Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Air Leakage Diagnosis in Heavy Duty Truck Engines with EGR and VGT

Examensarbete utfört i Vehicular Systems
vid Tekniska högskolan i Linköping
av

Josef Dagson & Samuel Nissilä Källström

LITH-ISY-EX--09/4210--SE

Linköping 2009

Linköpings universitet
TEKNISKA HÖGSKOLAN

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

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Air Leakage Diagnosis in Heavy Duty Truck Engines with EGR and VGT

Examensarbete utfört i Vehicular Systems
vid Tekniska högskolan i Linköping
av

Josef Dagson & Samuel Nissilä Källström

LITH-ISY-EX--09/4210--SE

Handledare: **Carl Svärd**
ISY, Linköping University/Scania CV AB
Jens Molin
Scania CV AB

Examinator: **Erik Frisk**
ISY, Linköping University

Linköping, 26 March, 2009

	Division, Department Avdelning, Institution Division of Automatic Control Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden	**Date** Datum 2009-03-26

Language Språk ☐ Swedish/Svenska ☒ English/Engelska ☐ ____________	**Report category** Rapporttyp ☐ Licentiatavhandling ☒ Examensarbete ☐ C-uppsats ☐ D-uppsats ☐ Övrig rapport ☐ ____________	**ISBN** — **ISRN** LITH-ISY-EX--09/4210--SE **Title of series, numbering** Serietitel och serienummer **ISSN** —

URL for electronic version

`http://www.control.isy.liu.se`

`http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-ZZZZ`

Title Titel	Diagnos av Luftläckage på Lastbilsmotorer med EGR och VGT Air Leakage Diagnosis in Heavy Duty Truck Engines with EGR and VGT
Author Författare	Josef Dagson & Samuel Nissilä Källström

Abstract
Sammanfattning

Scania CV AB is a leading company within development and production of buses, trucks as well as industrial and marine engines.

New environmental and safety legislations continuously demand higher quality from the products. An upcoming European legislation, Euro 6, implies that gas leakages from truck engines should be detected while driving. If the source of the leakage is not only detected, but also isolated, that is separated from other faults, the adjustments in the workshop goes faster since there is no need for leakage localisation. A faster reparation increases the up-time, i.e. the amount of time that the truck can be used.

This master thesis work uses current methods developed at Scania for residual generation to perform model-based leakage diagnosis. In this work, measurements are gathered for different sensor faults and two leakages. The measurements are used to evaluate the actual performance of the resulting diagnosis system.

The result, based on the residuals generated by the method, shows that leakages on the boost-side and the exhaust-side can be detected, and isolated from faults in the pressure sensors on the boost-side and the exhaust-side. The isolation of these four faults is considered the hardest to achieve among sensor faults and leakages why the full isolation performance is promising. Further measurements are needed to determine the full isolation performance of the diagnosis system.

The resulting system is reasoned to be suitable for execution in real time on-board the truck.

Keywords
Nyckelord diagnosis, model based, structural methods, residual generation, gas, air leakage

Abstract

Scania CV AB is a leading company within development and production of buses, trucks as well as industrial and marine engines.

New environmental and safety legislations continuously demand higher quality from the products. An upcoming European legislation, Euro 6, implies that gas leakages from truck engines should be detected while driving. If the source of the leakage is not only detected, but also isolated, that is separated from other faults, the adjustments in the workshop goes faster since there is no need for leakage localisation. A faster reparation increases the up-time, i.e. the amount of time that the truck can be used.

This master thesis work uses current methods developed at Scania for residual generation to perform model-based leakage diagnosis. In this work, measurements are gathered for different sensor faults and two leakages. The measurements are used to evaluate the actual performance of the resulting diagnosis system.

The result, based on the residuals generated by the method, shows that leakages on the boost-side and the exhaust-side can be detected, and isolated from faults in the pressure sensors on the boost-side and the exhaust-side. The isolation of these four faults is considered the hardest to achieve among sensor faults and leakages why the full isolation performance is promising. Further measurements are needed to determine the full isolation performance of the diagnosis system.

The resulting system is reasoned to be suitable for execution in real time on-board the truck.

Sammanfattning

Scania CV AB är en ledande koncern inom utveckling och produktion av bussar, lastbilar samt industri- och marinmotorer.

Nya lagkrav för miljö och säkerhet ställer ständigt högre krav på de tillverkade produkterna. Ett nära förestående lagkrav för lastbilar, Euro 6, innebär att gasläckage från motorn ska detekteras under körning. Om läckaget förutom att detekteras också kan isoleras, det vill säga särskiljas från andra fel, går reparationen i verkstaden snabbare då man slipper lokalisera läckaget. En snabbare reparation ökar up-time, det vill säga tiden som lastbilen kan användas på åkeriet.

I detta exjobb används befintliga metoder för residualgenerering framtagna på Scania för att åstadkomma modelbaserad läckagediagnos. Arbetet tar även fram mätdata för olika givarfel samt för två läckage i motorn. Denna mätdata används för att utvärdera det erhållna diagnossystemets faktiska prestanda.

Resultatet, som bygger på residualerna som metoden genererat, visar att läckage går att detektera, och att läckagen går att isolera från fel på tryckgivarsensorer på laddluftssidan och avgassidan. Denna isolering anses vara den svåraste att uppnå av alla sensorfel samt läckage varvid övrig isoleringsprestande verkar lovande. Däremot behövs mer mätdata för att säkert kunna fastställa övrig isoleringsprestanda. Diagnosmetoden lämpar sig troligen för exekvering i realtid ombord på lastbilen.

Acknowledgments

We would like to thank our examiner Erik Frisk, our supervisors Jens Molin and Carl Svärd who taught us time optimism and how to move walls, our fellow master thesis writers for many enjoyable lunches, the department of NESD for plenty of "fika"s filled with interesting discussions, the mechanics for help with design and implementation of the new hardware, all the other people at Scania that has endured our endless, yet terminated, questioning on all thinkable subjects.

Contents

Chapter 1

Introduction

This thesis was performed at Scania CV AB in Södertälje in cooperation with the department of Electrical Engineering ISY, division of Vehicular Systems, at Linköping University. The work was carried out at NESD, a department responsible for On-Board Diagnosis (OBD) at Scania, during fall 2008 to early spring 2009. Scania develops and manufactures trucks, buses and engines for marine and industrial use.

1.1 Background

There are many components in a truck that provide the engine with air, and additional ones used to purify the exhausts. These components are connected through pipes. A hole or a loose joint, both resulting in a leakage, can occur somewhere in the air or exhaust path, here after called the gas path. This might lead to reduced engine performance and/or increased exhaust levels. Legislative demands has been introduced during the last years requiring continuous supervision of sensors and actuators in order to gradually lower the emission levels. The next European standard, Euro 6, is expected to contain such limits that even the presence of leakages will have to be monitored. If the leakage could not only be detected, but also isolated, the adjustments in the workshop would go faster since there is no need for leakage localisation. A faster reparation would increase the up-time, i.e. the amount of time that the truck can be used.

With the wish to perform better diagnosis, and with legal demands from Euro 6, expected to contain restrictions to the gas leakage, this work became a current issue.

1.2 System Overview

The truck engine treated in this thesis is a diesel engine with both exhaust gas recirculation, EGR, and variable geometry turbine, VGT. The ambient air is taken in, filtered and then enters a compressor that increases the pressure before the cylinders, referred to as boost pressure, P_{boost}. Since the compressor not only increase the pressure, but also the temperature, there is an intercooler, I.C. , after the compressor to lower the temperature and thus the volume. This increases the amount of air that enters the cylinders, which increases engine performance.

Before the air enters the cylinders, it is mixed in the intake manifold with exhaust gases that are recycled through the EGR. The reason for adding exhaust gases to the air, is to lower the combustion temperature and by doing so lower the emissions. The air and exhaust mixture is blended with fuel, the amount is regulated by U_δ, and combusted in the cylinders. The exhaust gas leaves the cylinders and continues to the exhaust manifold where one part recycles to the intake manifold. The amount of recirculated gas is regulated by the EGR-valve, controled by U_{EGR}. The rest of the gas continues to the turbine that runs the compressor. The variable geometry turbine, VGT, controled by U_{VGT}, makes it possible to vary the amount of air through the compressor independently, to some limits, of the pressure difference over the turbine. The detailed usage of the valves in engine control can be found in [8]. An exhaust after treatment system is not considered in this thesis.

There are several sensors in the system, measuring temperatures, T, pressures, P, rotational speeds, N and ω, and air mass flow, W. For an overview of the gas flow and the sensors and actuators in the system, see Figure 1.1.

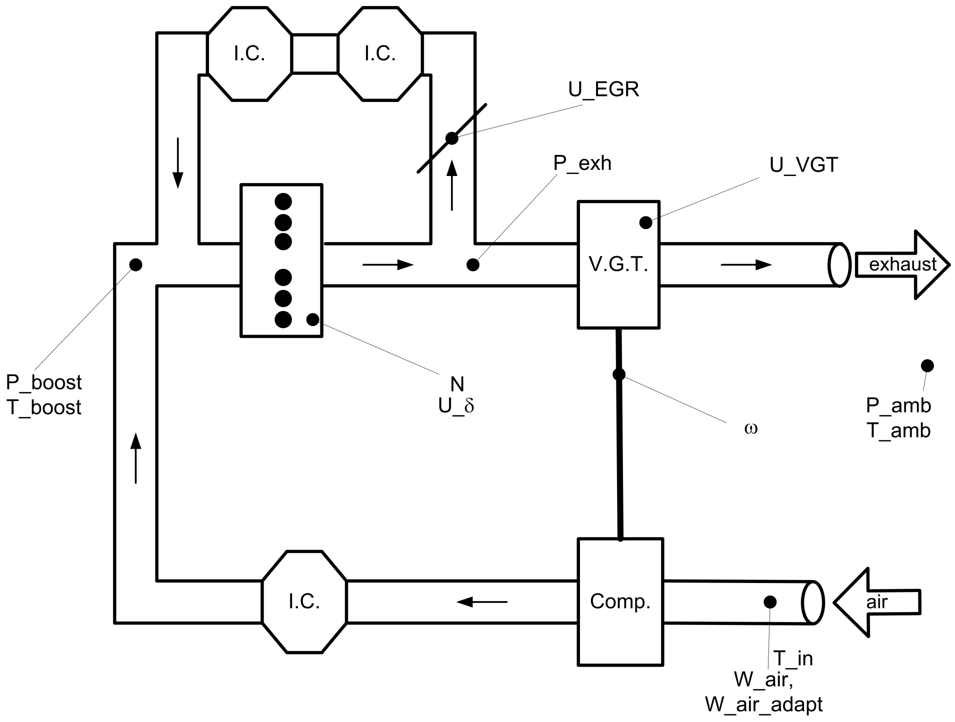


Figure 1.1. Sketch of the gas path in a diesel engine with EGR and VGT but without a catalytic converter or a particle filter. Sensors and actuators present in the system are marked with dots.

Though the figure might give the impression of a rather simple and straight forward system the recirculation in the EGR and the connection from the turbine to the compressor provides two feedback loops resulting in a more complex system then at first glance. The system also contains non-linear dynamics.

1.3 Existing Work

Some previous works treating leakages, such as [21], [27] and [1], already exist. But they deal with cars, not trucks, and the first two with gasoline SI-engines and not diesel engines. The third one uses statistical methods which requires knowledge about how often faults occur in different components and the behaviour of each considered fault that might occur in the respective components.

1.4 Purpose

The purpose of this thesis is to develop methodology for leakage diagnosis on a diesel engine with EGR and VGT in order to meet future Euro 6 legislative demands.

More concretely, this will be done by creating a diagnosis system with the developed method, and answering some of the following questions:

- Which methods can be used to perform leakage diagnosis on the engine?
- Is a model-based approach preferable compared to a non-model-based approach?
- In the case of a model based approach, is it necessary to model a leakage in detail in order to detect and isolate it?
- How well can leakages be detected and isolated from sensor faults?
- Would the result of the created diagnosis system be improved by the introduction of extra sensors, and in that case where?
- Is the created diagnosis system suitable for on-board usage?

1.5 Limitations

Due to time constraints, some limitations will be considered.

Uniformed Trucks Dispersion among analogous components in different trucks will not be specifically taken into account.

No System Failure Investigation Which fault types and sizes that leads to system failure, i.e. degradation of engine performance or exhaust levels violating legislative demands, will not be investigated. This is due to both time constraints and uncertainties considering Euro 6 contents. Instead the thesis will focus on finding as small faults, particularly leakages, as possible.

Single Faults Only single faults will be treated. The event that a sensor or actuator fault occurs at the same time as a leakage is not very likely why this is a reasonable restriction. Further, from a legislative point of view, the only double fault that risk to cause a problem is a fault leading to degraded combustion in combination with a malfunctioning particle filter, according to expert engineers at Scania. This would increase the rate of emissions. Since the particle filter is not considered in this work the problem is avoided.

Existing Engine Models Existing models will be used and, when needed, modified. Since the existing engine models have been created and refined for a much longer time than what is available for this thesis, focus will to be on leakage detection, not engine modelling.

Two Leakage Areas Two potential leakage areas will be treated. The areas are representative and covers the places where leakages are most likely to occur, and most desirable to detect.

Diagnosis System Dependabilities The diagnosis system will be considered as a separate system without treating it together with the control system, [7], [20] and [27], which possibly could prove advantageous. The influence of the control system is still taken into account since the measurements in the thesis are from a controlled system.

1.6 Method and References

The working process for the thesis is to seek information about the system, different diagnosis methods and previous related works. The information is primarily looked for in published works such as books, official reports, patent registrations etc.

A general method, applicable to the system in question, is chosen to generate the diagnosis system. Some new functionality is implemented and some prior implementations of which some are improved, is used in this task. The resulting diagnosis system is tested and evaluated with real measurements gathered from a truck.

