$\Delta_{u_{x_\text{egr}}}$	$\Delta_{u_{x_\text{vgt}}}$
	Mean	48.1	82.9	33.2	41.1	87.0	39.2	66.5	77.8	90.7
DT	Min	5.0	2.3	18.7	18.7	4.8	11.9	9.4	2.9	6.1
	Max	83.6	35.9	72.5	115.0	290.5	61.3	166.8	116.9	144.3
	Mean	168.7	228.6	47.2	148.0	142.7	190.4	246.8	315.7	430.5
IT	Min	45.5	173.3	28.5	96.6	142.7	57.1	62.0	5.3	129.8
	Max	346.3	283.2	94.0	223.8	142.7	784.7	329.6	545.8	612.8
MDR		0	0	0	0	0	0	0	0	0
MIR		0.42	0.75	0	0.58	0.83	0.25	0.42	0.67	0.67
FDR		0.11	0.082	0.064	0.067	0.053	0.049	0.056	0.063	0.069

presented in Table 9.

It can be seen in Table 12 that the missed isolation rate (MIR) is lower for all faults, in comparison with Table 9. In addition, the isolation times are lower for all faults, and for some faults, e.g., $\Delta_{y_{p_\text{ic}}}$, the difference is significant. Furthermore, the detection times are identical, or comparable, with those given in Table 9. It may be noted that there is a slight increase in false detection rate. This is a direct consequence of the additional empty entries in the adjusted FSM shown in Table 10. Every detection of a fault by a test whose corresponding entry in Table 10 has been removed, now counts as a false detection.

7 Conclusions

It has been illustrated how an FDI-system for an automotive diesel engine can be designed by application of a generic automated design methodology. No specific adaption of the methodology to the automotive diesel engine system was made. Through the application, it has been empirically shown that employment of mixed causality substantially increased the number of realizable residual generators. Foremost, this leads to increased fault isolability as is evident by comparison of Tables 4 and 6. Moreover, it has been demonstrated how model errors of time-varying nature and magnitude can be handled in the framework of statistical likelihood-based residual evaluation. Illustrations are given in Figures 9 and 10.

The FDI-system, and thus the potential of the automated design methodology, has been evaluated using road and test-bed measurements. The overall performance of the FDI-system is good in comparison with the required design effort. The fault sensitivities of the individual fault detection tests have been investigated by means of the estimated averaged test power (8). It was concluded that the fault sensitivity indicated in the FSM in Table 3, not fully corresponded to the fault sensitivity as given by the averaged test powers shown in Table 7. Specifically, this results in high missed isolation rates. It has been illustrated that an adjustment of the original FSM by utilization of the averaged test powers, resulting in the adjusted FSM in Table 10, gives an FDI-system capable of isolating more faults from each other, as can be seen by a comparison of Tables 11 and 4. In addition, this also resulted in increased fault isolation performance, in terms of substantially lower missed isolation rater and lower isolation times, in comparison with the original FSM, which can be seen by a comparison of Tables 12 and 9.

Acknowledgment

This work was sponsored by Scania and VINNOVA (Swedish Governmental Agency for Innovation Systems).

A Model Equations

$$
\begin{aligned}
e_1: &\quad \dot{p}_{\mathrm{ic}} = \frac{R_{\mathrm{a}} T_{\mathrm{im}}}{V_{\mathrm{ic}}} \left(W_{\mathrm{c}} - W_{\mathrm{th}} \right) \\
e_2: &\quad \dot{p}_{\mathrm{im}} = \frac{R_{\mathrm{a}} T_{\mathrm{im}}}{V_{\mathrm{im}}} \left(W_{\mathrm{th}} + W_{\mathrm{egr}} - W_{\mathrm{ei}} \right) \\
e_3: &\quad \dot{p}_{\mathrm{em}} = \frac{R_{\mathrm{e}} T_{\mathrm{em}}}{V_{\mathrm{em}}} \left(W_{\mathrm{eo}} - W_{\mathrm{egr}} - W_{\mathrm{t}} \right) + \frac{R_{\mathrm{e}}}{V_{\mathrm{em}} c_{\mathrm{ve}}} \left(W_{\mathrm{in}} c_{\mathrm{ve}} \left(T_{\mathrm{em,in}} - T_{\mathrm{em}} \right) \right. \\
&\qquad \left. + R_{\mathrm{e}} \left(T_{\mathrm{em,in}} W_{\mathrm{in}} - T_{\mathrm{em}} W_{\mathrm{out}} \right) \right) \\
e_4: &\quad \dot{T}_{\mathrm{em}} = \frac{R_{\mathrm{e}} T_{\mathrm{em}}}{p_{\mathrm{em}} V_{\mathrm{em}} c_{\mathrm{ve}}} \left(W_{\mathrm{in}} c_{\mathrm{ve}} \left(T_{\mathrm{em,in}} - T_{\mathrm{em}} \right) + R_{\mathrm{e}} \left(T_{\mathrm{em,in}} W_{\mathrm{in}} - T_{\mathrm{em}} W_{\mathrm{out}} \right) \right) \\
e_5: &\quad W_{\mathrm{in}} = \max(W_{\mathrm{eo}}, 0) + \max(-W_{\mathrm{egr}}, 0) + \max(-W_{\mathrm{t}}, 0) \\
e_6: &\quad W_{\mathrm{out}} = \max(-W_{\mathrm{eo}}, 0) + \max(W_{\mathrm{egr}}, 0) + \max(W_{\mathrm{t}}, 0) \\
e_7: &\quad W_{\mathrm{th}} = \frac{p_{\mathrm{ic}} A_{\mathrm{th,max}}}{\sqrt{T_{\mathrm{im}} R_{\mathrm{a}}}} \Psi_{\mathrm{th}}^{\gamma_{\mathrm{th}}} (\Pi_{\mathrm{th}}) f_{\mathrm{th}}(x_{\mathrm{th}})
\end{aligned}
$$

$$e_8: \quad \Pi_{\mathrm{th}} = f_{\Pi_{\mathrm{th}}}(p_{\mathrm{im}}, p_{\mathrm{ic}})$$

$$e_9: \quad W_{\mathrm{ei}} = \frac{\eta_{\mathrm{vol}} p_{\mathrm{im}} n_{\mathrm{e}} V_d}{120 R_{\mathrm{a}} T_{\mathrm{im}}}$$

$$e_{10}: \quad \eta_{\mathrm{vol}} = c_{\mathrm{vol1}} \frac{r_{\mathrm{c}} - \left(\frac{p_{\mathrm{em}}}{p_{\mathrm{im}}}\right)^{1/\gamma_{\mathrm{e}}}}{r_{\mathrm{c}} - 1} + c_{\mathrm{vol2}} W_{\mathrm{f}}^2 + c_{\mathrm{vol3}} W_{\mathrm{f}} + c_{\mathrm{vol4}}$$

$$e_{11}: \quad W_{\mathrm{f}} = \frac{10^{-6}}{120} \delta n_{\mathrm{e}} n_{cyl}$$

$$e_{12}: \quad W_{\mathrm{eo}} = W_{\mathrm{f}} + W_{\mathrm{ei}}$$

$$e_{13}: \quad T_{\mathrm{e}} = T_{\mathrm{im}} + \frac{q_{HV} f_{T_{\mathrm{e}} W_{\mathrm{f}}}(W_{\mathrm{f}}) f_{T_{\mathrm{e}} n_{\mathrm{e}}}(n_{\mathrm{e}})}{c_{\mathrm{pe}} W_{\mathrm{eo}}}$$

$$e_{14}: \quad T_{\mathrm{em,in}} = T_{\mathrm{amb}} + (T_{\mathrm{e}} - T_{\mathrm{amb}}) \exp(-\frac{h_{tot} \pi d_{pipe} l_{pipe} n_{pipe}}{W_{\mathrm{eo}} c_{\mathrm{pe}}})$$

$$e_{15}: \quad W_{\mathrm{egr}} = f_{W_{\mathrm{egr}}}(p_{\mathrm{im}}, p_{\mathrm{em}}, T_{\mathrm{em}}, x_{\mathrm{egr}})$$

$$e_{16}: \quad \dot{\omega}_{\mathrm{t}} = \frac{P_{\mathrm{t}} \eta_m - P_{\mathrm{c}}}{J_{\mathrm{t}} \omega_{\mathrm{t}}}$$

$$e_{17}: \quad P_{\mathrm{t}} \eta_{\mathrm{m}} = \eta_{\mathrm{tm}} W_{\mathrm{t}} c_{\mathrm{pe}} T_{\mathrm{em}} \left(1 - \Pi_{\mathrm{t}}^{1-1/\gamma_{\mathrm{e}}}\right)$$

$$e_{18}: \quad \eta_{\mathrm{tm}} = \eta_{\mathrm{tm,BSR}}(BSR) \eta_{\mathrm{tm},\omega_{\mathrm{t}}}(\omega_{\mathrm{t}}) \eta_{\mathrm{tm},x_{\mathrm{vgt}}}(x_{\mathrm{vgt}})$$

$$e_{19}: \quad BSR = \frac{R_{\mathrm{t}} \omega_{\mathrm{t}}}{\sqrt{2 c_{\mathrm{pe}} T_{\mathrm{em}} \left(1 - \Pi_{\mathrm{t}}^{1-1/\gamma_{\mathrm{e}}}\right)}}$$

$$e_{20}: \quad \Pi_{\mathrm{t}} = \frac{p_{\mathrm{t}}}{p_{\mathrm{em}}}$$

$$e_{21}: \quad W_{\mathrm{t}} = \frac{A_{\mathrm{vgt,max}} p_{\mathrm{em}}}{\sqrt{T_{\mathrm{em}} R_{\mathrm{e}}}} f_{\Pi_{\mathrm{t}}}(\Pi_{\mathrm{t}}) f_{\omega_{\mathrm{t}}}(\omega_{\mathrm{t,corr}}) f_{\mathrm{vgt}}(x_{\mathrm{vgt}})$$

$$e_{22}: \quad \omega_{t,corr} = \frac{\omega_{\mathrm{t}}}{100 \sqrt{T_{\mathrm{em}}}}$$

$$e_{23}: \quad P_{\mathrm{c}} = \frac{W_{\mathrm{c}} c_{pa} T_{\mathrm{bc}}}{\eta_{\mathrm{c}}} \left(\Pi_{\mathrm{c}}^{1-1/\gamma_{\mathrm{a}}} - 1\right)$$

$$e_{24}: \quad \Pi_{\mathrm{c}} = \frac{p_{\mathrm{ic}}}{p_{\mathrm{bc}}}$$

$$e_{25}: \quad \eta_{\mathrm{c}} = \eta_{c,W}(W_{c,corr}, \Pi_{\mathrm{c}}) \eta_{c,\Pi}(\Pi_{\mathrm{c}})$$