1.7 Contributions

The main contributions of this work are:

- One approach to detect leakages in the gas path of a truck, see Figure 5.1 in Chapter 5.
- Creation of hardware and equipment to acquire measurement data from leakages, Section 4.1.
- Combining and improving implementations of existing methods for residual generation, see Chapter 5, Section 6.3.1 and Section 6.3.2.
- Evaluation of the created diagnostic tests with real measurements, Section 6.5.

1.8 Structure

Chapter 2 - Background Theory This chapter gives a basal knowledge about diagnosis in general and consistency based diagnosis in particular. The theories behind the methods used in this work are presented. The chapter gives a reader without a solid diagnosis background the possibility to understand the rest of this work.

Chapter 3 - Engine Model The engine model as well as fault models are introduced and validated in this chapter. Here it is also explained which behavioural modes that are considered in this thesis.

Chapter 4 - Experimental Setup and Data Acquisition The experimental setup needed to acquire fault free measurements, as well as measurements when different faults are present, is displayed in this chapter. The way that the setup is used in is also shown.

Chapter 5 - Test Quantity Generation Methods using theories in Chapter 2 are presented in this chapter. Some implementations of the methods are presented and exemplified. Finally it is explained how the different methods are combined.

Chapter 6 - Results The findings relevant for this thesis as well as the context in which they belong are presented here.

Chapter 7 - Analysis and Discussion This chapter is dedicated to a discussions about some of the results and to how they can be used.

Chapter 8 - Concluding Words A summary and some suggestions about possible future work are presented here.

Appendix Some bulky material that does not fit in the floating text of the other chapters are here presented for the interested reader.

Chapter 2

Background Theory

This chapter starts by giving the reader some prerequisites followed by a short introduction to different ways of performing diagnosis. After that a deeper presentation of the elements that are used by the approach in this thesis is carried out.

The last parts of this chapter are dedicated to theory that will come in handy when dealing with large complex models that can not be exhaustively treated by hand and without a systematic method.

2.1 Preliminaries

A *fault*, according to [22], is an unpermitted deviation of at least one characteristic property or variable of the system from acceptable/usual/standard/nominal behaviour. A fault can be further divided into an actuator, process or sensor fault, depending on what part of the system it affects.

A *residual* is a signal that, in the absence of faults, deviate from zero only due to modeling uncertainties. The nominal value is zero or close to zero under actual working conditions. In order for the residual to be useful within diagnosis applications, it is desirable that the residual diverges from zero for certain faults, the *monitored faults*, while not reacting to other faults, the *unmonitored faults*.

Fault detection is the act of deciding whether a fault has occurred or not, sometimes including the determination of when the eventual fault occurred. To determine where the fault has occurred, i.e. to conclude in what component(s) of the system a fault is present is called *fault isolation*. The determination of the size and time-variant behaviour of the fault is called *fault identification* [22].

Fault accommodation, according to [22], is the procedure of reconfiguring the system and/or the controller so that the desired system operation can be maintained in spite of a present fault.

Sensor faults can be detected by a residual subtracting an estimation of the sensor value with the actual sensor value. In order to achieve fault accommodation in the case of residuals constructed this way, the sensor reading from a faulty sensor can be replaced with its estimate and thus it is sufficient to detect and isolate the faulty sensor in this case. In case of an actuator fault which do not cause a complete loss of command, a solution would be to compute a new control signal to the actuator in a way that the fault is compensated. Hence an estimate of the fault characteristics is also needed in order to reach fault accommodation in this case i.e fault identification is also needed.

The procedure of fault detection, isolation and identifications is in [12] defined as *fault diagnosis*. Different definitions exists and in this thesis the term is considered to include fault detection, isolation and if needed, identification. Fault diagnosis is sometimes referred to as diagnosis but the term is not to be confused with *a diagnosis*, also called *a diagnosis statement*, which is the outcome after having diagnosed a system.

If diagnosis is performed without affecting the system it is called *passive diagnosis*. If the system is actively exited in order to reveal possible faults (that might not show clearly otherwise) it is called *active diagnosis*.

2.2 Methods for Diagnosis

In this section some of the approaches used today for automated diagnosis are presented. These methods have become available with the increased computational capacity following the birth of computers [22].

2.2.1 Limit Checking Approach

In this method operating ranges are assigned to for example the sensors of a system. As long as the sensor readings are within their predefined ranges the behaviour of the system is considered faultless. This approach has traditionally been the choice for automated diagnosis [22]. In this thesis limit checking of sensors and actuators will be regarded as something implemented in the trucks by default for electrical diagnosis, i.e. by checking voltages and/or currents, and thus as something of less interest.

2.2.2 Hardware Redundancy Approach

Another approach to automated diagnosis is duplication (or triplication or more) of hardware. This is called *hardware redundancy*. A normal piece of hardware to duplicate is the sensor, e.g. put two sensors at the same place in a system. The faultless behaviour in that case is that the sensors readings do not differ much. If a significant difference exists, the conclusion would be that one of the sensors are broken. By introducing a third sensor it would also be possible isolate the broken sensor. Consider for example the sensors y_1, y_2 and y_3. The following three residuals can be formed $r_1 = y_1 - y_2$, $r_2 = y_1 - y_3$ and $r_3 = y_2 - y_3$. Two of the residuals would give a significant divergence from zero in the case of one sensor being faulty. The third residual would ideally be zero but due to measurement disturbances it is normally said to be close to zero. The faulty sensor would be the one that is included in the two residuals that are further from zero than the third one. Hardware redundancy is used in safety critical applications but is often avoided in other cases and also in this thesis due to cost, space or weight considerations [22].

2.2.3 Relational and Pattern Recognition Approach

These approaches need information about the faulty behaviour of the system [6]. Realistic information about how often components get faulty, the fault nature and size needs to be available or collected. Considering that trucks are designed to work without faults even in extreme conditions for a very long time, it would probably require constant logging of data during many years in the trucks out in the field to collect the information needed. This approach is thus considered out of scope for this thesis. The interested reader is referred to [23] and [25] for information about the relational approach and the pattern recognition approach respectively.

2.2.4 Consistency-Based Approach

There are many diagnosis methods in this category. Another name for diagnosis using this approach would be *model-based diagnosis* since all consistency-based methods use a model of the system that is diagnosed. The model, simply put a description of a system, is used to provide redundancy instead of additional hardware. Sometimes the notion *analytical model* is used, when dealing with a mathematical model. When diagnosing a system, the model is used together with system observations, e.g. measurements, to calculate a so called *diagnosis statement* S. More precisely the model is used to form so called diagnostic tests $\lambda_k(z)$. The information from all the tests is then evaluated by an isolation logic in order to form the diagnosis statement S. The principle is illustrated in Figure 2.1 [22].

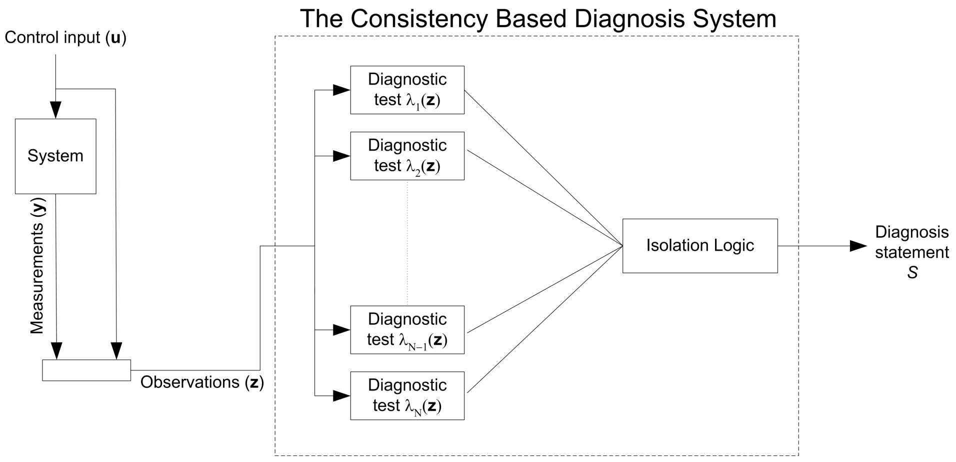


Figure 2.1. Structure of a consistency based diagnosis system.

2.3 Consistency-Based Diagnosis Systems

In the first part of this section the fundamentals of a consistency based diagnosis system are described. As can be seen in Figure 2.1, the diagnosis system is constructed of diagnostic tests and an isolation logic. These elements are demonstrated in the two last parts of this section. For a more exhaustive exposition, please refer to [4] or [22].

Most of the examples in this section are based on continuous variable models. These models are normally used by a community within consistency-based diagnosis called Fault Detection and Isolation (FDI) . This community has evolved in the field of automatic control from the seventies and uses control and statistical theory techniques [6]. Another community that has emerged more recently within consistency-based diagnosis is the Diagnostic (DX) community, which is based on the field of computer science and artificial intelligence. The terms and notations in this section is consistent with the presentation in [4] and comes from both the FDI and the DX communities. The interested reader is referred to [6] for a comparative analysis of FDI and DX.

Recently a lot of work has been put into developing a common terminology for fault diagnosis, and to identify the similarities and the complementary features in the problem definitions and problem solutions developed by the two communities. The goal is to contribute toward a unifying framework, which will enable researchers and practitioners to take advantage of the synergy in the complementary techniques employed in engineering and computer science in order to be able to solve even more complex diagnostic tasks [3].

2.3.1 General Principles

Some concepts needed for the understanding of consistency-based diagnosis are now explained. In order to describe faults that might occur in a system, the term component is used. The idea of a component is that it is something that can brake. Normally, one can consider a component as either faulty or non-faulty. In general, however, there are many ways in which a component can break or become faulty. For this reason, it is convenient to talk about the *behavioural mode* of the component. Normally the behavioural modes of a component are *no fault* and one or more *fault modes*. No fault is often abbreviated as NF and some examples of fault modes that can be used for different fault behaviours of, for instance, a sensor, are bias fault, BF, and gain fault, GF. A component is only in one behavioural mode at each time instant. The Example 2.1 is a modification of an example in [22] and gives an idea of how behavioural modes are used in the context of consistency-based diagnosis.

Example 2.1: Bicycle Diagnosis

Consider for instance a bicycle. Let the observations be the forward speed and the pedalling speed, and let the chain be the diagnosed component. If the pedalling speed is non zero and the bicycle is moving, one can assign the behavioural modes NF or BROKEN to the chain (considering that we could be in a slope). However, if the pedals are moving while the bike is not,the chain is assigned the behavioural mode BROKEN since it is the only behavioural mode assignment to the chain that is consistent with the observations and with our knowledge of how the bike works (the model).

It is often convenient to refer to all behavioural modes of all components in the system. Hence the concept of *system behavioural modes* is introduced. Consider a system consisting of a pipe and a pressure sensor. The component behavioural modes studied are NF and LF, leakage fault, for the pipe and NF and PSF, pressure sensor fault, for the pressure sensor. A consistency based diagnosis system might then be used to assign any of the following system behavioural modes: NF, LF, PSF or LF & PSF given system observations.

In order to be able to assign fault modes given observations, i.e. to isolate the faults, the faults must be modeled [4]. A *fault model* is the representation of where in the system a fault can occur, e.g. which component that can become faulty, and how the fault behaves. Complex fault models leads to a more complex model in total but augments the possibility to isolate between faults [22]. If no prior knowledge is available on how a fault behaves, a general fault model can be applied. The disadvantage with general fault representation is that in a system with many faults represented by general fault models, it can be difficult or even impossible to isolate between the faults [22]. Fault modelling is thus a trade-off between model complexity and diagnosis performance. The Example 2.2 shows some different kinds of fault models such as general fault, gain fault and bias fault. See Example 2.3 for how fault models and behavioural modes are used in consistency-based diagnosis.

Example 2.2: Fault Models

Consider the measurement y_P of a pressure P. Some different fault models are:

$$y_P = \begin{cases} P + f & \text{general fault model} \\ P \times f_g, \ \dot{f}_g = 0 & \text{gain fault} \\ P + f_b, \ \dot{f}_b = 0 & \text{bias fault} \end{cases}$$

Example 2.3: Fault Representation

Consider the sensors y_W and y_P measuring the mass flow W and the pressure P in a tank, and the model $\dot{P} = K \times W$ where K is a constant. Let the diagnosed components be the sensors. The model can be used to estimate for example P using y_W and vice verse. Assume that prior knowledge shows that a constant bias fault, f_{i_b} with $\dot{f}_{i_b} = 0$ where $i \in \{P, W\}$, is added to the sensors when faulty. The estimated and the measured values can then be compared for instance by forming the residual r.

$$\begin{aligned} r &= (y_P + f_{P_b}) - \int (K \times (y_W + f_{W_b})) \\ &= y_P - \int (K \times y_W) + f_{P_b} - \int (K \times f_{W_b}) \\ &= f_{P_b} - K \int f_{W_b} \end{aligned}$$

The restrictive fault modelling is in this case a prerequisite for isolating faults in both sensors from each other since both sensor faults will affect the residual. Let FWS and FPS be the bias fault mode for the mass flow sensor and the pressure sensor respectively. If r deviates from zero by a constant value, the only assignment of a system behavioural mode consistent with the model and the observation would be FPS. If r increases linearly, the only consistent system behavioural mode assignment would be FWS. Note that no model uncertainties or measurement noise is assumed in this example.

Having acquired the basic knowledge of the concepts of consistency-based diagnosis, it is time for a more precise definition. Let a model used for diagnosis be called a *diagnostic model.* Consistency-based diagnosis systems might use different kinds of models and have different assumptions concerning what measurement information that is available, but they all work according to the following definition [4].