$$e_{26}: \quad W_{\mathrm{c,corr}} = \frac{\sqrt{(T_{\mathrm{bc}}/T_{\mathrm{ref}})}}{\sqrt{(p_{\mathrm{bc}}/p_{\mathrm{ref}})}} W_{\mathrm{c}}$$

$$e_{27}: \quad W_{\mathrm{c}} = \frac{p_{\mathrm{bc}} \pi R_{\mathrm{c}}^3 \omega_{\mathrm{t}}}{R_{\mathrm{a}} T_{\mathrm{bc}}} \Phi_{\mathrm{c}}$$

$$e_{28}: \quad \Phi_{\mathrm{c}} = \frac{k_{c1} - k_{c3} \Psi_{\mathrm{c}}}{k_{c2} - \Psi_{\mathrm{c}}}$$

$$e_{29}: \quad k_{c1} = k_{c11}\left(\min(Ma, Ma_{max})\right)^2 + k_{c12}\min(Ma, Ma_{max}) + k_{c13}$$

$$e_{30}: \quad k_{c2} = k_{c21}\left(\min(Ma, Ma_{max})\right)^2 + k_{c22}\min(Ma, Ma_{max}) + k_{c23}$$

$$e_{31}: \quad k_{c3} = k_{c31}\left(\min(Ma, Ma_{max})\right)^2 + k_{c32}\min(Ma, Ma_{max}) + k_{c33}$$

$$e_{32}: \quad Ma = \frac{R_\mathrm{c}\omega_\mathrm{t}}{\sqrt{\gamma_\mathrm{a} R_\mathrm{a} T_\mathrm{bc}}}$$

$$e_{33}: \quad \Psi_\mathrm{c} = \frac{2 c_{pa} T_\mathrm{bc}\left(\Pi_\mathrm{c}^{1-1/\gamma_\mathrm{a}} - 1\right)}{R_\mathrm{c}^2 \omega_\mathrm{t}^2}$$

$$e_{34}: \quad p_\mathrm{bc} = p_\mathrm{amb}$$

$$e_{35}: \quad T_\mathrm{bc} = T_\mathrm{amb}$$

$$e_{36}: \quad y_{p_\mathrm{amb}} = p_\mathrm{amb} + \Delta_{y_{p_\mathrm{amb}}}$$

$$e_{37}: \quad y_{T_\mathrm{amb}} = T_\mathrm{amb} + \Delta_{y_{T_\mathrm{amb}}}$$

$$e_{38}: \quad y_{p_\mathrm{ic}} = p_\mathrm{ic} + \Delta_{y_{p_\mathrm{ic}}}$$

$$e_{39}: \quad y_{p_\mathrm{im}} = p_\mathrm{im} + \Delta_{y_{p_\mathrm{im}}}$$

$$e_{40}: \quad y_{T_\mathrm{im}} = T_\mathrm{im} + \Delta_{y_{T_\mathrm{im}}}$$

$$e_{41}: \quad y_{p_\mathrm{em}} = p_\mathrm{em} + \Delta_{y_{p_\mathrm{em}}}$$

$$e_{42}: \quad u_{x_\mathrm{th}} = x_\mathrm{th} + \Delta_{u_{x_\mathrm{th}}}$$

$$e_{43}: \quad u_{x_\mathrm{egr}} = x_\mathrm{egr} + \Delta_{u_{x_\mathrm{egr}}}$$

$$e_{44}: \quad u_{x_\mathrm{vgt}} = x_\mathrm{vgt} + \Delta_{u_{x_\mathrm{vgt}}}$$

$$e_{45}: \quad u_\delta = \delta$$

$$e_{46}: \quad y_{n_\mathrm{e}} = n_\mathrm{e}$$

References

I. M. Al-Salami, S. X. Ding, and P. Zhang. Statistical based residual evaluation for fault detection in networked control systems. In *Proceedings of Workshop on Advances Control and Diagnosis*, Nancy, France, November 2006. Nancy University.

M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, 1993.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, second edition, 2006.

M. R. Blas and M. Blanke. Stereo vision with texture learning for fault-tolerant automatic baling. *Computers and Electronics in Agriculture*, 75(1):159 – 68, 2011.

California EPA. Sections 1971.1, 1968.2, and 1971.5 of title 13, california code of regulations: HD OBD and OBD II regulations. http://www.arb.ca.gov/msprog/obdprog/hdobdreg.htm, 2010. California Environmental Protection Agency, Air Resources Board.

G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, second edition, 2001.

J. P. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *Proceedings of IFAC Control Ind. Syst.*, pages 841–846, Belfort, France, 1997.

V. Cocquempot, R. Izadi-Zamanabadi, M. Staroswiecki, and M. Blanke. Residual generation for the ship benchmark using structural approach. In *Proceedings of the UKACC International Conference on Control '98*, pages 1480–1485, September 1998.

J. de Kleer and B. C Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1): 97–130, 1987.

D. Dustegor, V. Cocquempot, and M. Staroswiecki. Structural analysis for residual generation: Towards implementation. In *Proceedings of the 2004 IEEE Inter. Conf. on Control App.*, pages 1217–1222, 2004.

European Parliament. Regulation No 595/2009 of the european parliament and of the council of 18 june 2009 on type-approval of motor vehicles and engines with respect to emissions from heavy duty vehicles (Euro VI) and on access to vehicle repair and maintenance information and amending Regulation (EC) No 715/2007 and Directive 2007/46/EC and repealing Directives 80/1269/EEC, 2005/55/EC and 2005/78/EC, 2009. European Parliament and the Council of the European Union.

P. Fogh Odgaard, J. Stoustrup, and M. Kinnaert. Fault tolerant control of wind turbines - a benchmark model. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 155–160, Barcelona, Spain, 2009.

J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, 1998.

R. Greiner, B. A. Smith, and R. W. Wilkerson. A correction to the algorithm in reiter's theory of diagnosis. *Artificial Intelligence*, 41:79–88, 1989.

E. Höckerdal, E. Frisk, and L. Eriksson. EKF-based adaptation of look-up tables with an air mass-flow sensor application. *Control Engineering Practice*, 19(5):442–453, 2011a.

E. Höckerdal, E. Frisk, and L. Eriksson. Bias reduction in DAE estimators by model augmentation: Observability analysis and experimental evaluation. In *50th IEEE Conference on Decision and Control*, Orlando, Florida, USA, 2011b.

R. Izadi-Zamanabadi. Structural analysis approach to fault fiagnosis with application to fixed-wing aircraft motion. In *Proceedings of the 2002 American Control Conference*, volume 5, pages 3949–3954, 2002.

M. Krysander. *Design and Analysis of Diagnosis Systems Using Structural Methods*. PhD thesis, Linköpings universitet, June 2006.

M. Krysander, J. Åslund, and M. Nyberg. An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(1):197–206, 2008.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.

J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, May 2010.

J. Mattingley and S. Boyd. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

M. Nyberg. Automatic design of diagnosis systems with application to an automotive engine. *Control Engineering Practice*, 87(8):993–1005, 1999.

M. Nyberg and M. Krysander. Statistical properties and design criterions for AI-based fault isolation. In *Proceedings of the 17th IFAC World Congress*, pages 7356–7362, Seoul, Korea, 2008.

M. Nyberg and T. Stutte. Model based diagnosis of the air path of an automotive diesel engine. *Control Engineering Practice*, 12(5):513 – 525, 2004.

Y. Peng, A. Youssouf, P. Arte, and M. Kinnaert. A complete procedure for residual generation and evaluation with application to a heat exchanger. *IEEE Transactions on Control Systems Technology*, 5(6):542 – 555, 1997.

S. Ploix, M. Desinde, and S. Touaf. Automatic design of detection tests in complex dynamic systems. In *Proceedings of 16th IFAC World Congress*, Prague, Czech Republic, 2005.

B. Pulido and C. Alonso-González. Possible conflicts: a compilation technique for consistency-based diagnosis. *"IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics", Special Issue on Diagnosis of Complex Systems*, 34(5):2192–2206, 2004.

R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

M. Staroswiecki. *Fault Diagnosis and Fault Tolerant Control*, chapter Structural Analysis for Fault Detection and Isolation and for Fault Tolerant Control. Encyclopedia of Life Support Systems, Eolss Publishers, Oxford, UK, 2002.

M. Staroswiecki and P. Declerck. Analytical redundancy in non-linear interconnected systems by means of structural analysis. In *Proceedings of IFAC AIPAC'89*, pages 51–55, Nancy, France, 1989.

C. Svärd and M. Nyberg. Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(6):1310–1328, 2010.

C. Svärd and M. Nyberg. Automated design of an FDI-system for the wind turbine benchmark. *Journal of Control Science and Engineering*, vol. 2012, 2012. Article ID 989873, 13 pages.

L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component-supported analytical redundancy. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 36(6):1146–1160, November 2006.

United Nations. Regulation no. 49: Uniform provisions concerning the measures to be taken against the emission of gaseous and particulate pollutants from compressionignition engines for use in vehicles, and the emission of gaseous pollutants from positive-ignition engines fuelled with natural gas or liquefied petroleum gas for use in vehicles, 2008. ECE-R49.

United States EPA. 40 CFR Part 86, 89, et al: Control of air pollution from new motor vehicles and new motor vehicle engines; final rule. http://www.epa.gov/obd/regtech/heavy.htm, 2009. United States Environmental Protection Agency.

O. Vainio, M. Renfors, and T. Saramaki. Recursive implementation of fir differentiators with optimum noise attenuation. *IEEE Transactions on Instrumentation and Measurement*, 46(5):1202 –11207, oct 1997.

J. Wahlström and L. Eriksson. Modeling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 225(7), July 2011.

X. Wei, H. Liu, and Y. Qin. Fault diagnosis of rail vehicle suspension systems by using glrt. In *Control and Decision Conference (CCDC), 2011 Chinese*, pages 1932 –1936, may 2011.

A. Willsky and H. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control*, 21(1): 108 – 112, 1976.

Paper E

Automated Design of an FDI-System for the Wind Turbine Benchmark[☆]

[☆] Published in *Journal of Control Science and Engineering*, Volume 2012, Article ID 989873, 13 pages, 2012.

Automated Design of an FDI-System for the Wind Turbine Benchmark

Carl Svärd and Mattias Nyberg

*Vehicular Systems, Department of Electrical Engineering,
Linköping University, SE-581 83 Linköping, Sweden.*

Abstract

We propose an FDI-system for the wind turbine benchmark designed by application of a generic automated method. No specific adaptation of the method for the wind turbine benchmark is needed, and the number of required human decisions, assumptions, as well as parameter choices, is minimized. The method contains in essence three steps: generation of candidate residual generators, residual generator selection, and diagnostic test construction. The proposed FDI-system performs well in spite of no specific adaptation or tuning to the benchmark. All faults in the pre-defined test sequence can be detected and all faults, except a double fault, can also be isolated shortly thereafter. In addition, there are no false or missed detections.