Definition 2.1 (Consistency-Based Diagnosis System) *Given a set of observations and a diagnostic model, the task of a consistency based diagnosis system is to generate a set S of diagnoses, the diagnosis statement, which contains the system behavioural modes that can explain the observations (i.e. that are consistent with the observations and the model).*

Fault detection and isolation performance in a consistency-based diagnosis system is, as indicated earlier, directly dependant on the model used. The terms *detectability* and *isolability* are model properties and will now be formally defined with the help of the notion of an *observation set.*

Definition 2.2 (Observation Set) *Given a diagnostic model M together with the assignments of the system behavioural mode to b, the observation set O_b is the set of all possible values of the observation vector $\mathbf{z}$ such that the model M is consistent [22].*

Definition 2.3 (Isolability) *A system behavioural mode b_1 is isolable from the system behavioural mode b_2, in model M, if $O_{b_1} \not\subseteq O_{b_2}$ [22].*

Definition 2.4 (Detectability) *A system behavioural mode b is detectable in model M if b is isolable from the system behavioural mode* **NF** *in the model M [22].*

The Definitions 2.2, 2.3 and 2.4 are illustrated by Figure 2.2 in the six following situations.

a) The observation set O_{NF} is shown.

b) Fault mode O_b is not detectable since all the observations of $\mathbf{z}$ that are consistent with O_b are also consistent with O_{NF}.

c) Fault mode O_b is detectable since at least some observations of $\mathbf{z}$ can only be explained by O_b.

d) The fault mode O_b is detectable.

e) The fault modes O_{b_1} and O_{b_2} are detectable. O_{b_2} is isolable from O_{b_1} but not vice verse.

f) Both fault modes are detectable and isolable from each other.

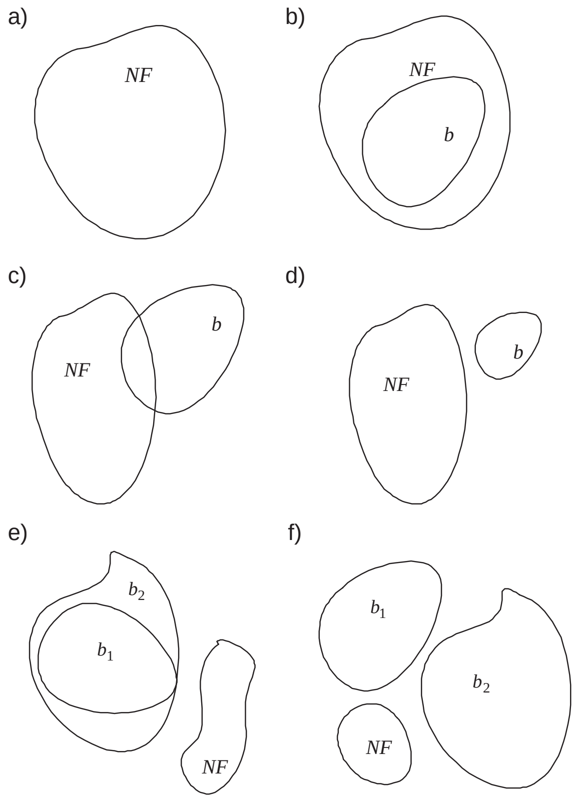


Figure 2.2. Isolability and detectability illustrated by observation sets.

2.3.2 Diagnostic Tests

This section treats diagnostic tests, a major component in a consistency-based diagnosis system, see Figure 2.1.

The structure of a diagnostic test $\lambda_k(z)$ is shown in Figure 2.3. The output from each diagnostic test is binary. Other approaches than thresholding exists in order to generate a binary output from an often continuous test quantity T_k, but thresholding is the most common way [22]. The test quantities T_k can be constructed using different models and

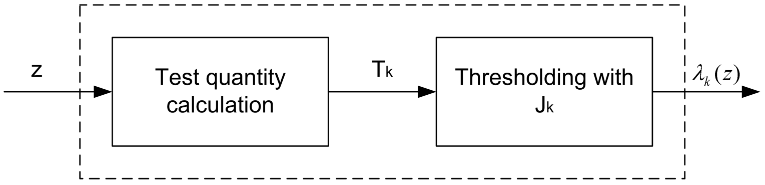


Figure 2.3. A diagnostic test.

different methods. No matter how they are constructed the idea is to make different tests quantities that reacts to faults belonging to different system behavioural modes. This can be illustrated by an *influence structure*. See Example 2.4.

Influence Structure

The influence structure is a representation of how test quantities *ideally* are affected by faults in different system behavioural modes [22]. The meaning of *ideally* in this case is that no unmodelled disturbances exists as well as no measurement noise. The influence structure is a matrix where each row corresponds to a test quantity T_i and each column to a system behavioural mode, NF or some F_j. Sometimes when the NF-column only contains zeros, it is not included in the matrix. The position A_{ij} in the matrix is defined as

$$A_{ij} = \begin{cases} 1 & \text{if } T_i \text{ always reacts to all faults in fault mode } F_j \\ x & \text{if } T_i \text{ reacts under some operating points to some faults in fault mode } F_j \\ 0 & \text{if } T_i \text{ never reacts to any faults in fault mode } F_j \end{cases}$$

Example 2.4: Influence Structure

Consider a situation with two test quantities, T_1 and T_2. The test quantity $T_1 = y_1 - u$ is sensitive to faults in sensor one and actuator u and reacts to some faults in F_1 and some faults in F_u. The test quantity $T_2 = y_2 - y_1$ is sensitive to faults in sensor one F_1 and sensor two F_2 and reacts to all faults in F_2 but only some faults in F_1. The influence structure for T_1 and T_2 is

	F_1	F_2	F_u
T_1	x	0	x
T_2	x	1	0

Sensitivity Matrix

In order to be able to rate test quantities subject to real measurements before having created thresholds, the notion *sensitivity matrix* is introduced. It is defined in the same way as the influence structure but instead of assigning the 0s, xs and the 1s in the matrix analytically, the assignment is done according to measurements of the system subject to as many different faults as possible. Model uncertainties and measurement noise is thus taken into consideration. See Example 2.5.

Example 2.5: Sensitivity Matrix

Consider the same situation as in Example 2.4. Now let's say that some faults in fault mode F_2 due to model errors actually effects the test quantity T_1. The zero will be turned into an x. The sensitivity matrix is thus

	F_1	F_2	F_u
T_1	x	x	x
T_2	x	1	0

Test Quantity Creation

In order to construct test quantities it is necessary and sufficient that the model over the system in question contains *analytical redundancy* [22], which can be defined as follows:

Definition 2.5 (Analytical Redundancy) *There exists analytical redundancy if there are two or more different ways to determine a variable x by only using the observations $\mathbf{z}$ and the model, i.e. $x = f_1(\mathbf{z})$ and $x = f_2(\mathbf{z})$ where $f_1(\mathbf{z}) \not\equiv f_2(\mathbf{z})$.*

In the case of a linear model there are many methods that, given a specification of the wanted diagnosis performance, can be used to create a set of test quantities which will match the performance in question, providing that the model is sufficiently redundant. An example of how analytical redundancy is used to create a detection system can be found in [5], where the starting point is a state space model, and an approach called *Parity Space* [22] is used.

When dealing with a system that contains non-linear behaviour, a non linear model is needed to describe the behaviour. The model can be linearized around certain operating points [27] of the system whereupon linear methods can be used for each chosen operating point. However, linearisation always induces errors to the test quantities which has to be taken into account during the design process. These errors can be reduced by introducing more operating points but for a highly dynamical system it might prove that an unmanageable number of operating points has to be treated in order to achieve wanted diagnostic performance.

In the general non-linear case where linearisation is not used there is no specific method that given some specifications generates the best set of test quantities that match the criteria. There are however methods for some special kinds of non-linear models, see for example [9] where polynomial differential-algebraic systems is treated or [16] in the case of Lipschitz non-linearities.

If the model contains analytical redundancy it is possible to form Analytical Redundancy Relations (ARR) which are consistent in the absence of faults and can thus be used to form residuals. The residuals can be used directly as test quantities or after having been filtered, typically with a low pass filter.

2.3.3 Isolation Logic

With a multitude of diagnostic tests, see Figure 2.1, the information from each diagnostic test can be combined to form a diagnosis statement S. This is the task of the isolation logic. There are different approaches to do this: column matching approach, structured hypothesis tests approach and minimal hitting set approach. For more details on the methods see [22]. The column matching approach is used in this thesis.

One way of presenting the information from the diagnostic tests is with a *decision structure*. The decision structure can be used by, for example, the column matching approach to perform isolation. The resulting performance of the isolation logic, i.e. which fault modes that can be isolated, is presented in an isolation structure.

Decision Structure

An influence structure is practical and easy to use since it is based on ideal conditions. In reality, ideal conditions seldom exists and thus an influence structure can not be relied upon. Instead a decision structure [22] is used. The decision structure is based on the same formulation as the influence structure but uses thresholded test quantities, i.e. the outcome from the diagnostic tests $\lambda_k(z)$. The definition of the position A_{ij} in the decision structure is defined as

$$A_{ij} = \begin{cases} 1 & \text{if } \lambda_i \text{ always reacts to all faults in fault mode } F_j \\ x & \text{if } \lambda_i \text{ reacts under some operating points to some faults in fault mode } F_j \\ 0 & \text{if } \lambda_i \text{ never reacts to any faults in fault mode } F_j \end{cases}$$

For an example of a decision structure, see Example 2.6.

Example 2.6: Decision Structure

Consider the same situation as in Example 2.4. Let's say that the threshold J_2 for the diagnostic test T_2 has to be chosen high in order to totally avoid false detection due to model uncertainties or measurement noise. This may lead to that some faults in fault mode F_2 never makes the test quantity T_2 exceed its threshold. Thus, λ_2 will no longer react to all faults in fault mode F_2. This will result in the 1 turning into an x. The decision structure is thus

	F_1	F_2	F_u
λ_1	x	0	x
λ_2	x	x	0

Column Matching Approach

The column matching approach use the information in the decision structure to form one diagnosis statement conform with the result from all diagnostic tests. One way of constructing an isolation logic from a decision structure is to find the intersection of all the different diagnostic substatements. An example of an decision structure is

	F_1	F_2	F_u
λ_1	x	0	x
λ_2	x	1	0
λ_3	0	x	x

Assume that λ_1 and λ_3 has reacted, but that λ_2 has not. The three diagnostic substatements are: $\{F_1, F_u\}$, $\{\neg F_2\}$ and $\{F_2, F_u\}$. The intersection of these three statements is F_u, and the diagnosis statement S would in this case be $S = \{F_u\}$.

Isolation Structure

In order to isolate one fault mode from an other, the modes needs to be isolable, see Definition 2.3. So to isolate fault mode F_i from fault mode F_j, $i \neq j$, some test quantities are needed that reacts to the different fault modes. More concretely, F_i can be isolated from F_j if there exists a test quantity T such that T is sensitive to F_i but not to F_j.

The information about which fault modes that can be isolated from each other is gathered in an *isolation structure*. The isolation structure is a matrix with a fault mode associated to each row and column. Each element in the matrix indicates if the fault mode on row i can be isolated from the fault mode in column j or not. A '0' indicates that the fault mode on row i can be isolated from the fault mode in column j, a '1' that it can not. See Example 2.7.

Example 2.7: Isolation Structure

In the isolation structure below it can be seen that fault mode F_1 can be isolated from fault mode F_2 but not from F_1 or F_u. In other words, F_1 can be isolated from F_2 but not from F_u. With the same reasoning, F_u can be isolated from F_1 but not from F_2. Note that F_u can be isolated from F_1, but F_1 can not be isolated from F_u.

	F_1	F_2	F_u
F_1	1	0	1
F_2	1	1	0
F_u	0	1	1

For perfect isolation a unity matrix is required, i.e. 1 on the diagonal and 0 every where else, which implies that each fault mode can only be taken for it self.

Minimal Hitting Set

The minimal hitting set is defined in Definition 2.6. The minimal hitting set method can be used in isolation purposes to find a minimal diagnosis statement among a multitude of diagnostic substatements. This application of the minimal hitting set method is presented in [22]. But, the same technique can also be used to find a minimal set of diagnostic tests with a specific isolation performance from an original, usually bigger, set of tests.

Definition 2.6 (Minimal Hitting Set) *Let C be a collection of subsets of S. A hitting set of C is a subset $S' \subseteq S$ so that S' contains at least one element from each subset in C. A minimal hitting set, MHS, is a set S' that contains as few elements as possible.*

A minimal hitting set can be found in the following way. Start with a given influence structure, and an empty set of tests, $mhs = \{\}$. Then create a set, $isol$, of all tests that isolate fault mode k from fault mode l, $l \neq k$. Form the cross product of mhs and $isol$, and eliminate all non-minimal sets. If mhs is empty, the new set is identical to $isol$. If $isol$ is empty, fault mode k can not be isolated from fault mode l. Update mhs to only contain the new minimal sets. Repeat this process until all fault modes have been treated. Then mhs contains all minimal hitting sets found.

The finding of a minimal hitting set is illustrated in Example 2.8.

Example 2.8: Minimal Hitting Set for Selection of Diagnostic Tests

Consider a number of diagnostic tests, λ_1 - λ_6, with in the influence structure:

	F_1	F_2	F_3
λ_1 :	x	x	
λ_2 :	x		x
λ_3 :		x	x
λ_4 :	x		
λ_5 :		x	x
λ_6 :			x

Start by finding all tests that isolate fault mode F_1 from F_2: $\{\{\lambda_2\}, \{\lambda_4\}\}$. Since mhs is empty, the new set is the same as $isol$. Update mhs with these sets.

Next, find tests that isolate fault mode F_1 from F_3: $\{\{\lambda_1\}, \{\lambda_4\}\}$. The cross product of mhs and $isol$ does after elimination of non-minimal sets result in the minimal sets: $\{\{\lambda_1, \lambda_2\}, \{\lambda_4\}\}$.

The set of tests that isolate F_2 from F_1 are: $\{\{\lambda_3\}, \{\lambda_5\}\}$, resulting in the minimal sets: $\{\{\lambda_1, \lambda_2, \lambda_3\}, \{\lambda_1, \lambda_2, \lambda_5\}, \{\lambda_3, \lambda_4\}, \{\lambda_4, \lambda_5\}\}$.

And so on, until all three fault modes have been treated. The resulting minimal hitting sets are: $\{\{\lambda_1, \lambda_2, \lambda_3\}, \{\lambda_1, \lambda_2, \lambda_5\}, \{\lambda_1, \lambda_4, \lambda_5, \lambda_6\}, \{\lambda_1, \lambda_3, \lambda_4, \lambda_6\}\}$. Since λ_3 and λ_5 have the same sensitivity, they are interchangeable in the final sets, which is visible.

2.4 Structural Analysis

Structural analysis is a tool that enables to work with large and complex non-linear differential and algebraic equation systems in an efficient way [19]. By looking merely at the structure, there are often ways to draw conclusions on the properties of the original system.

Some different steps and concepts within structural analysis will be illustrated in examples, building on each other, with the start in the model described by (2.1) that has the corresponding Simulink implementation in Figure 2.4.

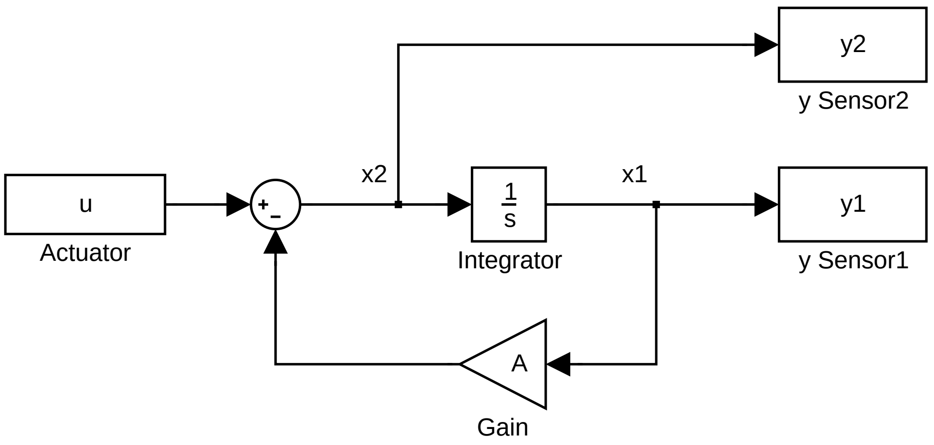


Figure 2.4. A Simulink implementation of the system described by (2.1).

$$
\begin{aligned}
e_1 &: \quad \dot{x}_1 &&= x_2 \\
e_2 &: \quad x_2 &&= -Ax_1 + u \\
e_3 &: \quad y_1 &&= x_1 \\
e_4 &: \quad y_2 &&= x_2
\end{aligned}
\tag{2.1}
$$

2.4.1 Structural Model

A structural model, SM, is created from model equations describing the system. There are different approaches to how derivatives are treated when creating the SM, see for example [18]. In this thesis, the appearance of signs, constants, computation, exponents and derivatives will be ignored in the SM, only the *presence* of the variables in an equation will be taken into account, i.e. if the variable is contained in the equation or not. So structurally, $\dot{x}$ and x are equivalent.

The structural model is a matrix with each line corresponding to an equation, e_i, and each column corresponding to a variable, x_j. The position A_{ij} in the matrix is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if } x_j \text{ is contained in equation } e_i \\ 0 & \text{otherwise} \end{cases}$$

Equation e in (2.2) is represented by the structural model presented in Table 2.1, since x_1, x_2, x_4 and u are contained in the equation but not x_3.

$$e : \dot{x}_1 = 10 - \frac{1}{2} x_2 \sqrt{x_4} + 5u \tag{2.2}$$

	x_1	x_2	x_3	x_4	u
e:	1	1	0	1	1

Table 2.1. Structural representation of e in (2.2).

Example 2.9: Transition from Model Equations to Structural Model

Consider the model described by (2.1). It can be transformed into a corresponding SM seen in Table 2.2. The variables x_1 and x_2 are contained in e_1 in (2.1), and their presence in that equation is marked with a '1'. The variables u, y_1 and y_2 are not contained in e_1 and their absence is marked by a '0'. The same reasoning is applied to the other equations.

	x_1	x_2	u	y_1	y_2
e_1 :	1	1	0	0	0
e_2 :	1	1	1	0	0
e_3 :	1	0	0	1	0
e_4 :	0	1	0	0	1

Table 2.2. Structural model for the exemplifying model (2.1).

2.4.2 Minimal Structurally Over-determined

In order to create diagnostic tests, analytical redundancy is needed, recall Section 2.3.2. For analytical redundancy to exist, a system must be structurally over-determined or consist of a least one structurally over-determined subsystem. A system of equations is structurally over-determined if it contains more equations than unknown variables, and structurally under-determined if it contains more unknown variables than equations. If a system contains as many equations as variables, it is said to be structurally just-determined. It is necessary but not always sufficient that the system is structurally over-determined in order for analytical redundancy to exist, as the following equation system demonstrates.

$$\begin{aligned} e_1 &: \quad x_1 + x_2 &= 0 \\ e_2 &: \quad 2x_1 + 2x_2 &= 0 \\ e_3 &: \quad 3x_1 + 3x_2 &= 0 \end{aligned} \tag{2.3}$$

After applying simple multiplicative operations to the equations it can be seen that the equations are equal. The equation system is thus structurally over-determined, containing three equations and two unknown variables, but analytically under-determined since the equations are identical. This problem is usually avoided though by simply not introducing different equations containing the same information.

The structural determinedness of a system can be indicated by the number of equations compared to the number of variables. A system having 4 more equations than unknowns is a '+4 system'. A structurally over-determined system is a minimal structurally over-determined, MSO , system if no proper subset is a structurally over-determined set. For exemplification see Example 2.10. MSOs can be used for residual creation by removing one equation, using it as the residual equation, and solving the just-determined system that is left. One efficient way of finding MSOs has been suggested in [18].

Example 2.10: Illustration of MSOs

The system described by the SM in Table 2.2 is overdetermined since it contains four equations and only two unknown variables, x_1 and x_2. Equation e_1, e_2 and e_3 forms one MSO, with three equations and two unknown variables. Equation e_1, e_2 and e_4 forms an other MSO with three equations and two unknown. In total there are four MSOs; $\{e_1, e_2, e_3\}$, $\{e_1, e_2, e_4\}$, $\{e_1, e_3, e_4\}$ and $\{e_2, e_3, e_4\}$.

2.5 Computation Sequence

A computation sequence, CS, is an ordered set of pairs, each consisting of a set of equations and a set of variables. It specifies in which order variables X in a just-determined system can be computed, and from which equations E they should be computed. For a formal definition see [24], in which also an algorithm for finding computation sequences is given. First structural information is used to find the order in which the variables should be computed, then the variables are tried to be computed with the available solving tools. If all variables can be computed, the ordered set of pairs is saved as a computation sequence.

2.5.1 Finding Computation Sequences

Since the order in which the variables should be computed is formed from a structural model, there is no guarantee that a variable x can in fact be computed from an equation e. There might be non-invertable equations, like $y = max(0, x)$. The equation can also be so complex that no explicit solution can be found with the used solving tool. In either of these cases, no computation sequence is created.

Example 2.11: Computation Sequence

Consider MSO_1 found in Exemple 2.10. Remove equation e_1, leaving the equations e_2 and e_3 that form a just-determined system, with two equations and two unknown variables.

	x_1	x_2	u	y_1	y_2
e_2 :	1	1	1	0	0
e_3 :	1	0	0	1	0

An order in which to compute $\{x_1, x_2\}$ is $C_1 = ((x_1, e_3), (x_2, e_2))$, i.e. compute

1. x_1 from e_3
2. x_2 from e_2

Using algebraic solving tools, the variables can be computed as

$$\begin{aligned} x_1 &= y_1 \\ x_2 &= -Ax_1 + u \end{aligned}$$

Since all variables can be computed, the ordered set of pairs is saved as the computation sequence C_1.

In Example 2.12, where equations e_1 and e_2 are considered for the generation of a computation sequence, a complication in the form of strongly connected component is present since there are alternative ways of computing the unknown variables.

Example 2.12: Strongly Connected Component

Consider the equations e_1 and e_2 in the exemplifying model (2.1). These equations form a just-determined system. Here however, the computation order can not be uniquely decided due to an algebraic loop. An algebraic loop is a set of two or more variables that has to be solved simultaneously. The block formed by equation e_1 and e_2 and the variables x_1 and x_2 is a strongly connected component, SCC, of size 2. The size is a measurement of how many variables that are represented in the SCC.

	x_1	x_2	u	y_1	y_2
e_1 :	1	1	0	0	0
e_2 :	1	1	1	0	0

Table 2.3. Structural model for e_1 and e_2 in the model (2.1).

Using existing tools, the SCCs can be solved analytically or numerically and a computation sequence might be found.

In this thesis, tools for solving algebraic (non differential) equation, tools for solving ordinary differential equations and differentiation tools are considered. How differential equations are handled is a special case in the step to compute a variable and is based on *causality*.

2.5.2 Integral and Derivative Causality

In this context, causality is a concept defining how variables can be computed from differential equations. There are two kinds of causalities, derivative causality and integral causality [4]. For a differential equation like $\dot{x}_1 = x_2$ there are two different ways to compute the variables.

With derivative causality, x_2 is the only variable that can be computed and is found as

$$x_2 = \frac{d}{dt}x_1$$

In order to use derivative causality, time derivatives of variables are needed. In general they are not known why an estimate of the derivatives needs to be calculated. Good time derivative estimations are often considered difficult to obtain due to noisy signals.

With integral causality only x_1 can be computed

$$x_1 = x_1(t_0) + \int_{t_0}^{t} (x_2(\tau))d\tau$$

In order to be able to use integral causality either the initial condition, $x(t_0)$, or global stability is needed. This is due to the fact that global stability assures that the integrated variable will always converge towards the correct value. An initial condition is needed when global stability does not exist but it might not be sufficient due to stability issues. When relying on global stability in the context of diagnosis one must assure that sufficient time is admitted for the integrated variables to converge before the diagnosis system is activated.

A mixed causality approach allows both integral and derivative causality in the same computation sequence.

The method used in this thesis handles mixed causality.

2.6 Method for Residual Generation

There is a clear benefit of having an automatic method to generate residuals from a model. For example, if minor changes are applied to the model, the residuals needs to be regenerated. There may also be an interest to find residuals for other models.The method for residual generation that is used in this thesis is described in [24] and provides the means to automatically regenerate the residuals if the model is changed. The method generates, in the following four steps, a residual from a model.

1. Find an MSO set in the model.
2. Form a just-determind set from the MSO by removing an equation.
3. Find a computation sequence in the just-determined set of equations.
4. Use the removed redundant equation as a residual equation.

For the first step, many methods exists, for example [18]. It is in theory possible to generate at least as many residuals from an MSO as the number of equations it consists of. How residuals can be found from an MSO using this method is shown in Example 2.13.

Example 2.13: Residual Generation

Take the first MSO found in Example 2.10, consisting of $\{e_1,\ e_2,\ e_3\}$:

$$\begin{aligned} e_1: \quad \dot{x}_1 &= x_2 \\ e_2: \quad x_2 &= -Ax_1 + u \\ e_3: \quad y_1 &= x_1 \end{aligned}$$

Removing one equation, for example e_1, rebounds to Example 2.11 with the computation sequence $C_1 = ((x_1, e_3), (x_2, e_2))$ corresponding to:

$$\begin{aligned} x_1 &= y_1 \\ x_2 &= -Ax_1 + u \end{aligned}$$

Now use the removed equation, $e_1 : \dot{x}_1 = x_2$, to create a residual as $r = x_2 - \dot{x}_1$ or $r = \dot{x}_1 - x_2$

In the same way equation e_2 and e_3 can be removed resulting in two other possible residuals. All three residuals and the computation sequences that can be found in the MSO are the following.

$$\begin{array}{lll} C_1 & C_2 & C_3 \\ x_1 = y_1 & x_1 = y_1 & \dot{x}_1 = x_2 \\ x_2 = -Ax_1 + u & x_2 = \dot{x}_1 & x_2 = -Ax_1 + u \\ r_1 = \dot{x}_1 - x_2 & r_2 = Ax_1 + x_2 - u & r_3 = x_1 - y_1 \end{array}$$

The three residuals are:

$$r_1 = \dot{y}_1 - u + Ay_1 \quad r_2 = Ay_1 + \dot{y}_1 - u \quad r_3 = x_1 - y_1$$

Note that r_1 and r_2 are identical if the computed variables are substituted into the residual equation. In order to use r_1 and r_2, derivative causality is needed. For r_3, integral causality is needed, since the residual results in a classical state space model [13] of a system on the form:

$$\begin{cases} \dot{x}_1 = -Ax_1 + u \\ r_3 = x_1 - y_1 \end{cases}$$

2.7 Signal Processing

There are different ways of processing signals in order to enhance or repress specific behaviours. A very common process is low pass filtering [13] that reduces the impact of rapid changes in the signal. In this thesis, two additional signal processing methods will be used in particular, both are types of cumulative sums.

One cumulative sum, CumSum, adds to its previous value, S_{t-1}, the value of the current sample, u_t and acquires thus the present value.

$$S_t = S_{t-1} + u_t$$

The initial value of the sum is zero, $S_0 = 0$. This method is particularly effective to detect a change of sign of the mean value of the signal.

An other cumulative sum, CuSum [14], sums up the input and subtract a fix value to compensate for disturbances. If the sum gets negative, it is reset to zero.

$$\begin{aligned} S_t &= S_{t-1} + u_t - v \\ S_t &= 0 \text{ if } S_t < 0 \end{aligned}$$

The initial value of the sum is zero, $S_0 = 0$, and v is a design parameter to compensate for disturbances. To detect a negative change in the input signal, a two-sided test can be preformed or the input signal can be squared.