1 Introduction

Wind turbines stand for a growing part of power production. The demands for reliability are high, since wind turbines are expensive and their off-time should be minimized. One potential way to meet the reliability demands is to adopt fault tolerant control (FTC), i.e., prevent faults from developing into failures by taking appropriate actions. A typical action is reconfiguration of the control system. An essential part of an FTC-system is the fault detection and isolation (FDI) system, see, e.g., Blanke et al. (2006). To obtain good detection and isolation of faults, model-based FDI is often necessary.

Design of a complete model-based FDI-system is a complex task and involves by necessity several decisions, for example, method choices, tuning of parameters, and assumptions regarding noise distributions and the nature of the faults to be diagnosed. In general, an optimal solution requires detailed knowledge of the behavior of the considered system, something that is rarely available for real applications. In this paper, inspired by work with real industrial applications, we propose an automated design method that minimizes the number of required human decisions and assumptions. Furthermore, we investigate the potential of designing an FDI-system for the wind turbine benchmark, see Fogh Odgaard et al. (2009), using this automated method.

The design method is composed of three main steps. In the first step, a large set of candidate residual generators are generated using the algorithm described in Krysander et al. (2008). In the second step, the residual generators most suitable to be included in the final FDI-system are selected and realized by means of a greedy selection algorithm, based on ideas elaborated in Svärd et al. (2011). The realization, or construction, of residual generators is done by use of the algorithms presented in Svärd and Nyberg (2010). In the third and final step, we design diagnostic tests based on the residuals obtained as output from the selected set of residual generators. The diagnostic tests relies on a novel methodology based on a comparison of the probability distributions of no-fault residuals, estimated offline using no-fault training data, and the distributions of residuals estimated online using current data.

As it turns out, the proposed FDI-system performs well when evaluated on the test sequence described in Fogh Odgaard et al. (2009). A tailor-made FDI-system perfectly tuned for the wind turbine benchmark would probably perform better than the one we propose. However, in relation to the minimal effort required for application of the automated design method, and in spite of no extra tuning or specific adaptation to the benchmark, the performance of the FDI-system is satisfactory; all faults in the test sequence can be detected within feasible time, and there are no false or missed detections. Further, all faults, except a double fault, can also be isolated.

The wind turbine benchmark model and the strategy used for modeling of faults, are described in Section 2. Section 3 presents an overview of the design method. The method for constructing residual generators is described in Section 4, and the approach used for selecting residual generators is described in Section 5. The method for design of diagnostic tests, and the fault isolation scheme is considered in Section 6. Some implementation specific details are discussed in Section 7. The performance of the designed FDI-system is evaluated and discussed in Section 8, and Section 9 concludes the paper.

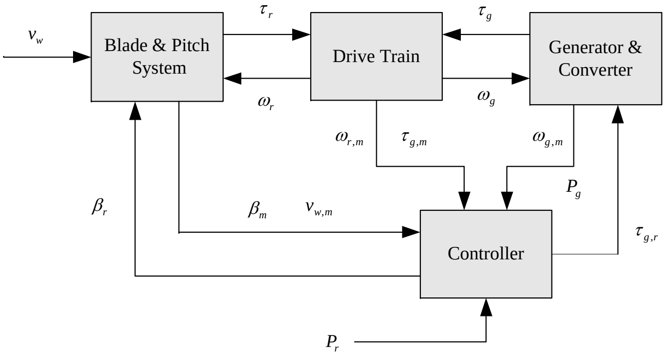


Figure 1: Overview of the wind turbine system.

2 The Wind Turbine Model

The wind turbine system is described and modeled in Fogh Odgaard et al. (2009), to which is referred for details. The considered wind turbine system has three rotor blades and the system contains four sub-systems: blade and pitch system, drive train, generator and converter, and controller, see Figure 1 and Table 1.

2.1 State-Space Realization of Transfer Functions

The pitch system and converter are modeled as frequency domain transfer functions. The residual generation algorithm we intend to apply, assume a model described in differential and algebraic equations. To obtain a model in this form, the transfer functions are realized as time-domain state-space systems.

The relation between pitch angle reference β_r and pitch angle output β_i, for each of the three blades and thus for $i = 1, 2, 3$, can be realized in state-space form using observable canonical form, see, e.g., Rugh (1996), as follows

$$\dot{x}_{\beta_{i1}}(t) = -2\zeta\omega_n x_{\beta_{i1}}(t) + x_{\beta_{i2}}(t) \tag{1a}$$

$$\dot{x}_{\beta_{i2}}(t) = -\omega_n^2 x_{\beta_{i1}}(t) + \omega_n^2 \beta_r(t) \tag{1b}$$

$$\beta_i(t) = x_{\beta_{i1}}(t), \tag{1c}$$

where ζ, ω_n are parameters, and $x_{\beta_{i1}}$, $x_{\beta_{i2}}$ state variables. Using the same approach, the relation between converter reference $\tau_{g,r}$ and output τ_g can be written as

$$\dot{x}_{\tau_g}(t) = -\alpha_{gc} x_{\tau_g}(t) + \alpha_{gc}\tau_{g,r}(t) \tag{2a}$$

$$\tau_g(t) = x_{\tau_g}(t), \tag{2b}$$

where α_{gc} is a parameter, and x_{τ_g} the state variable.

Table 1: Signals in the wind turbine system.

Signal	Description
v_w	Wind speed
$v_{w,m}$	Wind speed measurement
β_r	Pitch angle reference
β_m	Pitch angle measurement
ω_r	Angular rotor speed
$\omega_{r,m}$	Angular rotor speed measurement
ω_g	Generator rotor speed
$\omega_{g,m}$	Generator rotor speed measurement
τ_r	Rotor torque
τ_g	Generator torque
$\tau_{g,r}$	Generator torque reference
$\tau_{g,m}$	Generator torque measurement
P_r	Power reference
P_g	Generator power

2.2 Fault Modeling

The set of faults to consider for the wind turbine is specified in Fogh Odgaard et al. (2009) and given by

$$F = \{\Delta\beta_1, \Delta\beta_2, \Delta\beta_3, \Delta\tau_g, \Delta\omega_g, \Delta\beta_{1,m1}, \Delta\beta_{1,m2}, \Delta\beta_{2,m1}, \Delta\beta_{2,m2}, \Delta\beta_{3,m1}, \Delta\beta_{3,m2}, \Delta\omega_{r,m1}, \Delta\omega_{r,m2}, \Delta\omega_{g,m1}, \Delta\omega_{g,m2}\},$$

where $\Delta\beta_1$, $\Delta\beta_2$, $\Delta\beta_3$, and $\Delta\tau_g$ are actuator faults, $\Delta\omega_g$ a system fault, and $\Delta\beta_{1,m1}$, $\Delta\beta_{1,m2}$, $\Delta\beta_{2,m1}$, $\Delta\beta_{2,m2}$, $\Delta\beta_{3,m1}$, $\Delta\beta_{3,m2}$, $\Delta\omega_{r,m1}$, $\Delta\omega_{r,m2}$, $\Delta\omega_{g,m1}$, and $\Delta\omega_{g,m2}$, sensor faults.

To incorporate fault information in the nominal model, we have chosen to model all faults as additive signals in corresponding equations. Thus, we are not taking into account all information regarding the nature of faults given in Fogh Odgaard et al. (2009). Consider for example fault $\Delta\beta_1$ which represents an actuator fault in pitch system 1, see (1), resulting in changed dynamics of β_1 due to dropped main line pressure or high air content in the oil. One possible way to model this fault would be as a deviation in parameters ω_n and ζ in (1a) and (1b). With the chosen approach, the fault is instead modeled as an additive signal in (1c) for $i = 1$, i.e., $\beta_1 = x_{\beta_{11}} + \Delta\beta_1$.

Note that the adopted fault modeling approach is general and no assumptions are made regarding for example the time-behavior of faults. Thus, the approach is able to handle for example multiplicative faults even though the fault signal is assumed to be additive. Consider for example a multiplicative fault in β_1 given by $\beta_1 = \delta \cdot x_{\beta_{11}}$ where $\delta \neq 1$, which can be equivalently described by $\beta_1 = x_{\beta_{11}} + \Delta\beta_1$, where $\Delta\beta_1 = x_{\beta_{11}}(\delta - 1)$.

The main argument for using this, more general, approach is that we consider it hard, or even impossible, to know exactly how a faulty component behaves in reality. Furthermore, data from all fault-cases for evaluation and validation of a more detailed model are seldom available. Modeling faults in this way also results in a minimum of fault modes. This is beneficial since it gives a smaller model which simplifies several steps in model-based diagnosis, e.g., residual generation and isolation. In addition, regarding how diagnosis information is utilized, e.g., for Fault Tolerant Control, it is unnecessary to distinguish between different fault modes if they are associated with the same action or consequence. Indeed, this applies to all sensor faults in the wind turbine, since the system should be reconfigured regardless of the type of sensor fault, i.e., *fixed value* or *gain factor*, see Table 2 in Fogh Odgaard et al. (2009). Last, but not least, an additional important motivator is simplicity, since extending the nominal model with additive fault signals in this way is straightforward and easy.

2.3 Model Extensions

According to Fogh Odgaard et al. (2009), the same pitch angle reference signal β_r is fed to all three pitch systems (1), i.e., $\beta_{i,r} = \beta_r$ for $i = 1, 2, 3$. However, according to the provided Simulink© model, see Fogh Odgaard (2011), the individual reference signals are instead calculated in a control loop outside the pitch system as

$$\beta_{i,r} = \beta_r + \beta_i - \left(\frac{\beta_{i,m1} + \beta_{i,m2}}{2}\right), \quad i = 1, 2, 3 \tag{3}$$

where β_i is given by (1), and $\beta_{i,m1}$ and $\beta_{i,m2}$ are sensor measurements. To incorporate this information in the design of the FDI system, the original wind turbine model is extended with the relations between $\beta_{i,r}$ and β_r given by (3).

2.4 The Model with Faults

The complete model of the wind turbine model, with fault signals denoted by Δ, used in this work for design of an FDI-system is given below.

$$\begin{aligned}
e_1: &\quad \tau_r = \sum_{i=1}^{3} \frac{\rho \pi R^3 C_q(\lambda, \beta_i) v_w^2}{6} \\
e_2: &\quad \lambda = \frac{\omega_r R}{v_w} \\
e_3, e_5, e_7: &\quad \dot{x}_{\beta_{i1}} = -2\zeta\omega_n x_{\beta_{i1}} + x_{\beta_{i2}}, \quad i = 1, 2, 3 \\
e_4, e_6, e_8: &\quad \dot{x}_{\beta_{i2}} = -\omega_n^2 x_{\beta_{i1}} + \omega_n^2 \beta_{i,r}, \quad i = 1, 2, 3 \\
e_9, e_{10}, e_{11}: &\quad \beta_i = x_{\beta_{i1}} + \Delta\beta_i, \quad i = 1, 2, 3 \\
e_{12}: &\quad \dot{\omega}_g = \left(\frac{\eta_{dt} B_{dt}}{N_g J_g}\right)\omega_r + \left(\frac{-\frac{\eta_{dt} B_{dt}}{N_g^2} - B_g}{J_g}\right)\omega_g + \left(\frac{\eta_{dt} K_{dt}}{N_g J_g}\right)\theta_\Delta - \left(\frac{1}{J_g}\right)\tau_g + \Delta\omega_g \\
e_{13}: &\quad \dot{\omega}_r = -\left(\frac{B_{dt} - B_r}{J_r}\right)\omega_r + \left(\frac{B_{dt}}{N_g J_r}\right)\omega_g - \left(\frac{K_{dt}}{J_r}\right)\theta_\Delta + \left(\frac{1}{J_r}\right)\tau_r \\
e_{14}: &\quad \dot{\theta}_\Delta = \omega_r - \left(\frac{1}{N_g}\right)\omega_g
\end{aligned}$$

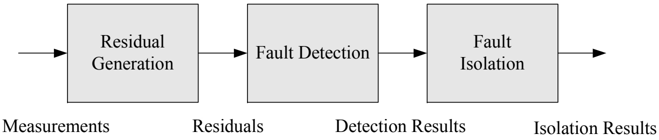


Figure 2: Schematic overview of the FDI-system.

$$\begin{aligned}
e_{15}:&\quad \dot{x}_{\tau_g} = -\alpha_{gc} x_{\tau_g} + \alpha_{gc}\tau_{g,r}\\
e_{16}:&\quad \tau_g = x_{\tau_g} + \Delta\tau_g\\
e_{17}:&\quad P_g = \eta_{gc}\omega_g\tau_g\\
e_{18}, e_{20}, e_{22}:&\quad \beta_{i,m1} = \beta_i + \Delta\beta_{i,m1}, \quad i = 1,2,3\\
e_{19}, e_{21}, e_{23}:&\quad \beta_{i,m2} = \beta_i + \Delta\beta_{i,m2}, \quad i = 1,2,3\\
e_{24}, e_{25}:&\quad \omega_{r,mj} = \omega_r + \Delta\omega_{r,mj}, \quad j = 1,2\\
e_{26}, e_{27}:&\quad \omega_{g,mj} = \omega_g + \Delta\omega_{g,mj}, \quad j = 1,2\\
e_{28}:&\quad v_{w,m} = v_w\\
e_{29}:&\quad \tau_{g,m} = \tau_g\\
e_{30}:&\quad P_{g,m} = P_g\\
e_{31}, e_{32}, e_{33}:&\quad \beta_{i,r} = \beta_r + \beta_i - \left(\frac{\beta_{i,m1} + \beta_{i,m2}}{2}\right), i = 1,2,3
\end{aligned}$$

3 Overview of Design Method

The proposed FDI-system for the wind turbine is comprised of three sub-systems: residual generation, fault detection and fault isolation, see Figure 2.

Measurements, i.e., sensor readings, from the wind turbine are fed to a bank of residual generators whose output is a set of residuals. The residuals are used as input to the fault detection block, which contains diagnostic tests based on the residuals. The output from this block, one signal for each residual, indicates if a fault has been detected in the part of the system monitored by the corresponding residual. The result from the fault detection is fed to the fault isolation block in which the detected fault(s) are isolated.

The proposed method supports design of the residual generation and fault detection blocks. Design of the fault isolation block is briefly discussed in Section 6.2. The method contains three essential steps:

1. Generate candidate residual generators,
2. Select and realize residual generators,
3. Construct diagnostic tests,

see Figure 3. In the first step, a large set of candidate residual generators are generated. In the second step, the residual generators most suitable to be included in the final FDI-

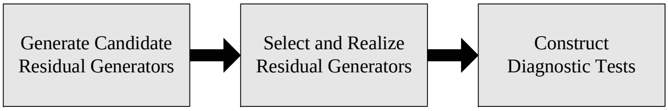

Generate Candidate Residual Generators
Select and Realize Residual Generators
Construct Diagnostic Tests

Figure 3: Overview of the design method.

system are selected and realized. In the third and final step, we design diagnostic tests based on the residuals obtained as output from the selected set of residual generators.

In the subsequent sections, we describe in detail the different steps of the design method used to create the proposed FDI-system for the wind turbine benchmark system. As input to the design method, or prerequisites, we assume a model of the system and no-fault training data. The data is assumed to be expressed as measurements, either real or simulated, of the inputs and outputs of the model in realistic and representative no-fault operating conditions.

4 Residual Generation

The set of residual generators used in the FDI-system are based upon the ideas originally described in Staroswiecki and Declerck (1989), where unknown variables in a model are computed by solving equation sets one at a time in a sequence and a residual is obtained by evaluating a redundant equation. Similar approaches are described and exploited in for example Cassar and Staroswiecki (1997); Staroswiecki (2002); Pulido and Alonso-González (2004); Ploix et al. (2005); Travé-Massuyès et al. (2006); Blanke et al. (2006); Svärd and Nyberg (2010). This class of residual generation methods, referred to as *sequential residual generation*, has shown to be successful for real applications and also has the potential to be automated to a high extent.

4.1 Sequential Residual Generation

Some concepts and results of sequential residual generation given in Svärd and Nyberg (2010), to which we also refer for technical details, will now be briefly recapitulated. We consider a model (E, X, D, Y) to be a set of differential and algebraic equations $E = \{e_1, e_2, \dots, e_{n_E}\}$ containing unknown variables $X = \{x_1, x_2, \dots, x_{n_X}\}$, differential variables $D = \{\dot{x}_1, \dot{x}_2, \dots, \dot{x}_{n_X}\}$, and known variables $Y = \{y_1, y_2, \dots, y_{n_Y}\}$. The equations in E are, without loss of generality, assumed to be on the form

$$e_i: \quad f_i(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}) = 0, \quad i = 1, 2, \dots, n_E, \tag{4}$$

where $\dot{\mathbf{x}}$, $\mathbf{x}$ and $\mathbf{y}$ are vectors of the variables in D, X, and Y respectively. Note that the model of the wind turbine presented in Section 2.4 can trivially be cast into this form.

Computation Sequence

As said above, the main idea in sequential residual generation is to compute unknown variables in the model by solving equation sets one at a time in a sequence, and then evaluate a redundant equation to obtain a residual. An essential component in the design of a residual generator is therefore a *computation sequence*, which describes the order in which the variables should be computed. In Svärd and Nyberg (2010), a computation sequence is defined as an ordered set of variable and equation pairs

$$\mathcal{C} = \left((V_1, E_1), (V_2, E_2), \ldots, (V_k, E_k) \right), \tag{5}$$

where $V_i \subseteq X \bigcup D$ and $E_i \subseteq E$. The computation sequence $\mathcal{C}$ implies that first the variables in V_1 are computed from equations E_1, then the variables in V_2 from equations E_2, possibly using the already computed variables in V_1, and so forth.

For an example, consider the computation sequence

$$\mathcal{C} = \left(\left(\left\{ \tau_g \right\}, \left\{ e_{29} \right\} \right), \left(\left\{ \omega_r \right\}, \left\{ e_{24} \right\} \right), \left(\left\{ \dot{\theta}_\Delta \right\}, \left\{ e_{14} \right\} \right), \left(\left\{ \dot{\omega}_g \right\}, \left\{ e_{12} \right\} \right) \right) \tag{6}$$

for computation of a subset of the unknown variables in wind turbine model presented in Section 2.4. According to the computation sequence (6), the series of computations begins with computation of variable τ_g using equation e_{29}, then variable ω_r is computed using equation e_{24}, and so on, ending with computation of variable $\dot{\omega}_g$, or in fact ω_g from equation e_{12}.

By construction, see Svärd and Nyberg (2010), it is guaranteed that no variable is needed before it has been computed. Hence, the series of computations described by the computation sequence exhibit an upper triangular structure. For the computation sequence (6), this series of computations is given by

$$\tau_g = \tau_{g,m} \tag{7a}$$

$$\omega_r = \omega_{r,m1} \tag{7b}$$

$$\dot{\theta}_\Delta = \omega_r - \left(\frac{1}{N_g} \right) \omega_g \tag{7c}$$

$$\dot{\omega}_g = \left(\frac{\eta_{dt} B_{dt}}{N_g J_g} \right) \omega_r + \left(\frac{-\frac{\eta_{dt} B_{dt}}{N_g^2} - B_g}{J_g} \right) \omega_g + \left(\frac{\eta_{dt} K_{dt}}{N_g J_g} \right) \theta_\Delta - \left(\frac{1}{J_g} \right) \tau_g \tag{7d}$$

Whether it is possible or not to compute the specified variables from the corresponding equations depends naturally on the properties of the equations. Equally important are however prerequisites in terms of *causality assumption*, i.e., regarding integral and/or derivative causality, and the properties of the *computational tools*, that are available for use, for a detailed discussion see, e.g., Svärd and Nyberg (2010). The computation sequence (6) makes use of solely integral causality when the variables θ_Δ and ω_g are computed using equations e_{14} and e_{12}, respectively.

Sequential Residual Generator

Having computed the unknown variables in $V_1 \bigcup V_2 \bigcup \ldots \bigcup V_k$ according to the computation sequence $\mathcal{C}$ in (5), a residual can be obtained by evaluating a redundant equation e, i.e., $e \in E \setminus E_1 \bigcup E_2 \ldots \bigcup E_k$ with $\mathrm{var}_X(e) \subseteq \mathrm{var}_X(E_1 \bigcup E_2 \ldots \bigcup E_k)$, where the operator

$\mathrm{var}_X(\cdot)$ returns the unknown variables that are contained in an equation set. A residual generator based on a computation sequence $\mathcal{C}$ and redundant equation e is referred to as a *sequential residual generator*.

The computation sequence (6), together with equation e_{26} constitute a sequential residual generator for the wind turbine model. When all variables in the computation sequence (6) have been computed according to (7), the residual is computed as $r = \omega_{g,m1} - \omega_g$.

Finding Sequential Residual Generators

Regarding implementation aspects, e.g., complexity and computational load, it is unnecessary to compute variables that are not contained in the residual equation, or not used to compute any of the variables contained in the residual equation. Furthermore, it is also desirable that computation of variables in each step is performed from as small equation sets as possible. It can be shown, see Svärd and Nyberg (2010), that the equations in a computation sequence fulfilling the above properties, together with a redundant residual equation, in fact correspond to a Minimal Structurally Overdetermined (MSO) set, see Krysander et al. (2008). In other words, a necessary condition for the existence of a sequential residual generator for a model is that the model, or a sub-model, is an MSO set.