Chapter 3

Engine Model

In this chapter the whole engine model is introduced, some adaptations are presented, fault models are created, behavioural modes are introduced whereupon the model is validated. Please refer to Appendix A for the resulting model. Figure 3.1 shows the Simulink implementation of the resulting model.

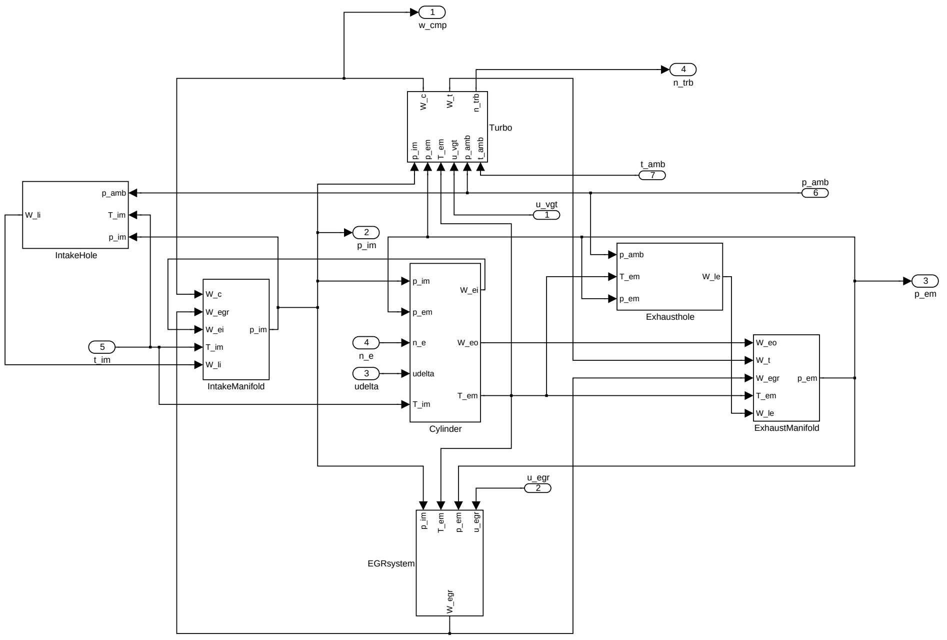


Figure 3.1. Overview of the Simulink implementation of the resulting engine model.

3.1 Existing Model

In the beginning of this thesis, some models for subsystems of the engine was considered as the basis for residual generation. But each submodel used at least one input signal that does not exist as actual measurement. Those values would have to be calculated from other measurements which would connect the submodels. Instead of using these models with the risk of not fully knowing the origin of some signals, a whole engine model presented in [26] was chosen as a base for this project. All submodels were still included in the engine model giving the residual generation method described in Section 2.6 the possibility to use any combination of the submodels.

Here follows a summary of the model, see [26] for the full model. The model is described in a state space form (3.1), where the state vector x_w contains the states (3.2), the input vector u_m contains the input variables (3.3), the control vector u_c contains the actuators (3.4) and the output vector y contains the output variables.

$$\begin{aligned} x_w &= f(x_w, u_m, u_c) \\ y &= g(x_w, u_m, u_c) \end{aligned} \tag{3.1}$$

The state vector contains the following states: boost and exhaust pressure, p_{im} and p_{em}, oxygen mass fractions on the boost- and exhaust-side, X_{Oim} and X_{Oem}, rotational speed for the turbocharger, ω_t, and two states for the position of the EGR-vault and the VGT, $\tilde{u}_{egr}$ and $\tilde{u}_{vgt}$.

$$x_w = (p_{im},\ p_{em},\ X_{Oim},\ X_{Oem},\ \omega_t,\ \tilde{u}_{egr},\ \tilde{u}_{vgt})^T \tag{3.2}$$

The input vector contains the following variables: ambient pressure, p_{amb}, ambient temperature, T_{amb}, boost temperature, T_{im}, and rotational engine speed, n_e.

$$u_m = (p_{amb},\ T_{amb},\ T_{im},\ n_e)^T \tag{3.3}$$

The control vector contains the following actuator variables: mass fuel injection, u_δ, EGR-vault position, u_{egr}, and VGT-position, u_{vgt}.

$$u_c = (u_\delta,\ u_{egr},\ u_{vgt})^T \tag{3.4}$$

Finally, the output vector contains the following variables: the massflow through the compressor, w_{cmp}, the pressure at the boost- and exhaust-side, p_{im} and p_{em}, and the rotational speed for the turbocharger, ω_t

$$y = (w_{cmp},\ p_{im},\ p_{em},\ n_{trb})^T \tag{3.5}$$

The states and variables of the state vector, the input vector, the control vector and the output vector can be seen in Tabel 3.1.

State or variable	Explication
p_{im}	boost pressure
p_{em}	exhaust pressure
X_{Oim}	boost oxygen mass fraction
X_{Oem}	exhaust oxygen mass fraction
ω_t	rotational speed for the turbocharger
$\tilde{u}_{egr}$	EGR-vault position
$\tilde{u}_{vgt}$	VGT position
p_{amb}	ambient pressure
T_{amb}	ambient temperature
T_{im}	boost pressure
n_e	rotational engine speed
u_δ	mass fuel injection
u_{egr}	EGR-vault position control value
u_{vgt}	VGT position control value
w_{cmp}	massflow through the compressor

Table 3.1. The states and variables of the state vector, the input vector, the control vector and the output vector.

3.2 Model Adaptation

Some changes were applied, partly to simplify the model, partly to adjust it for the toolbox it was subject to. The two states for oxygen concentrations before and after the cylinders, X_{Oim} and X_{Oem}, formed a subsystem that did not affect the other five states. The subsystem did not contain any sensors thus it neither added any redundancy to the system. The states were solely simulated values and were consequently removed. In future works the oxygen concentrations might however be used for improved fault diagnosis if oxygen mass fraction sensors are included in the engine. Additional subsystems that had neither influence to the remaining five states nor connection to any sensor were also removed.

These simplifications could of course been left undone resulting in a bigger model with equations that would serve no purpose for the final diagnosis system.

One modification that was not optional was the elimination of the iterative process to find the exhaust gas temperature. The method that would be applied needed either a continuous model or a discreet, not a mixture. By adding two new states representing the temperature in a cylinder, T_1, and the residual gas fraction, x_r, and creating their derivatives, the discrete model was made continuous the same way as in [17].

$$\dot{x}_r = \frac{x_{r,k+1} - x_{r,k}}{\Delta t} = \frac{\dfrac{\Pi_e^{1/\gamma_a} x_p^{-1/\gamma_a}}{r_c x_v} - x_r}{\Delta t} \tag{3.6}$$

$$\dot{T}_1 = \frac{T_{1,k+1} - T_{1,k}}{\Delta t} = \frac{x_r T_e + (1 - x_r) T_{im} - T_1}{\Delta t} \tag{3.7}$$

with

$$q_{in} = \frac{W_f q_{HV}}{W_{ei} + W_f}(1 - x_r) \tag{3.8}$$

$$x_p = 1 + \frac{q_{in} x_{cv}}{c_{va} T_1 r_c^{\gamma_a - 1}} \tag{3.9}$$

$$x_v = 1 + \frac{q_{in}(1 - x_{cv})}{c_{pa}\left(\dfrac{q_{in} x_{cv}}{c_{va}} + T_1 r_c^{\gamma_a - 1}\right)} \tag{3.10}$$

$$T_e = \eta_{sc} \Pi_e^{1-1/\gamma_a} r_c^{1-\gamma_a} x_p^{1/\gamma_a - 1} \left(q_{in}\left(\frac{1 - x_{cv}}{c_{pa}} + \frac{x_{cv}}{c_{va}}\right) + T_1 r_c^{\gamma_a - 1}\right) \tag{3.11}$$

The calculation of $T_{1,k+1}$ in the derivative of T_1, (3.7), uses $x_{r,k}$, not $x_{r,k+1}$ as in the iterative form. Looking at the equations from the Seliger cycle [8] that the iterative form was derived from, this does not seem to present a problem, which is validated by the mean relative error between $x_{r,k}$ and $x_{r,k+1}$ being less then 10^{-4}. In fact, x_r was found to be almost constant and to further simplify the model, this state was removed and x_r was replaced by a constant value that was found as the mean of x_r in a simulation. As a result x_v was not needed and therefore removed. The cooling of the gas from the engine was calculated using a constant temperature, T_w, instead of the ambient temperature.

$$T_{em} = T_w + (T_e - T_w)e^{-\frac{h_{tot}\pi d_{pipe} l_{pipe} n_{pipe}}{W_{eo} c_{pe}}} \tag{3.12}$$

The dynamic models for the EGR-vault and the VGT-position would serve to distinguish actuator faults from sensor faults. By removing the dynamics of the vaults from the model and replacing the actuator values by the measured positions, the model was further simplified. If it would be necessary to separate actuator faults from sensor faults in these two cases, a simple test containing only these models could be preformed. This modification enables neglection of the two states $\tilde{u}_{egr}$ and $\tilde{u}_{vgt}$.

These modifications resulted in a system consisting of the four states

$$x = (p_{im},\ p_{em},\ \omega_t,\ T_1)^T \tag{3.13}$$

All the equations describing the final model can be seen in Appendix A.

3.3 Fault Models

3.3.1 Leakage Fault Models

The leakages were modelled in two different ways in order to investigate the potential benefits of a more complex leakage model over a general leakage model:

- As a general leakage fault, W_{leak}.
- As a leakage area causing a leakage flow, A_{leak}.

General Fault

In this case the fault appears as in the pressure differential equations (A.1) and (A.2) as:

$$\frac{d}{dt}p = \frac{RT}{V}\left(W_{in} - W_{out} - W_{leak}\right) \tag{3.14}$$

Leakage Area Causing a Leakage Flow

Here, the fault is modelled in the a similar way as for the general fault, but the leakage flow, caused by a hole area, is modelled as a compressible flow through a restriction [15]

$$W_{leak} = \frac{A_{eff} p \Psi}{\sqrt{TR}}, \tag{3.15}$$

depending on the effective leakage area A_{eff} corresponding to A_{leak}, the pressure p before the restriction, a scaling function Ψ, depending on the pressure quotient $\Pi = \frac{p_{after}}{p_{before}}$, the temperature T before the restriction, and the gas constant R. The area of the leakage is considered constant, i.e.

$$\dot{A}_{eff} = 0 \tag{3.16}$$

It is assumed that the pressure inside the system is strictly higher than the pressure outside the leakage, $\frac{p_{after}}{p_{before}} < 1$, granting no back flow through the leakage into the system. This is a rather reasonable assumption that simplifies simulations of the model by setting a lower limit of the flow, equal to zero. The flow through a restriction can never be higher then sonic flow, i.e. 1 Mach. This occurs for the pressure quotient Π_{opt} that is calculated as

$$\Pi_{opt} = \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma}{\gamma - 1}} \tag{3.17}$$

This gives the upper limit of the flow.

The final limitations for the pressure quotient Π is described as

$$\Pi = \begin{cases} \Pi_{opt} & \text{if } \frac{p_{after}}{p_{before}} < \Pi_{opt} \\ \frac{p_{after}}{p_{before}} & \text{if } \Pi_{opt} \leqslant \frac{p_{after}}{p_{before}} \leqslant 1 \\ 1 & \text{if } 1 < \frac{p_{after}}{p_{before}} \end{cases} \tag{3.18}$$

On the boost-side the scaling function Ψ_{li} is calculated as

$$\Psi_{li} = \sqrt{\frac{2\gamma_a}{\gamma_a - 1}\left(\Pi_{li}^{2/\gamma_a} - \Pi_{li}^{1+1/\gamma_a}\right)} \tag{3.19}$$

The scaling function (3.19) does not give enough accuracy for the flow through the EGR-vault according to [26]. Instead a parabolic function is used. Since the leakage on the EGR-side in this thesis is modeled in the same environment (temperature, pressure and pressure pulsations from the cylinders) and in the same way (compressible flow through a restriction), the parabolic function is used in the leakage model at the exhaust-side.

$$\Psi_{le} = 1 - \left(\frac{1 - \Pi_{le}}{1 - \Pi_{le,opt}} - 1 \right)^2 \tag{3.20}$$

Due to time limitations, only the second way of modelling the leakage was thoroughly tested. It is thus this leakage model that is included in Appendix A.

3.3.2 Sensor and Actuator Fault Models

Actuator faults and sensor faults are both modeled as general faults: sensor reading = actual value + sensor fault. All sensors (p_{amb}, T_{amb}, T_{im}, n_e, w_{cmp}, p_{im}, p_{em}, n_{trb}) and actuators (u_δ, u_{egr}, u_{vgt}) are considered as potentially fautly.

3.4 Implementation

A Simulink implementation of the existing model, available at Scania, is used. By modifying the implementation, the adaptations in Section 3.2 are included in the Simulink model, as well as the leakage models from Section 3.3.1.

3.5 Behavioural Modes

The behavioural modes considered in this thesis can be viewed in Table 3.2.

Notation	Behaviour
NF	No Fault
AD	error in Actuator for Diesel injection
AE	error in Actuator for EGR-vault
AV	error in Actuator for VGT-vault
LB	Leakage on Boost-side
LE	Leakage on Exhaust-side
STA	error in the Sensor measuring Temperature of Ambient air
SPA	error in the Sensor measuring Pressure of Ambient air
SPB	error in the Sensor measuring Pressure at Boost
STB	error in the Sensor measuring Temperature at Boost
SPE	error in the Sensor measuring Pressure at Exhaust
SRE	error in Sensor for Rotation of Engine
SRT	error in Sensor for Rotation of Turbine
SA	error in the Sensor measuring Airmassflow

Table 3.2. All considered behavioural modes and their notations.