4.2 Candidate Residual Generators

As indicated above, a first step when searching for a sequential residual generator for a model may be to find an MSO set in the model. Thus, an MSO set can be regarded as a *candidate residual generator*. There are efficient algorithms for finding all MSO sets in large equation sets, see, e.g., Krysander et al. (2008).

Consider now the model of the wind turbine described in Section 2.4, with equations $E = \{e_1, e_2, \ldots, e_{33}\}$, unknown variables

$$\begin{aligned} X = \big\{ &\tau_r, \beta_1, \lambda, v_w, \beta_2, \beta_3, \omega_r, x_{\beta_{11}}, x_{\beta_{12}}, \beta_{1,r}, x_{\beta_{21}}, x_{\beta_{22}}, \\ &\beta_{2,r}, x_{\beta_{31}}, x_{\beta_{32}}, \beta_{3,r}, \omega_g, \theta_\Delta, \tau_g, x_{\tau_g}, P_g \big\}, \end{aligned}$$

and known, i.e., measured, variables

$$\begin{aligned} Y = \big\{ &\beta_r, \tau_{g,r}, \beta_{1,m1}, \beta_{1,m2}, \beta_{2,m1}, \beta_{2,m2}, \beta_{3,m1}, \\ &\beta_{3,m2}, \omega_{r,m1}, \omega_{r,m2}, \omega_{g,m1}, \omega_{g,m2}, v_{w,m}, \tau_{g,m}, P_{g,m} \big\}. \end{aligned}$$

In summary, the model contains 33 equations, 21 unknown variables, and 15 known variables. By utilizing the *structure*, i.e., which unknown variables are contained in which equation, see, e.g., Blanke et al. (2006), and a Matlab© implementation of the algorithm presented in Krysander et al. (2008), 1058 MSO sets were found in total.

5 Selecting Residual Generators

It is not feasible to implement and use all 1058 candidate residual generators, i.e., MSO sets, in the final FDI-system. A more attractive approach is instead to pick, from the set of all candidate residual generators, a smaller set of residual generators with desired properties.

5.1 Desired Properties of Residual Generators

The desired properties of the sought set of residual generators are:

1. the set of residual generators should enable us to isolate all single faults from each other;
2. a set of residual generators of smaller cardinality is preferred before a larger one, given that the two sets have equal isolability properties;
3. a residual generator based on an MSO set of smaller cardinality is preferred before a residual generator based on an MSO set of larger cardinality, given that the two sets have equal detectability and isolability properties.

Properties 2 and 3 are mainly motivated by implementation aspects such as complexity, computational load, and numerical issues.

We will base the selection of residual generators on quantitative, structural, properties of the MSO sets instead of more qualitative or analytical properties on the actual residual generators. The latter may result in better isolation performance but is considered intractable since it require that residual generators are implemented, executed and evaluated, and also access to representative measurement data for all fault cases.

5.2 Fault Detectability and Isolability

To be able to formally state the selection problem, the notions of detectability and isolability are needed. Assuming that each fault occurs in only one equation, let e_{f_i} denote the equation in an equation set E containing fault f_i, for example $e_{\Delta\beta_{1,m1}} = e_{18}$, see Section 2. Note that if a fault f_j occurs in more than one equation, the fault f_j can be replaced with a new variable x_{f_j} in these equations, and the equation $x_{f_j} = f_j$ added to the equation set. This added equation will then be the only equation where f_j occurs. To proceed, let $(\cdot)^+$ denote an operator extracting the overdetermined part of a set of equations. According to Krysander and Frisk (2008), a fault f_i is *structurally detectable* in the equation set E if $e_{f_i} \in (E)^+$ and *structurally isolable* from fault f_j in the equation set E if $e_{f_i} \in (E)^+$ and $e_{f_j} \notin (E)^+$.

For an example, consider the equation set $M = \{e_{26}, e_{29}, e_{24}, e_{14}, e_{12}\}$ containing the residual equation and equations from the computation sequence (5), studied in Section 4.1. First we note that the equation set M is an MSO set due to the property of sequential residual generators mentioned in Section 4.1. Further, since M is an MSO set, it holds that $(M)^+ = M$, see for example Krysander et al. (2008). Thus, it can for instance be deduced that fault $\Delta\omega_g$ is structurally isolable from fault $\Delta\beta_{1,m1}$ in M, since $e_{\Delta\omega_g} = e_{12}$, $e_{\Delta\beta_{1,m1}} = e_{18}$, and it holds that $e_{12} \in M$ and $e_{18} \notin M$, see Section 2.4.

By again utilizing the structure of the wind turbine model, the structural isolability properties of the model were calculated. All considered faults, see Section 2.2, can be (structurally) isolated from each other in the wind turbine model.

5.3 Selection Problem Formulation

We will now formulate the selection problem in terms of properties on a set of MSO sets. To this end, let $\mathcal{M}$ denote the set of all MSO sets in the model, and F the set of considered faults. Let $f_i, f_j \in F$ and define the *isolation class* for (f_i, f_j) as

$$I_{f_i f_j} = \left\{ S \in \mathcal{M} : e_{f_i} \in (S)^+ \wedge e_{f_j} \notin (S)^+ \right\}, \tag{8}$$

that is, $I_{f_i f_j}$ contains the MSO sets in $\mathcal{M}$ in which fault f_i is structurally isolable from fault f_j. Further, let

$$\mathcal{I} = \left\{ I_{f_i f_j} : \forall \left(f_i, f_j\right) \in F \times F, f_i \neq f_j \right\} \tag{9}$$

denote the set of all isolation classes needed for full isolation of all faults in F. For the wind turbine benchmark model and the set of 15 faults considered in Section 2.2, the set $\mathcal{I}$ contains in total $15 \times 15 - 15 = 210$ isolation classes for single fault isolation of all 15 faults, i.e., $|\mathcal{I}| = 210$, where the operator $|\cdot|$ returns the cardinality of a set.

To be able to satisfy the isolability property 1 stated above, we want to find a set $\mathcal{S} \subseteq \mathcal{M}$ with a non-empty intersection with all isolation classes, that is,

$$\forall I_{f_i f_j} \in \mathcal{I} \quad \mathcal{S} \cap I_{f_i f_j} \neq \varnothing. \tag{10}$$

The property (10) on $\mathcal{S}$ implies that we should find a so called *hitting set* for $\mathcal{I}$. To satisfy the property 2 we want to find an $\mathcal{S}$ so that $|\mathcal{S}|$ is minimized. Thus, the sought hitting set for $\mathcal{I}$ should be of minimal cardinality and we should find a so called *minimal cardinality hitting set* (MHS) for $\mathcal{I}$.

There are several possibilities for a metric that helps us find an $\mathcal{S}$ that satisfies property 3. We opt for simplicity and have therefore chosen to minimize $\sum_{S \in \mathcal{S}} |S|$. As an additional requirement, on top of 1, 2, and 3 in Section 5.1 we require that at least one residual generator can be constructed from every $S \in \mathcal{S}$.

5.4 Solving the Selection Problem

The problem of finding a minimal cardinality hitting set is known to be NP-hard, see, e.g., Garey and Johnson (1979). To overcome the complexity issues, we have chosen to compute an approximate solution to the problem in an iterative manner with a greedy selection approach as elaborated in Svärd et al. (2011).

To accomplish this, we need to specify a *utility function*, i.e., a function that evaluates the usefulness of a given MSO set, and also state the properties of a complete solution to the selection problem. Following the greedy selection approach, we add to the solution the MSO set with the largest utility until the solution is complete. Furthermore, we only add MSO sets from which at least one residual generator can be constructed.

Characterization of a Solution

We will now characterize a complete solution to the selection problem for use in the selection algorithm. First, we define the *isolation class coverage* of a set of MSO sets $\mathcal{S} \subseteq \mathcal{M}$ as

$$\sigma_{\mathcal{I}}(\mathcal{S}) = \left\{ I_{f_i f_j} \in \mathcal{I} : \exists S \in \mathcal{S}, S \in I_{f_i f_j} \right\}, \tag{11}$$

which states which of the isolation classes in $\mathcal{I}$ that are covered by the MSO sets in $\mathcal{S}$. The property 1 in Section 5.1, i.e., the isolation or hitting set property, can with the isolation class coverage notion be formulated as $\sigma_{\mathcal{I}}(\mathcal{S}) = \mathcal{I}$. This characterizes a complete solution of the selection problem.

Utility Function

To evaluate a specific MSO set, we want to take into account the properties 1, 2, and 3, above. For a given MSO set S, we will use the utility function

$$\mu_{\mathcal{I}}(S) = \gamma \left(\frac{|\sigma_{\mathcal{I}}(\{S\})|}{|\mathcal{I}|} \right) + (1-\gamma)\left(1 - \frac{|S|}{|\hat{S}|}\right), \tag{12}$$

where $\hat{S}$ is the MSO set in $\mathcal{M}$ with largest cardinality, and γ, $0 \le \gamma \le 1$, a weighting factor. The term $\frac{|\sigma_{\mathcal{I}}(\{S\})|}{|\mathcal{I}|}$ in (12) tells how many of the isolation classes in $\mathcal{I}$ that are covered by the MSO set S. Since we aim at covering all isolation classes with a minimum of MSO sets, property 2, we want to pick an MSO set that maximizes this term. The term $1 - \frac{|S|}{|\hat{S}|}$ relates the cardinality of S to the cardinality of all other sets in $\mathcal{M}$. Picking an MSO set that maximizes this term in (12) hence corresponds to picking the MSO set with smallest cardinality in $\mathcal{M}$. This will help us satisfy property 3. The weighting factor γ is used to trade between the two properties reflected by these two terms.

Note that an MSO set maximizing one term in (12) may minimize the other since an MSO set of larger cardinality likely cover more isolation classes than an MSO set of smaller cardinality.

5.5 The Selection Algorithm

The function selectResidualGenerators used for selecting residual generators by means of greedy selection is given in Algorithm 4. Input to the function is a set of MSO sets $\mathcal{M}$, i.e., a set of candidate residual generators, and a set of isolation classes $\mathcal{I}$. The output is a set of MSO sets $\mathcal{S} \subseteq \mathcal{M}$ and a set of residual generators $\mathcal{R}$ based on $\mathcal{S}$. The function findComputationSequence, described in Svärd and Nyberg (2010), is used to find a computation sequence in accordance with Section 4.1, given a just-determined set of equations. The function findComputationSequence can be found in Algorithm 5 in Appendix A.

For a formal discussion regarding the qualification of using a greedy heuristic for solving the residual generation selection problem, as well as the complexity properties of such algorithms, please refer to Svärd et al. (2011) and references therein.

Algorithm 4 Greedy Selection of Residual Generators

```
function selectResidualGenerators($\mathcal{M}, \mathcal{I}$)
    $\mathcal{S} := \varnothing$
    $\mathcal{R} := \varnothing$
    while $\mathcal{I} \neq \varnothing$ do
        $S := \arg\max_{S \in \mathcal{M}} \mu_{\mathcal{I}}(S)$
        $x := \mathrm{var}_X(S)$
        $R := \varnothing$
        for all $e \in S$ do
            $S' := S \setminus \{e\}$
            $\mathcal{C} :=$ findComputationSequence($S', x$)
            if $\mathcal{C} \neq \varnothing$ then
                $R := R \cup \{(\mathcal{C}, e)\}$
            end if
        end for
        if $R \neq \varnothing$ then
            $\mathcal{S} := \mathcal{S} \cup \{S\}$
            $\mathcal{R} := \mathcal{R} \cup \{R\}$
        end if
        $\mathcal{M} := \mathcal{M} \setminus \{S\}$
        $\mathcal{I} := \mathcal{I} \setminus \sigma_{\mathcal{I}}(\{S\})$
    end while
    return $(\mathcal{S}, \mathcal{R})$
end function
```

Selecting Residual Equation

Note that the total number of sequential residual generators that potentially can be constructed from an MSO set equals the number of equations in the set. All residual generators created from the same MSO set however have equal fault detectability and isolability properties according to Section 5.2. Nevertheless, their actual fault detectability and isolability may differ due for example different sensitivity for noise, etc. To make the final selection of which of the residual generators created from an MSO set that should be included in the final diagnosis system, evaluation by means on execution using real measurements from different fault cases is needed. Since we in this work only assume that no-fault data is available, see Section 3, this is not possible.

In this work, the selection of which residual generator to create from a given MSO set is done so that the final deployment of the FDI-system becomes as simple as possible. First of all, findComputationSequence was configured to prefer algebraic equations as residuals before differential equations, if possible. Second, in order to avoid implementation issues related to numerical differentiation, findComputationSequence was configured to prefer computation sequences using integral causality. Using this two-step heuristic, the selection of which residual generator to create from an MSO set, in practice, is more or less unambiguous. In those few cases where more than one candidate remains, we make an arbitrary selection.

5.6 Selected Residual Generators

Both functions selectResidualGenerators and findComputationSequence were implemented in Matlab©. As computational tool, see Svärd and Nyberg (2010), the algebraic equation solver Maple© was utilized, which allows symbolic solving of algebraic loops. The input to the algorithm was the set of all 1058 MSO sets for the wind-turbine benchmark model, see Section 4.2, and the set of all 210 isolation classes for single fault isolation of all considered faults, see Sections 2.2 and 5.3.

To investigate the sensitivity of selectResidualGenerators to the parameter γ, i.e., the trade-off between properties 2 and 3 stated in Section 5.3 and reflected by $|S|$ and $\sum_{S\in\mathcal{S}}|S|$, the algorithm was run with the wind turbine model and $0 \leq \gamma \leq 1$. The result is shown in Table 2, where $\mathcal{S}$ denotes the set returned by selectResidualGenerators. When $\gamma = 1$ the aim is to fulfill the isolation property with as few MSO sets as possible, no matter the size of the MSO sets. As seen in Table 2 this results in few, but large, MSO sets. The smaller the γ, the more attention is paid to the size of the MSO sets. It turns out that $0.1 \leq \gamma \leq 0.6$ gives a decent trade-off between $|\mathcal{S}|$ and $\sum_{S\in\mathcal{S}}|S|$ for the wind turbine model.

With $\gamma = 0.5$, the algorithm selected 16 MSO sets, i.e., $|\mathcal{S}| = 16$, and $\sum_{S\in\mathcal{S}}|S| = 61$. Of the 16 selected MSO sets, 7 contain algebraic equations only. The other 9 MSO sets contain both algebraic and differential equations. Thus, 7 of the 16 residual generators used in the final FDI-system are static and the remaining 9 are dynamic. All 9 dynamic residual generators, due to the configuration of the algorithm, use integral causality. The *total* number of found residual generators is 34, that is, $|\mathcal{R}| = 34$, see Section 5.5. Of these 34 residual generators, 18 are static and the remaining 16 are dynamic.

Table 2: selectResidualGenerators sensitivity to parameter γ.

γ	$\lvert\mathcal{S}\rvert$	$\sum_{S \in \mathcal{S}} \lvert S \rvert$
0.0	20	82
0.1	16	61
0.2	16	61
0.3	16	61
0.4	16	61
0.5	16	61
0.6	16	61
0.7	16	65
0.8	17	72
0.9	16	87
1.0	8	108

Fault Signature Matrix

Given an MSO set S its *fault signature* $F(S)$, with respect to the faults in F, is defined as

$$F(S) = \left\{ f_i \in F : e_{f_i} \in S \right\}.$$

For instance, the fault signature of the MSO set $S_1 = \{e_{26}, e_{27}\} \subseteq \mathcal{M}$ is $F(S_1) = \left\{\Delta\omega_{g,m1}, \Delta\omega_{g,m2}\right\}$. A convenient representation of the fault signature of a set of MSO sets $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$ with respect to F is the *fault signature matrix* (FSM) S with elements defined by

$$S_{ij} = \begin{cases} \text{x}, & \text{if } f_j \in F(S_i), S_i \in \mathcal{M} \\ 0, & \text{else.} \end{cases}$$

The FSM for the 16 MSO sets on which the selected residual generators are based, is given in Table 3.

6 Fault Detection and Isolation

For fault detection and isolation, diagnostic tests based on the output from each of the 16 residual generators are constructed. Since no assumptions are made regarding the nature of the faults that should be detected, see Section 2.2, nothing is known about the fault's temporal properties, size, rate of occurrence, etc. Hence, we may not be able to fully exploit the potential of some general method for change detection as for example the CUSUM-test, see, e.g., Gustafsson (2000).

As said in Section 3 we however assume that no-fault training data is available. To take advantage of this fact, and also handle uncertainties in terms of modeling errors and measurement noise, we base our diagnostic tests on a comparison of the estimated probability distributions of no-fault and current residuals. The former probability distributions are estimated offline using the available no-fault training data and the latter

Table 3: Fault Signature Matrix

	$\Delta\beta_1$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\omega_g$	$\Delta\tau_g$	$\Delta\beta_{1,m1}$	$\Delta\beta_{1,m2}$	$\Delta\beta_{2,m1}$	$\Delta\beta_{2,m2}$	$\Delta\beta_{3,m1}$	$\Delta\beta_{3,m2}$	$\Delta\omega_{r,m1}$	$\Delta\omega_{r,m2}$	$\Delta\omega_{g,m1}$	$\Delta\omega_{g,m2}$
R_1 (S_1)														X	X
R_2 (S_2)												X	X		
R_3 (S_3)										X	X				
R_4 (S_4)								X	X						
R_5 (S_5)						X	X								
R_6 (S_8)					X										
R_7 (S_{11})				X									X		X
R_8 (S_{27})			X								X				
R_9 (S_{29})		X							X						
R_{10} (S_{31})	X						X								
R_{11} (S_7)														X	
R_{12} (S_6)															X
R_{13} (S_{14})				X								X		X	
R_{14} (S_{28})			X							X					
R_{15} (S_{30})		X						X							
R_{16} (S_{32})	X					X									

online using current data. A clear advantage with this approach is that changes in mean and variance are handled in a unified way, since we consider the complete distribution of the residual.

6.1 Diagnostic Test Design

Let P^{NF} be a discrete estimate of the probability distribution of a residual from no-fault data, and P a discrete estimate of the distribution of the same residual from present data, both having n bins. Then the Kullback-Leibler (K-L) divergence, (Kullback and Leibler, 1951), between P and P^{NF} is given by

$$D\left(P\|P^{NF}\right) = \sum_{j=1}^{n} P\left(j\right) \log \frac{P\left(j\right)}{P^{NF}\left(j\right)}, \tag{13}$$

where $P\left(j\right)$ denotes the j:th bin of the discrete distribution P.

To apply the K-L divergence for construction of a diagnostic test, we proceed as follows. Given a representative batch of no-fault data $\mathcal{Z}^{NF}$, i.e., in our case measurements of the variables in the set Z which contains the inputs and outputs to the model, we run the set of residual generators and obtain a set of residuals. For each residual r_i, we then estimate its probability distribution and obtain P_i^{NF}, i.e., actually $P_i^{NF} \approx P\left(R_i|\mathcal{Z}^{NF}\right)$ where R_i is a stochastic variable, discretized in n bins, representing residual r_i. As said, this procedure can be done off-line. To estimate a probability distribution, we create a normalized histogram with n bins for the data from which the distribution should be estimated.

On-line, we continuously estimate the distribution of the current residual r_i using a sliding window containing N samples of r_i. If we by P_i^t denote the estimated distribution of r_i calculated at time t, i.e., $P_i^t \approx P\left(R_i|\mathcal{Z}^t\right)$, where $\mathcal{Z}^t$ denotes the batch of data in the sliding window at time t, the diagnostic test is designed as

$$T_i(t) = \begin{cases} 1, & \text{if } D\left(P_i^t \| P_i^{NF}\right) \geq J_i, \\ 0, & \text{else}, \end{cases} \tag{14}$$

where J_i is the threshold for alarm. The K-L divergence $D\left(P_i^t \| P_i^{NF}\right)$ is referred to as the test quantity of the diagnostic test T_i.

6.2 Fault Isolation Strategy

Due to uncertainties not captured by the given model nor present in the no-fault training data, the power of diagnostic tests are not ideal for all faults. That is, the probability of detection given a certain fault is not always 1. To take this into account, the isolation scheme will interpret an "x" in a certain row in Table 3 as if the test *may* respond if the corresponding fault occurs and consequently no conclusions are drawn if a test does not respond, see Nyberg (1999).

To obtain the total diagnosis statement from a set of alarming diagnostic tests, we simply match their fault signatures with the FSM given in Table 3. For example, if only test T_{10} alarms, we look at the row corresponding to R_{10} and conclude that either fault $\Delta\beta_1$ *or* $\Delta\beta_{1,m2}$ are present. If then also T_{16} alarms, we combine the row corresponding to R_{16} with the row corresponding to R_{10} and conclude that fault $\Delta\beta_1$ must be present.

To handle also multiple faults, we use the fault signatures in the original FSM in Table 3 to create an extended FSM with fault signatures also for multiple faults. This is done by column-wise OR-operations in the original FSM. For instance, the column in the FSM for the double fault $\Delta\omega_{g,m1} \wedge \Delta\omega_{g,m2}$ will get "x" in rows corresponding to R_1, R_7, R_{11}, R_{12}, and R_{13} and zeros elsewhere. In the fault isolation scheme, we first attempt to isolate all single faults using the original FSM in Table 3. If this does not succeed, we try to isolate double faults, and so forth.