3.6 Validation

The model provided in [26] was already validated in that work, having a mean relative error smaller than 12%. The modified version of this model created in [17] where the iterative parts were removed was verified in that work using simulations, and found to have a mean relative error of less then 0.4% compared to the non-modified model. The validation of the model including the additional modifications made in this work was done using real measurements from a driving cycle without faults. A mean relative error of 15% was found, to be compared with the original model in [26] having a mean relative error of 12%. The hole models as separate systems were not validated in the end, since too little measurement data was available.

Chapter 4

Experimental Setup and Data Acquisition

In this chapter the experimental setup used to collect data for model validation and test creation is shown whereupon the data acquisition principles are outlined.

4.1 Experimental Setup

In order to examine the system with a leakage compared to its normal condition, holes needed to be implemented in the truck. Some additional sensors were also needed in order to verify the fault models.

4.1.1 Hole Location

The airpaths in the engine were divided into groups, seen as dashed areas in Figure 4.1. These areas were chosen because a leakage is reasoned to have the same impact on measured signals no matter where in the area it is present. The flow caused by the leakage depends on the pressure on each side of the hole, the leakage area and the temperature before the leakage as well as on some constants, as has been explained in more detail in Section 3.3. At the exhaust-side, the pressure and temperature is more or less the same within the whole dashed area. The boost-side, however, contains three coolers and hence the temperature is more or less guaranteed to be different in different parts of the area. This is nevertheless not a big problem since the pressure and the pressure quotient will have a greater impact on the flow than the temperature, according to the model of the leakage flow that is used in this thesis. Since the coolers are approximated as ideal with no pressure fall over them, it is not unreasonable to model equal pressure in all pipes, in each of the areas.

According to the argumentation above, one hole should be placed in each leakage area and their exact location should not have a big influence. After a discussion with mechanics and supervisors a decision was made to produce one hole at the boost-side between the compressor and the intercooler and another hole in the beginning of the exhaust-side, right before the EGR-vault. The holes were constructed in the middle of the pipes due to space considerations. A benefit with this hole location was that extra pipes could be brought in and worked upon separately before replacing the corresponding existing pipes in the truck in one single operation. This avoided the need to occupy the truck during a longer period.

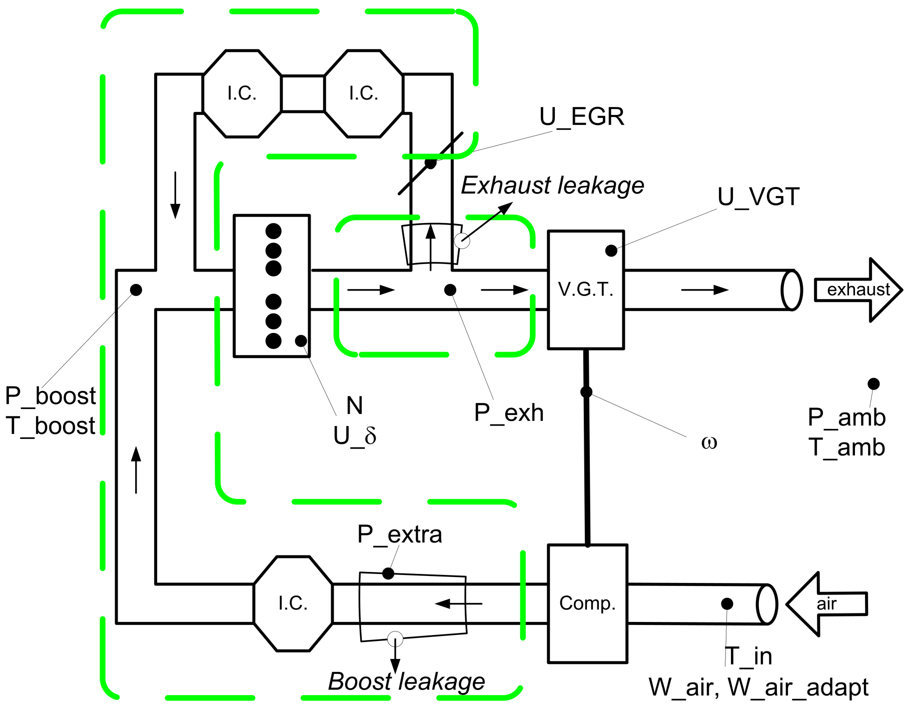


Figure 4.1. Sketch of the gas flow system in a diesel engine with EGR and VGT but without a catalytic converter or a particle filter. Sensors and actuators present in the system are marked as dots on the sketch as well. Leakage areas are marked with dashed lines.

4.1.2 Hole Implementation

Depending on how the leakage occurs in reality, the magnitude of the leakage might vary and therefore it is desirable to be able to change the size of the hole. An approach previously used in [21], with drilled holes in exchangeable bolts, was therefore used. The next step was to decide which hole diameters that would be interesting to acquire measurements from. Preferably the diameters should be in intervals from zero up to the critical size causing the engine to perform so badly that it is noticeable for a driver. This would give an overview of the engine behaviour when subject to leakages of different sizes and provide measurements with which a smallest detectable hole diameter can be decided. On the boost-side, the holes were chosen from 0 mm to 25 mm in diameter in steps of 5 mm. On the exhaust-side, where the pressure is higher, holes from 0 mm to 15 mm in steps of 3 mm were used.

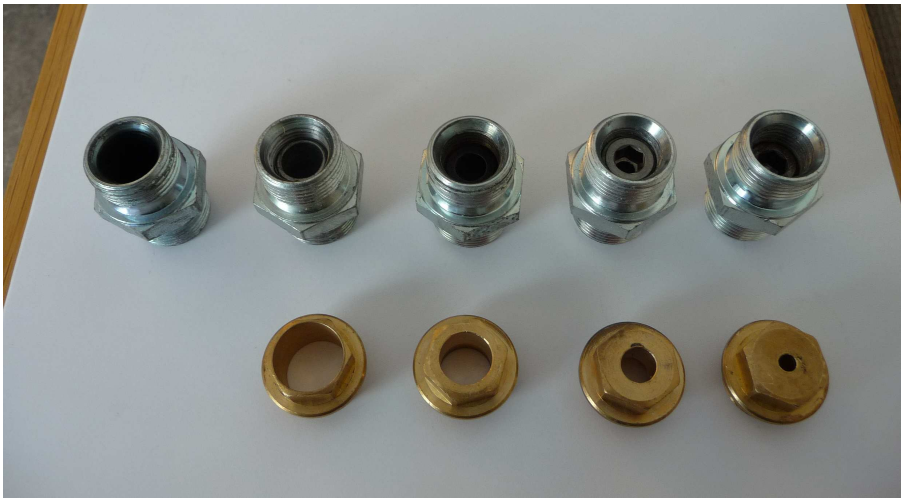


Figure 4.2. Bolts used to create different sizes of leakages. The upper bolts are for the EGR leakage, and have diameters from 3mm to 15mm in steps of 3mm. The lower bolts are for the boost leakages, having diameters from 5mm to 25mm in steps of 5mm. For the largest leakage on the boost-side, no bolt is used since the diameter of the bolt is as large as the desired hole diameter.

4.1.3 Additional Sensors and Tubes

With the possibility to change the size of the hole, the next challenge was to reroute the gas from the leakages. This is particularly important for the exhaust gas coming from the exhaust-side due to temperature reasons. The gas mixture at approximately 500 degrees Celsius would otherwise escape through the leakage right under the cab. A longer tube was therefore needed outside the bolt, to lead away the exhaust gas and letting it out behind the cab. In the tube a mass flow meter was supposed to be mounted. A similar tube for the boost-side was also needed for the implementation of extra sensors. The sensors in the tubes were to be implemented in order to validate fault models for the leakage areas or the leakage flows.

These sensors were not intended for diagnosis, but only for validational purposes. The main idea was to pick local sensors that was already being used at Scania for low cost and fast deliveries. A mass flow sensor based on the cooling of a platina wire was found, the same kind used to measure the mass flow through the compressor. This sensor was chosen for the boost-side but could however not be used on the exhaust-side due to high temperatures and the composition of the gas. A turbine-like sensor would not work either since it would clog due to soot according to an expert engineer at Scania with experience in the field of EGR-measurements. The employee claimed that a differential pressure sensor was the best way to measure the flow. Using this technique the volume flow would be obtained. With additional sensors measuring the pressure and the temperature, the mass flow could then be calculated.

The platina sensor was to be fitted in a tube and the differential pressure sensor comes premounted in a tube. The dimensions of the tube on the boost-side needed to be chosen wisely in order to keep the flow per area unit within a specified range. This was important in order to get a certain measurement accuracy. The tubes would also serve to stabilise the flow, which would augment the accuracy of the measuring but filter high frequency changes in the flow. Generally the following applies: the higher the flow per area, the better the measurement accuracy but the bigger the flow resistance. A high flow resistance would affect the leakage flow and result in a less realistic implementation of the leakage. Thus the choice of the tube dimensions was a trade off between measurement accuracy and leakage flow model accuracy.

In order to choose the tube dimensions for the boost leakage, an estimate of the leakage flow rate was needed. For small leakage areas this was done by looking at max values of the pressures at the intake- and the exhaust-side and then calculate the flow through the leakages according to the leakage models. The measurement data was taken from test drives in the mountains in Spain to assure extreme conditions. For bigger leakage areas, the fact that a leakage will lower the pressures must be taken into consideration. Actual measurements or simulations of the engine including the control algorithms in the case of leakages would be needed. However, no leakage measurements were available at this time. No Matlab implementations of the actual control algorithms in the truck were found either. Some searching revealed that introductory simulations of leakages had been carried out at one of the departments at Scania. Using their simulation results along with calculated extream value approximations as well as expert knowledge, an approximation of the flow rates could be done. The flow through the boost-side leakage was estimated to be in the range of 0-15 kg/min and the exhaust-side leakage in the range of 0-5 kg/min.

Having estimated the max flow rate through the holes with the biggest area for the boost- and the exhaust-side and the minimal flow rate for the respective holes with the smallest area the tube dimensions could be calculated. Finally the dimensions were chosen so that the tubes would have a diameter about twice the diameter of the biggest hole. With these dimensions it was assumed that the pressure loss due to the tube would be a neglectable error in comparison to the error caused by the leakage implementation. In reality the leakages would hardly be perfectly circular and they normally occur at the joints, not in the middle of the pipes as would be the case in this thesis. The platina wire massflowmeter found at Scania, was considered appropriate to use and the specifications for the differential pressure sensor could now be made.

The leakage at the boost-side would be implemented before the intercooler. The pressure sensor that would be used to calculated the flow through the leakage is situated after the intercooler. Ideally, there is no pressure loss in the dashed areas, see Figure 4.1, where both the leakage and sensor are located. However it was considered interesting to add an extra pressure sensor on the same side of the intercooler as the leakage to see how big the difference really was. This sensor would also serve to validate the leakage model used in the boost-side. Drawings of the tubes as well as the bolts for the holes were created in cooperation with the mechanics that would build them after having measured the available space under the cab.

In the end no differential pressure sensor was implemented since the price and the delivery time was too considerable. The platina wire sensor and the pressure sensor was however found locally at Scania and was handed to the mechanics together with the drawings for the pipes. When the pipes were ready, they were assembled in a truck.

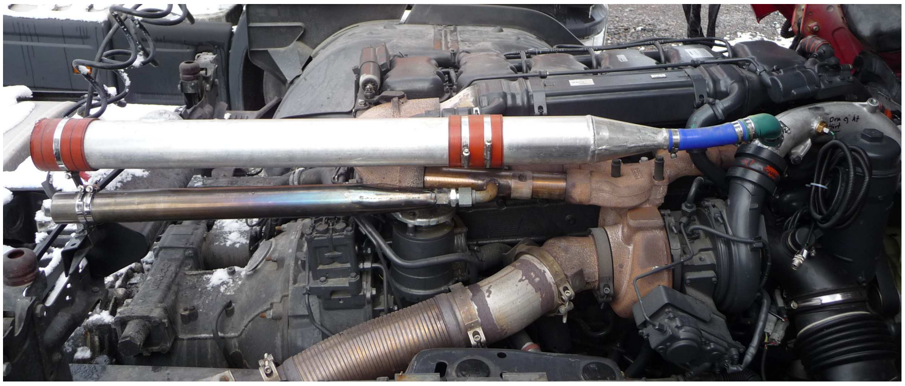


Figure 4.3. Overwiev of the new hardware for the leakage implementation in a truck. The upper of the two tubes in the middle of the photo is the tube for the gas from the boost-side leakage, and the lower is the tube for the gas from the exhaust-side leakage. For closeups of the holes, see Figure 4.4 for the boost-side, located to the right in the photo, and Figure 4.5 for the exhaust-side, located in the middle of the photo.

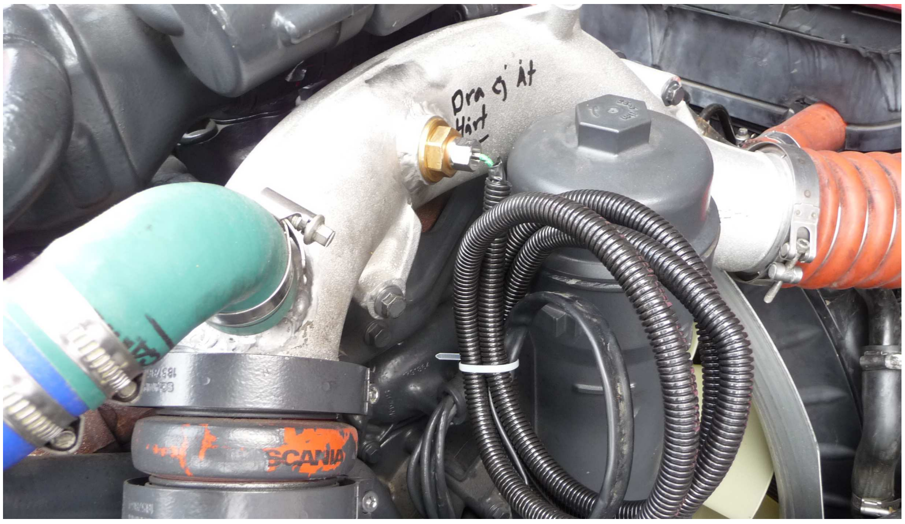


Figure 4.4. Closer view of the boost-side leakage implementation. The tube disappearing to the left encloses the leakage bolt. The extra pressure sensor can be seen in the middle of the photo with its rolled-up cable.

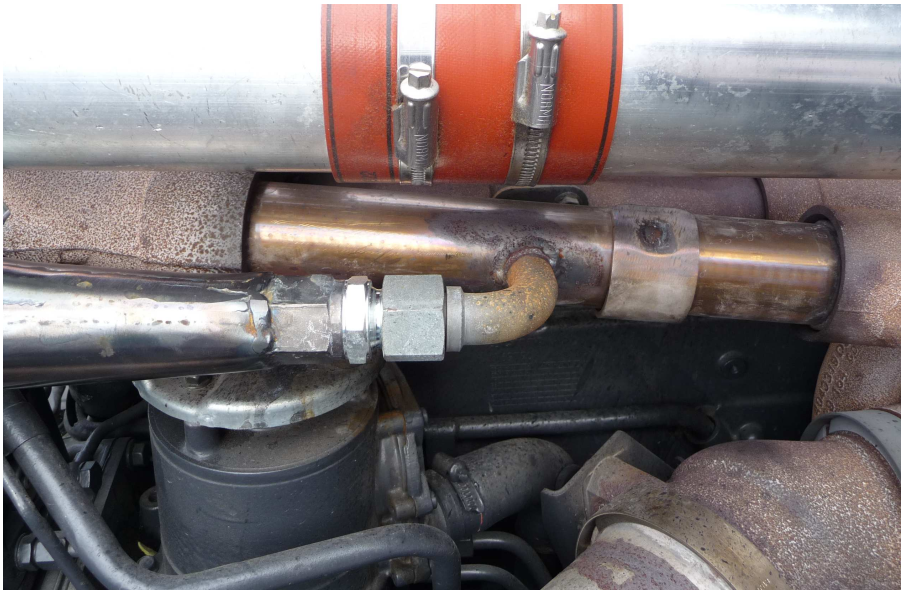


Figure 4.5. Closer view of the EGR leakage implementation. The bent pipe in the middle of the photo leads to the leakage bolt that is the splice between the two big screw nuts right after the bend.

4.1.4 Data Acquisition Hardware and Software

All sensor and actuator values in the truck could be accessed from the CAN-bus, an internal data bus, and logged with a standard laptop equipped with Vision, a measurement software. The additional sensors for pressure and air massflow were connected to Vision through Vision Electronic Data Acquisition (EDAQ) hardware, ensuring synchronised data. Cables between the additional sensors and the EDAQ hardware were built as well as cables and hubs ensuring power supply from the cigarette lighter of the truck.

The mechanics helped building an electronic device which could be used to introduce a bias, gain or constant value fault to two arbitrary sensors or actuators. Unfortunately there was no time to log data with this device in this thesis but it can come in handy in the daily work of the diagnosis department at Scania.

4.2 Data Acquisition Procedure

The measurements used in this work were gathered on a test track at Scania. A driving cycle was planned which included many different operating points and accelerations/decelerations. The truck was equipped with a ballast frame in order to mimic regular operating conditions. The retarder, a hydralic braking system, was also used selectively as a way of simulating heavy loads. At each driving cycle one fault was to be active. A fault could be any of the different leakage areas at the boost- or the exhaust-side or a sensor or actuator fault.

With the truck equipped with the leakages and the extra sensors, and the data acquisition hardware and software provided it was good to go. Despite rigid planning some problems occurred during the test drives due to play in some contacts, USB-limitations and icy roads. As a result of the current time considerations it was decided to concentrate on acquiring measurements from three different hole sizes per leakage area and from one general fault per pressure sensor. The decision to selectively log faults for the pressure sensors was motivated by the fact that these faults are the ones believed to be the hardest faults to isolate from leakage faults and vice verse.

Chapter 5

Test Quantity Generation

In the following sections there will be a brief presentation of the methods used in this thesis for generating test quantities. The starting point is a model containing analytical redundancy, see Section 2.3.2. The steps involved can be seen in Figure 5.1, with the new functionality contributed by this thesis encircled with dashed lines. The implementation of the steps from MSO to residual were heavily improved during the course of this thesis.

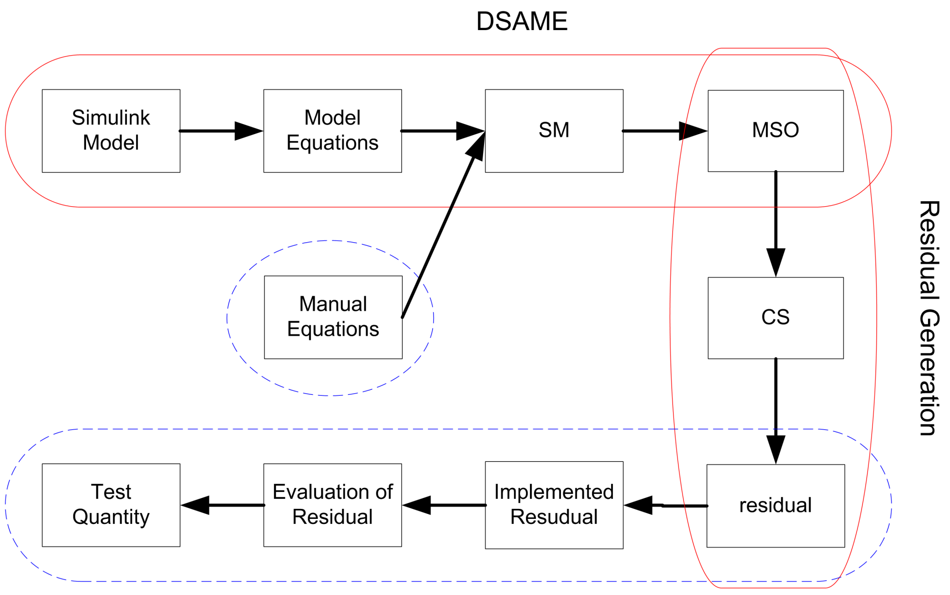


Figure 5.1. The process from a model to an evaluated residual. New functionality encircled with dashed lines. Heavy improvements were made to the implementation of the steps between MSO to residual.

5.1 Extraction of Model Equations from Simulink Models

The model equations can be extracted from a Simulink model using a MATLAB toolbox called DSAME. This toolbox is described in [11] and is developed by Scania and Linköping University. The extraction from a Simulink model is illustrated by Example 5.1.

Example 5.1: Simulink Model Extraction

The equations to the right in (5.1) is an example of how DSAME would rewrite the Simulink implementation, see Figure 5.2, of the equation to the left in (5.1). It can be seen that the extracted equation system gets larger (more equations) than the original equation system since DSAME creates "dummy equations", i.e. one equation is transformed to many.

$$
\begin{array}{ccl}
\text{Equation in Simulink} & & \text{DSAME representation} \\
 & & a_1 = u \\
 & & a_2 = 2 \\
 & & a_3 = a_1 a_2 \\
 & & a_4 = x_1 \\
y_2 = 4 - Ax_1 + 2u & \Longleftrightarrow & a_5 = A \\
 & & a_6 = a_4 a_5 \\
 & & a_7 = 4 \\
 & & a_8 = a_7 - a_6 + a_3 \\
 & & y_2 = a_8
\end{array}
\tag{5.1}
$$

5.2 Structural Methods to find MSOs from Model Equations

Using the theory presented in Section 2.4 the equations extracted from a model can be transformed into an SM where after all MSOs from the SM are generated. The process is illustrated in Section 2.4 and is here recalled for readers convenience.

An extracted equation system

$$
\begin{aligned}
e_1 &: \dot{x}_1 = x_2 \\
e_2 &: x_2 = -Ax_1 + u \\
e_3 &: y_1 = x_1 \\
e_4 &: y_2 = x_2
\end{aligned}
$$

is transformed to the structural model

$$
\begin{array}{c|cc|ccc}
 & x_1 & x_2 & u & y_1 & y_2 \\
\hline
e_1: & 1 & 1 & 0 & 0 & 0 \\
e_2: & 1 & 1 & 1 & 0 & 0 \\
e_3: & 1 & 0 & 0 & 1 & 0 \\
e_4: & 0 & 1 & 0 & 0 & 1
\end{array}
$$

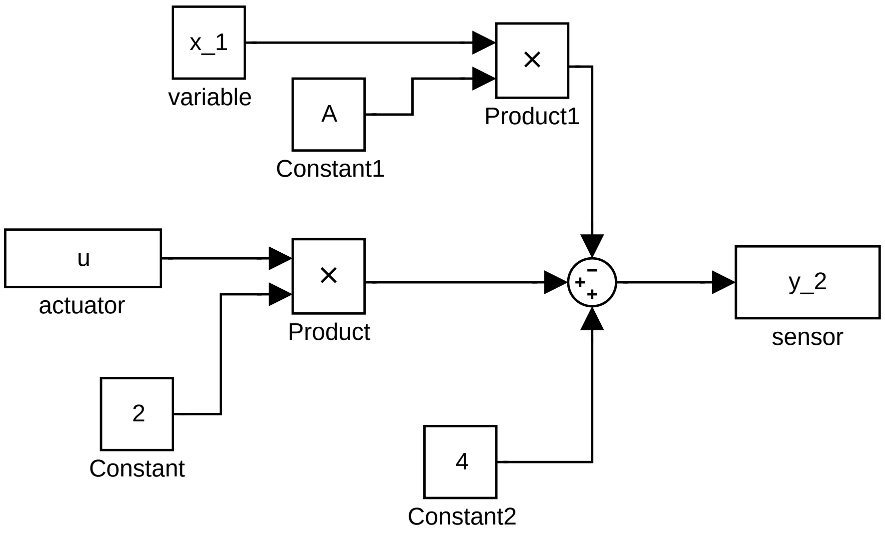


Figure 5.2. Simulink implementation of the equation to the left in (5.1) that is extracted by DSAME to the equations to the right in (5.1).

containing the four MSOs $\{e_1, e_2, e_3\}$, $\{e_1, e_2, e_4\}$, $\{e_1, e_3, e_4\}$ and $\{e_2, e_3, e_4\}$.

5.2.1 Sorting of MSOs into Classes

After having found the MSOs, they are divided into classes. Each class is theoretically sensitive to a unique combination of faults. It is thus in theory not necessary to use more than one MSO from each class for test quantity generation. Example 5.2 illustrates the sorting of MSOs, found in a previous model, into classes.

The toolbox DSAME is used to generate the SM and all MSOs, as well as sorting them into classes.

Example 5.2: MSO classes

Assume that the only faults in the exemplifying Simulink model in Figure 2.4, are associated with the actuator, f_u, and the two sensors, f_1 and f_2 respectively. The first MSO generated for this model, see Example 2.10, is formed out of the equations

$$\begin{aligned} \dot{x}_1 &= x_2 \\ x_2 &= -Ax_1 + u \\ y_1 &= x_1 \end{aligned} \tag{5.2}$$

and is sensitive to f_u and f_1 since sensor y_1 and actuator u are present in these equations. See Table 5.1 for the sensitivity for all MSOs generated from Example 2.10.

	f_u	f_1	f_2
MSO_1 :	x	x	
MSO_2 :	x		x
MSO_3 :		x	x
MSO_4 :	x	x	x

Table 5.1. A table showing which faults that effects the different MSOs found in the Simulink model in Figure 2.4.

In this example, each MSO has a unique set of faults and therefore each MSO will form its own class. If there would have been a fifth MSO, MSO_5, sensitive to f_1 and f_2 it would be in the same class as MSO_3.

5.2.2 Selection of MSOclasses

In an MSO that does not contain many equations, like the one in Example 2.10, there are few CS to try. Recall that there is one potential CS for each equation, as discussed in Section 2.6. Since the equations in this particular example are rather simple, these computations do not take very long time to execute. The time to obtain a computation sequence grows with the size of the equation system but more importantly with the complexity of the equations. As a consequence, a large equation system with many complex equations might take hours to solve, if a solution at all is found. This is of course dependent on the solving tool used. In a situation where there are many more classes of MSOs than faults, a well chosen subset of classes should still be enough to detect and isolate at least all single faults from each other. But how to choose as few classes as possible and still have a big enough selection to isolate all faults? The method choosen for this task in this thesis, is the Minimal Hitting-Set method, explained in Section 2.3.3.

5.3 Residual Creation and Implementation

When a subset of classes is found, one MSO from each class in the subset is selected. With the theory in Section 2.5, computation sequences are found. Recall the computation sequence for $\{x_1, x_2\}$ found in Example 2.11: $C_1 = ((x_1, e_3), (x_2, e_2))$, i.e. compute x_1 from e_3, then x_2 from e_2. The CS is used with a residual equation to create a residual. This residual and all other residuals used in this thesis are found using a Matlab implementation of the residual generation approach outlined in [24]. Matlabs solver tool Maple is used as solving tool for algebraic equations, and mixed causality is considered.

In order to be able to evaluate and choose residuals before introducing them in a diagnosis system in the truck, the residuals need to be implemented for offline testing. Three different approaches are suggested in this work: an implementation in Simulink built with blocks, an Matlab m-file function implementation or a C-code function implementation. In all three cases the input data is observations and the value of the residual is the output. Somewhere within the implementation the constants needed for the residual, as well as the computation sequence, is defined. In a first approach, all observations are given to all residuals as well as all constants declared, not only the observations and constants that are needed.