7 Implementation Details

The final FDI-system was implemented in Simulink© according to the structure in Figure 2. The 16 residual generators were implemented as Embedded Matlab Functions (EMF) in which the code was automatically generated from the structures obtained from the functions findComputationSequence and findResidualGenerators. The initial conditions for the states in the dynamic residual generators were derived from the corresponding sensor measurements, if available, otherwise set to zero. For instance, $\theta_\Delta(t_0) = 0$, $x_{\beta_{i1}}(t_0) = \frac{\beta_{i,m1}(t_0)+\beta_{i,m2}(t_0)}{2}$, and $\omega_g(t_0) = \frac{\omega_{g,m1}(t_0)+\omega_{g,m2}(t_0)}{2}$. This may cause transients in the residuals, but this is not considered a problem.

7.1 Parameter Discussion

Although the aim is to keep the number of parameters in the automated design method at a minimum, there are nevertheless some parameters that must be set. This section lists the needed parameters and discusses their influence on the performance of the FDI-system.

Number of Histogram Bins and Size of Sliding Window

The number of bins n in the histograms used as distribution estimates, is a trade-off between detection time, noise sensitivity, and complexity, in terms of computational power and memory. A large n results in fast detection, but on the other hand also in increased sensitivity for noise. Also, a large n requires more memory and involves more computations, in comparison with a smaller n.

The size N of the sliding window used to batch data for creation of the histograms is a trade-off between detection performance, noise sensitivity, and complexity. A large N will give the K-L test quantity low-pass characteristics, resulting in a smoothed K-L test quantity. This makes it possible to detect small changes in the estimated distributions. On the other hand, a large N requires more memory. The choice of N is also related to the number of bins n in the histograms and vice versa, since a small N together with a large n, will result in a sparse histogram. Hence, the choices of N and n must match.

For the wind turbine benchmark model, investigations however indicate that the method is quite insensitive to the values of n and N if $15 \leq n \leq 50$ and $2000 \leq N \leq 6000$. A decent trade-off, taking this into account, but also the complexity issues discussed above, is $n = 20$ and $N = 3000$, which are the values used in the final FDI-system.

Alarm Thresholds

The choice of alarm thresholds J_i, $i = 1, 2, \ldots, 16$, is a trade-off between detection time and the number of false detections. The higher the thresholds, the longer the detection time and the lower the rate of false alarms. The choice of alarm thresholds is related to the choices of n and N since both affect how sensitive a K-L test quantity is to noise, which in turn affects the rate of false detections. We aim at choosing the alarm thresholds so that the number of false detections is minimized, implying that the choice of J_i must match the choices of n and N. For the wind turbine benchmark model, the alarm thresholds were computed as a safety factor $\alpha = 1.1$ times the maximum value of the corresponding K-L test quantities from 100 simulations with no-fault data.

Isolation Validation Time

The only parameter involved in the fault isolation is the isolation validation time t_I^{val}. This parameter is used to compensate for the fact that the power of diagnostic tests not is ideal, see Section 6.2. This may for example result in that the detection times, for the same fault, are different for different diagnostic tests. To handle this, we demand that the output from the isolation has been equal for t_I^{val} samples before reporting the isolation result. By choosing a large t_I^{val}, we decrease the probability of false isolation, but on

Table 4: Fault Sequence

Fault	Time (s)	Description
$\Delta\omega_{r,m2}$	1000 - 1100	$\omega_{r,m2} = 1.1\omega_{r,m2}$
$\Delta\omega_{g,m2}$	1000 - 1100	$\omega_{g,m2} = 0.9\omega_{g,m2}$
$\Delta\omega_{r,m1}$	1500 - 1600	$\omega_{r,m1} = 1.4$ rad/s
$\Delta\beta_{1,m1}$	2000 - 2100	$\beta_{1,m1} = 5°$
$\Delta\beta_{2,m2}$	2300 - 2400	$\beta_{2,m2} = 1.2\beta_{2,m2}$
$\Delta\beta_{3,m1}$	2600 - 2700	$\beta_{3,m1} = 10°$
$\Delta\beta_2$	2900 - 3000	$\omega_n = \omega_{n2}$, $\zeta = \zeta_2$
$\Delta\beta_3$	3400 - 3500	$\omega_n = \omega_{n3}$, $\zeta = \zeta_3$
$\Delta\tau_g$	3800 - 3900	$\tau_g = \tau_g + 2000$ Nm

the other hand increase the isolation time. For the wind turbine benchmark model, the isolation validation time t_I^{val} was set to 4 samples.

8 Evaluation and Results

To evaluate the performance of the proposed FDI-system, we use the test cases described in Fogh Odgaard et al. (2009). The test cases are based on measured wind data and a sequence of injected faults. The set of injected faults, their time of occurrence and description, is specified in Table 4. The sequence contains 5 sensor faults and 3 actuator faults. Note that two faults are injected at 1000-1100 s, i.e., at this time we have the double fault $\Delta\omega_{r,m2} \wedge \Delta\omega_{g,m2}$.

The no-fault distributions used in the evaluation were estimated from residual data stemming from 100 Monte Carlo simulations with no-fault data, i.e., inputs, corresponding to the measured variables in Z. Each set of no-fault data was generated with the provided wind turbine model with different noise realizations according to the model.

8.1 Results and Analysis

By means of Monte Carlo simulations, the FDI-system was simulated 100 times with data from the provided wind turbine model set-up according to the above described test sequence.

Based on the results from the 100 runs, the mean time of detection $\overline{T}_D$, maximum time of detection T_D^{max}, minimum time of detection T_D^{min}, mean time of isolation $\overline{T}_I$, minimum time of isolation T_I^{min}, the total number of missed detections MD, and the total number of false detections FD, for each of the faults in the test sequence, were computed. The results along with the specified detection requirements (Fogh Odgaard et al., 2009), given in the row Req., are shown in Table 5, where all time values are given in seconds. Note that the specified requirements concern detection, and not isolation.

Table 5: FDI Results. Time values in seconds.

	$\Delta\omega_{r,m2}$ $\Delta\omega_{g,m2}$	$\Delta\omega_{r,m1}$	$\Delta\beta_{1,m1}$	$\Delta\beta_{2,m2}$	$\Delta\beta_{3,m1}$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\tau_g$
Req.	0.1	0.1	0.1	0.1	0.1	0.08	6	0.05
$\overline{T}_D$	0.040	0.16	0.058	4.30	0.069	51.57	18.1	7.94
T_D^{max}	0.04	0.27	0.07	6.10	0.07	51.88	19.05	7.98
T_D^{min}	0.03	0.06	0.05	0.40	0.06	50.57	16.37	7.90
$\overline{T}_I$	-	2.53	0.12	88.85	0.13	56.95	31.84	7.99
T_I^{max}	-	3.13	0.12	114.26	0.13	120.73	111.96	8.03
T_I^{min}	-	1.89	0.11	13.17	0.12	51.62	17.91	7.95
MD	0	0	0	0	0	0	0	0
FD	0	0	0	0	0	0	0	0

According to the row corresponding to T_D^{max} in Table 5, all faults in the test sequence could be detected. For faults $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$, $\Delta\beta_{1,m1}$, $\Delta\beta_{3,m1}$ detection requirements are met, by means of both $\overline{T}_D$ and T_D^{max}.

All faults, except the double fault $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$ could also be isolated. However, the mean time of isolation, $\overline{T}_I$, for some faults, e.g., $\Delta\beta_{2,m2}$, is substantially longer than the corresponding mean time of detection. The main reason for this is that some tests respond slower to faults than other. As said, fault $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$ could not be isolated. In fact, this fault is not uniquely isolable with the isolation strategy described in Section 6.2 since the test response of fault $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$ is a subset of the test response of fault $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m1}$, see Table 3. Both faults $\Delta\omega_{g,m2}$ and $\Delta\omega_{r,m2}$ are however contained in the diagnosis statement computed after the faults have been detected.

It seems like sensor faults, e.g., $\Delta\beta_{3,m1}$ tend to be easier to detect than actuator faults as for example $\Delta\tau_g$ and $\Delta\beta_2$. One possible explanation may be that actuator faults in general cause changes in dynamics, whose effects are attenuated by modeling errors, noise, etc.

As can be seen in the last two rows of Table 5, there are no missed or false detections in any of the 100 test runs.

8.2 Case Study of Fault $\Delta\omega_{r,m1}$

To study in more detail how the FDI-system handles faults, we consider the sensor fault $\Delta\omega_{r,m1}$. The fault corresponds to a fixed value of 1.4 rad/s being measured by sensor $\omega_{r,m1}$ and occurs at time $t = 1500$ s. According to the FSM in Table 3, the residuals sensitive to fault $\Delta\omega_{r,m1}$ are r_2 and r_{13}, obtained as output from the residual generators R_2 and R_{13}, respectively. These residuals along with the corresponding K-L test quantities are shown in Figure 4. As can be seen, both the residuals and the test quantities respond distinctively to the fault.

To also illustrate the isolation procedure, we show in Figure 5 the result of the diagnostic tests T_2 and T_{13} (top), the isolation result associated to faults $\Delta\omega_{r,m1}$ (middle)

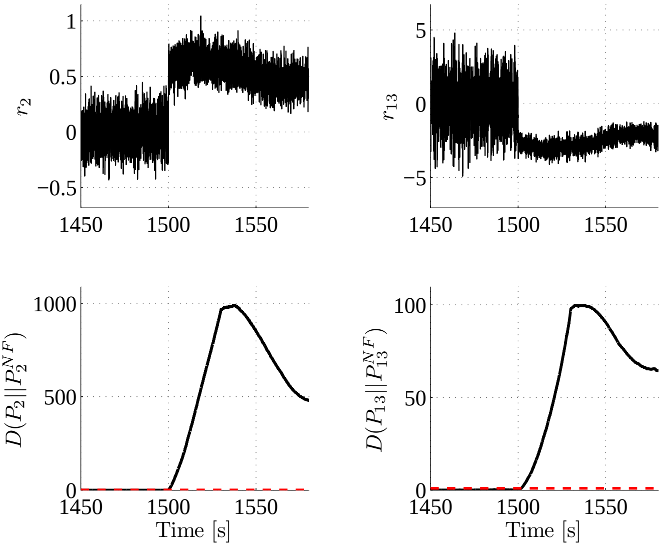


Figure 4: Affected residuals r_2 (top-left) and r_{13} (top-right), and the corresponding K-L test quantities $D\left(P_2^t\|P_2^{NF}\right)$ (bottom-left) and $D\left(P_{13}^t\|P_{13}^{NF}\right)$ (bottom-right) at the time of occurrence of fault $\Delta\omega_{r,m1}$.

and $\Delta\omega_{r,m2}$ (bottom), and also the signal that indicates when the isolation procedure is done (middle and bottom). As can be seen in Figure 5, the first test that reacts to the fault is T_2. This occurs at $t = 1500.23$ s. Since T_2 is sensitive to both fault $\Delta\omega_{r,m1}$ and $\Delta\omega_{r,m2}$ and no other test has alarmed, the diagnosis statement is that either $\Delta\omega_{r,m1}$ or $\Delta\omega_{r,m2}$ may be present, and no fault can be isolated. At $t = 1502.55$ s, test T_{13} alarms. Test T_{13} is sensitive to faults $\Delta\omega_g$, $\Delta\omega_{r,m1}$, and $\Delta\omega_{r,m2}$, and the updated total diagnosis statement based on that both T_2 and T_{13} have alarmed thus becomes $\Delta\omega_{r,m1}$, see Table 3. This occurs at time $t = 1502.59$ s.

9 Conclusions

We have proposed an FDI-system for the wind turbine benchmark designed by application of a generic automated design method, in which the number of required human decisions and assumptions are minimized. No specific adaptation of the method for the wind turbine benchmark was needed. The method contains in essence three steps: generation of candidate residual generators; residual generator selection; and diagnostic test construction. The second step is done by means of greedy selection, and the third step is based on a novel method utilizing the K-L divergence.

The performance of the proposed FDI-system has been evaluated using the predefined test sequence for the wind turbine benchmark. The FDI-system performs well; all faults in the test sequence were detected within feasible time and all faults, except a

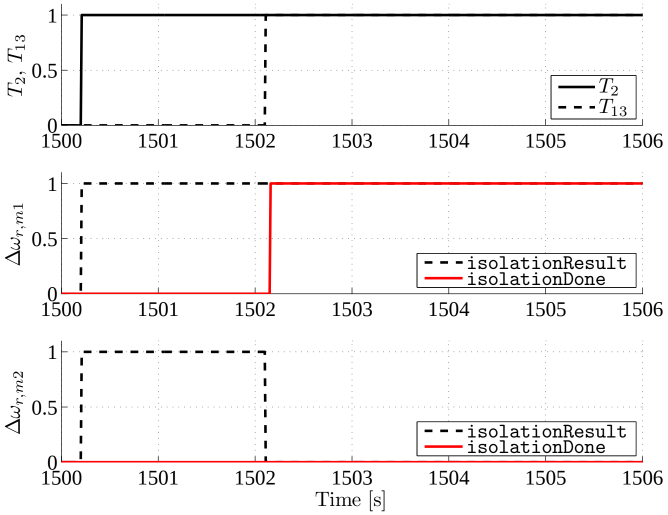


Figure 5: Isolation procedure for fault $\Delta\omega_{r,m1}$. Top figure shows diagnostic tests T_2 and T_{13}. Middle and bottom figures show the isolation result corresponding to faults $\Delta\omega_{r,m1}$ and $\Delta\omega_{r,m2}$, respectively, and when the isolation procedure is done.

double fault, could be isolated shortly thereafter. In addition, there are no false or missed detections. A tailor-made, finely tuned, FDI-system for the benchmark would probably perform better. However, in relation to the required design effort, and that no specific adaptation or tuning of the method to the benchmark was done, the performance is satisfactory.

Acknowledgment

This work was supported by Scania CV AB, Södertälje, Sweden.

A Algorithm for Finding a Computation Sequence

To make the paper more self-contained, the function findComputationSequence described in Svärd and Nyberg (2010) is given below as Algorithm 5. The function takes a just-determined equation set $E' \subseteq E$ and a set of unknown variables $X' \subseteq X$, and returns an ordered set $\mathcal{C}$ as output. The algorithm assumes availability of a computational tool in the form of a algebraic equation (AE) solver such as for example Maple, see Svärd and Nyberg (2010) for a thorough discussion regarding this. The function findAllSCCs is assumed to return an ordered set of equation and variable pairs, where each pair corresponds to a strongly connected component (SCC) of the structure of the equation set with respect to the variable set. There are efficient algorithms for finding SCCs in directed graphs, for example the DM-decomposition (Dulmage and Mendelsohn, 1958).

In Matlab, the DM-decomposition is implemented in the function `dmperm`. Other functions used in findComputationSequence are:

- Diff and unDiff, takes a variable set as input and returns its differentiated and undifferentiated correspondence.
- isInitCondKnown determines if the initial conditions of the given variables are known and consistent, and the function isDifferentiable determines if the given variables can be differentiated with the available differentiation tool.
- isJustDetermined is used to determine if the structure of the given equation set, with respect to the given variable set, is just-determined. This is essential, since otherwise the computation of SCCs makes no sense.
- getDifferentialEquations takes a set of equations and a set of differentiated variables as input, and returns the differential equations in which the given differentiated variables are contained.
- isToolSolvable determines if the available algebraic equation solver can solve the given equations for the given set of variables.
- Append, takes an ordered set and an element as input and simply appends the element to the end of the set.
- The operator $|\cdot|$, taking a set as input, is assumed to return the number of elements in the set and the notion $A\left(i\right)$ is used to refer to the i:th element of the ordered set A.

Algorithm 5 Find a Computation Sequence

```
function findComputationSequence($E', X'$)
    $\mathcal{C} := \varnothing$
    $S :=$ findAllSCCs($E', X'$)
    for $i = 1, 2, \ldots, |S|$ do
        $(E_i, X_i) := S\,(i)$
        $D_i :=$ Diff($X_i$)
        $Z_i := \mathrm{var}_D(E_i) \cap D_i$
        $W_i := X_i \setminus$ unDiff($Z_i$)
        if not isInitCondKnown($Z_i$) then
            return $\varnothing$
        end if
        $E_{Z_i} :=$ getDifferentialEquations($E_i, Z_i$)
        $E_{W_i} := E_i \setminus E_{Z_i}$
        $S_{Z_i} :=$ findAllSCCs($E_{Z_i}, Z_i$)
        for $j = 1, 2, \ldots, |S_{Z_i}|$ do
            $\left(E^j_{Z_i}, Z^j_i\right) := S_{Z_i}\,(j)$
            if isToolSolvable($Z^j_i, E^j_{Z_i}$) then
                Append($\mathcal{C}, \left(Z^j_i, E^j_{Z_i}\right)$)
            else
                return $\varnothing$
            end if
        end for
        if isJustDetermined($E_{W_i}, W_i$) then
            $S_{W_i} :=$ findAllSCCs($E_{W_i}, W_i$)
            for $j = 1, 2, \ldots, |S_{W_i}|$ do
                $\left(E^j_{W_i}, W^j_i\right) := S_{W_i}\,(j)$
                if isToolSolvable($W^j_i$,$E^j_{W_i}$) then
                    Append($\mathcal{C}, \left(W^j_i, E^j_{W_i}\right)$)
                else
                    return $\varnothing$
                end if
            end for
        else
            return $\varnothing$
        end if
    end for
    return $\mathcal{C}$
end function
```

References

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, second edition, 2006.

J. P. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *Proceedings of IFAC Control Ind. Syst.*, pages 841–846, Belfort, France, 1997.

A. L. Dulmage and N. S. Mendelsohn. Coverings of bi-partite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

P. Fogh Odgaard. Wind turbine benchmark model, 2011. http://www.kk-electronic.com/Default.aspx?ID=9385.

P. Fogh Odgaard, J. Stoustrup, and M. Kinnaert. Fault tolerant control of wind turbines – a benchmark model. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 155–160, Barcelona, Spain, 2009.

M. R. Garey and D. S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, 2000.

M. Krysander and E. Frisk. Sensor placement for fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(6):1398–1410, 2008.

M. Krysander, J. Åslund, and M. Nyberg. An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(1):197–206, 2008.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

M. Nyberg. Automatic design of diagnosis systems with application to an automotive engine. *Control Engineering Practice*, 87(8):993–1005, 1999.

S. Ploix, M. Desinde, and S. Touaf. Automatic design of detection tests in complex dynamic systems. In *Proceedings of 16th IFAC World Congress*, Prague, Czech Republic, 2005.

B. Pulido and C. Alonso-González. Possible conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics, Special Issue on Diagnosis of Complex Systems*, 34(5):2192–2206, 2004.

W. J. Rugh. *Linear System Theory*, chapter 13. Prentice Hall Information and System Sciences, 1996.

M. Staroswiecki. *Fault Diagnosis and Fault Tolerant Control*, chapter Structural Analysis for Fault Detection and Isolation and for Fault Tolerant Control. Encyclopedia of Life Support Systems, Eolss Publishers, Oxford, UK, 2002.

M. Staroswiecki and P. Declerck. Analytical redundancy in non-linear interconnected systems by means of structural analysis. In *Proceedings of IFAC AIPAC'89*, pages 51–55, Nancy, France, 1989.

C. Svärd and M. Nyberg. Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(6):1310–1328, 2010.

C. Svärd, M. Nyberg, and E. Frisk. A greedy approach for selection of residual generators. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis (DX-11)*, Murnau, Germany, 2011.

L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component-supported analytical redundancy. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 36(6):1146–1160, 2006.

Linköping studies in science and technology, Dissertations
Division of Vehicular Systems
Department of Electrical Engineering
Linköping University

No 1 Magnus Pettersson, *Driveline Modeling and Control*, 1997.

No 2 Lars Eriksson, *Spark Advance Modeling and Control*, 1999.

No 3 Mattias Nyberg, *Model Based Fault Diagnosis: Methods, Theory, and Automotive Engine Applications*, 1999.

No 4 Erik Frisk, *Residual Generation for Fault Diagnosis*, 2001.

No 5 Per Andersson, *Air Charge Estimation in Turbocharged Spark Ignition Engines*, 2005.

No 6 Mattias Krysander, *Design and Analysis of Diagnosis Systems Using Structural Methods*, 2006.

No 7 Jonas Biteus, *Fault Isolation in Distributed Embedded Systems*, 2007.

No 8 Ylva Nilsson, *Modelling for Fuel Optimal Control of a Variable Compression Engine*, 2007.

No 9 Markus Klein, *Single-Zone Cylinder Pressure Modeling and Estimation for Heat Release Analysis of SI Engines*, 2007.

No 10 Anders Fröberg, *Efficient Simulation and Optimal Control for Vehicle Propulsion*, 2008.

No 11 Per Öberg, *A DAE Formulation for Multi-Zone Thermodynamic Models and its Application to CVCP Engines*, 2009.

No 12 Johan Wahlström, *Control of EGR and VGT for Emission Control and Pumping Work Minimization in Diesel Engines*, 2009.

No 13 Anna Pernestål, *Probabilistic Fault Diagnosis with Automotive Applications*, 2009.

No 14 Erik Hellström, *Look-ahead Control of Heavy Vehicles*, 2010.

No 15 Erik Höckerdal, *Model Error Compensation in ODE and DAE Estimators with Automotive Engine Applications*, 2011.

No 16 Carl Svärd, *Methods for Automated Design of Fault Detection and Isolation Systems with Automotive Applications*, 2012.