Some advantages/disadvangages with the different approaches are seen in Table 5.2.

	Advantage	**Disadvantage**
Simulink implementation:	Easy to visualize. All in one model.	Complex to implement automaticly.
MATLAB m-file function:	Simple to implement and execute. Measurements already in MATLAB format.	
C-code function:	Closer to final implementation in a truck.	Import of MATLAB data.

Table 5.2. Some advantages and disadvantages with different three different methods proposed for implementation of residuals.

Both the Simulink method and the m-file method are tested, but only the m-file approach is used due to implementation practical reasons.

To facilitate the implementation, a choice is made to use computation sequences with integral causality or pure algebraic relations. Integral causality leads to a residual on a state space form, which can easily be implemented using Euler forward approximation with fixed step length of a sample period, 0.1 s.

5.4 Evaluation of Residuals

5.4.1 Stability

Just because a residual is found and can be generated, does not mean that it can be used. It might be unstable even during normal driving conditions for fault free input data. Earlier works [2] have suggested methods to examine and evaluate the stability of residuals. These are considered if the stability of a large number of residuals need to be examined.

5.4.2 Operating Conditions

No model can perfectly describe the reality. It is common that a model, and as a consequence the residuals based on that model, is more accurate under certain operating conditions. For example, a pressure within a specific range, a temperature below a certain level or during high engine torque. If favorable operating conditions can be found, the residuals can be used solely during these and thus become more reliable. The accuracy of the residual might also depend on a combination of different engine variables within certain limits. In this thesis only a search for single dependencies is made since dependencies of every combination of variables is such a big task that it is considered unfeasable. The investigation of single dependencies is carried out for each residual, by studying its plot against plots of specific variables.

5.5 Designing Test Quantities

Depending on the appearance of the residual, the signal processing methods presented in Section 2.7 are applied. A noisy residual is primarily subject to lowpass filtering. If the mean value changes with the fault, a sliding mean approach is tried. If the result is not satisfying in the sense that fault detection becomes significantly easier, a recalibration of the method or a different signal processing method is tried. This is in many ways an iterative process. The processed residual is used as a test quantity.

Chapter 6

Results

6.1 Fault Measurements

New hardware has been created mainly in order to acquire real leakage measurements under controlled forms. The main hardware contributions are tubes, bolts and cables. This hardware as well as the method in which it was used can be employed in future works when for instance leakage data is needed.

Leakage measurements were gathered for circular holes with diameters 5, 15 and 25 mm on the boost-side and 3, 9 and 15 mm on the exhaust-side. Measurements from a negative bias fault in the boost pressure sensor as well as in the exhaust pressure sensor, with the size of about 10 percent of the average pressures at the respective sides, were also gathered.

6.2 Simulink Implementation

When the modified engine model, Section 3.2, was implemented in Simulink and a structural model was created with DSAME, the structural determindeness, Section 2.4.2, of the system could be calculated. The modified model was a +2 system. This was very surprising since the un-modified model was a +3 system. The source of the problem was found to be the implementation of the exhaust-side leakage. This was even more surprising since the leakage model contained four unknown variables and four equations. Thus the structural over- or under-determinedness of the system should remain the same. The reason for this behaviour was that the Simulink implementation of the exhaust-side leakage used the same blocks as for the EGR-vault, which contained some blocks that DSAME did not handle correctly. After having changed the blocks in question for the EGR-vault model and for the leakage model on the exhaust-side, the system turned out to be a +4 system.

6.3 Method for Residual Generation

6.3.1 Extraction of MSOs from Simulink Model

Using the toolbox and methods presented in Section 5.1, the model equations were extracted from the Simulink model. The extraction provided many dummy equations, see Section 5.1, and the 52 equations describing the engine model became 267 after the extraction. The variable names became very long since the names include all names of the blocks that contains the variable.

To avoid the long variable names and the enlargement of the equation systems, some new functionality was created to build model equations from a textfile-defined model. This also avoided implementational faults since no blocks were used that might lead to misinterpretation.

The change of blocks in the Simulink implementation increasing the determinedness, see Section 6.2, resulted in the finding of about 600 MSOs, sorted in 300 MSO classes. A significant difference compared with the previous 90 MSOs in [17] for the un-modified model. It also made it theoretically possible to detect faults in the EGR-vault actuator, which was not possible before the change of blocks.

6.3.2 Residual Generation from MSOs

Residual generation proved to be the major work in this thesis since a lot of debugging and adaption of the MATLAB implementation was needed.

The process of finding computation sequences could take very long time, especially if it involved solving complex equation systems, generally connected to large SCCs, with MAPLE. Sometimes the whole program froze. In an attempt to reduce the freezes and thus avoid the need of restarting the program and the process, new functionality was introduced allowing to abort the search for a computation sequence if SCCs above a certain size was found. It was found that if allowing SCCs up to size 7 almost no freezes occured. Other changes were also introduced to the MATLAB implementation, saving the work more continuously and allowing to track which SCC that causes the program to freeze.

Some equation systems proved to complex to find explicit solutions to, with the available tools for equation solving. In the case that implicit solutions were found, they were discarded in this thesis.

In other equation systems there were explicit solutions, but not unique ones. In quadratic equations, when solving for the squared variable, there are quite naturally two solutions. Many of these variables have limitations making only one of the solutions physically correct. But which one? And if the solutions are used in another quadratic equation there will be four different solutions, only one being correct. The proposed solution to this problem was to present all roots, i.e. solutions, to the equations and notify the user that the incorrect solution(s) should be removed.

6.3.3 Number of and Implementation of Residuals

Each of the 600 MSOs might produce approximately 45 residuals (an estimate of the average number of equations per MSO) resulting in about 27 000 residuals in total. There was not enough time to find and evaluate all the possible residuals since this would have taken weeks with the computation power available and with the stability issues outlined in Section 6.3.2. Many MSOs proved to be so complex that no computation sequences and thus no residuals could be found with the available solving tools. In spite of that, residuals were found in more than 100 MSOs using mixed causality and a maximal SCC size of 7. Out of those residuals, four residuals that did not use derivative causality were implemented and evaluated. One of those residuals, an algebraic one, can be found in Appendix B to illustrate the appearance of a residual.

The residuals were first implemented in Simulink as S-functions or embedded m-files. This was rather time consuming and hard to implement automatically for every residual. An easier way was to create m-file functions with the observations as input and the value of the residual as output. Within the function was a declaration of constants, the computation sequence and the residual equation.

These residual functions were created by another function and could therefore be auto-generated for each found computation sequence. Another function was written to calculate the values of the residuals given measurements.

The wish to automatically generate the residual code from a found computation sequence, see Section 6.3.2, lead to the creation of some new functionality. This functionality only supports integral causality but could, without too much work, be modified to use derivative causality and/or mixed causality.

6.4 Test Quantity Generation

When the first residuals were implemented and evaluated, it became clear that the residuals did not behave according to the influence structure. It was shown that certain faults affected residuals that should not be sensitive to the faults in question. Some faults that should make the residual respond more strongly than in the fault free case, actually made it diverge less from zero. See Figure 6.1 for an example of a residual that responds to a fault it should not react to according to its influence structure.

One residual for example, gave a significant reaction to a leakage on the boost-side, while it should not be sensitive to any leakages. A possible explanation is that this fault, a boost-side leakage, has a similar effect on the system as a faulty massflow sensor affected by a positive offset, the latter being a fault that the residual should be sensitive to. Simplified, the residual is based on the difference between massflow in, W_{in}, and massflow out, W_{out}. If W_{in} shows a too big value, the flow in will be greater than the flow out. If there is a leakage, the flow out will seem smaller than it is, and thus the flow in will be greater than the flow out.

6.4.1 Residual Processing

After the residual signals were calculated, they were lowpass filtered [13]. A discussion was held as to the need of filtering the measurement data before the residuals were calculated. This did not seem necessary since the residuals gave a rather good result when plotted. The residuals were plotted merging a fault free driving cycle and each of the

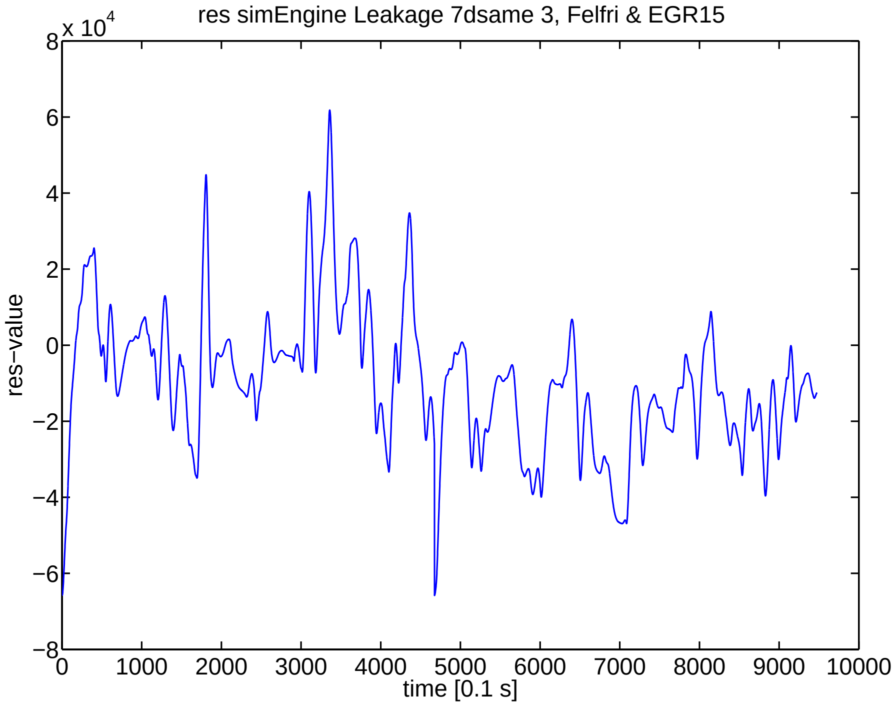


Figure 6.1. A residual that should not react to a leakage on the exhaust-side according to the influence structure, but does react to it. The first half is fault free measurements, the second half is measurements with a leakage on the exhaust-side. The residual has been filtered in order to more easily see the change of mean value betweeen the fault free case and the leakage.

driving cycles for the different fault respectively. That way, a virtual driving cycle was produced that was fault free in the start whereupon a fault occurred midways.

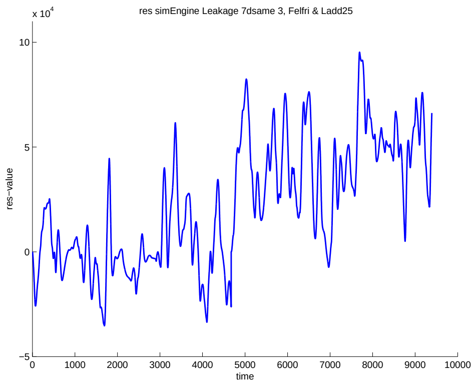
Most residuals were too fluctuating to set reliable thresholds. This was due to model uncertainties and disturbances. For some of the residuals it was clear that the mean value changed with a fault, so a cumulative sum was tried as a signal processing method for the residual, see Section 2.7. But first all residuals were adjusted with the mean value of a representative fault free case, giving the residual a mean value of zero. This might be tricky to do in reality since each truck and sensor set is individual.

The mean value of some residuals increased for certain faults, and decreased for other. See for example Figure 6.2 where it can be seen that the mean value of the residual decreases for an exhaust-side leakage or the sensor fault on the exhaust-side, and increases for a boost-side leakage or the sensor fault on the boost-side.

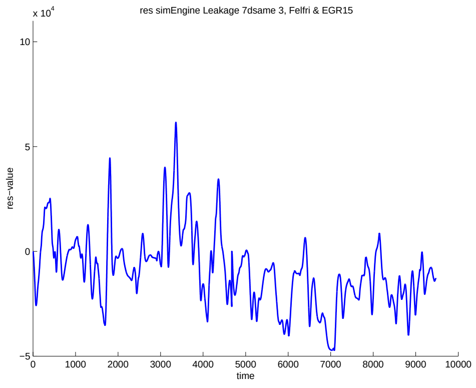
With two tests from that residual, one that detects increased mean values and the other detecting decreased mean values, faults can be located to either the exhaust-side or the boost-side with the decision structure:

	LE	LI	SPE	SPB
$T_1(resC\ up)$	0	x	0	x
$T_2(resC\ down)$	x	0	x	0

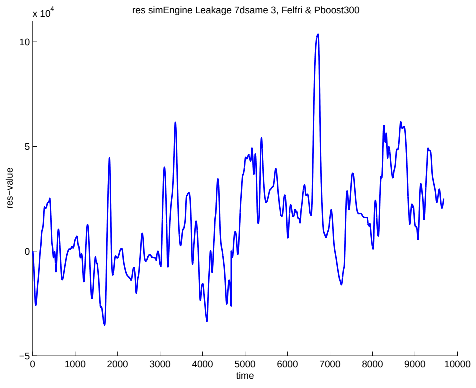
where LE is a leakage on the exhaust-side, LI is a leakage on the boost-side, SPE is a fault in the exhaust pressure sensor and SPB is a fault in the boost pressure sensor.



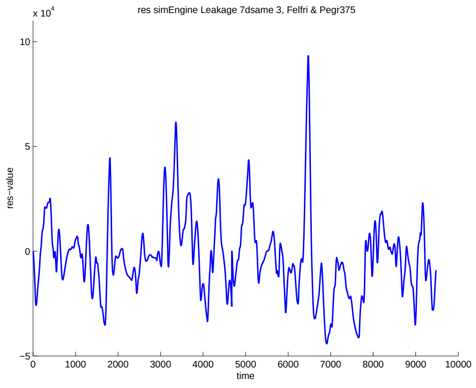
Residual C for a 25 mm boost-side leakage.



Residual C for a 15 mm exhaust-side leakage.



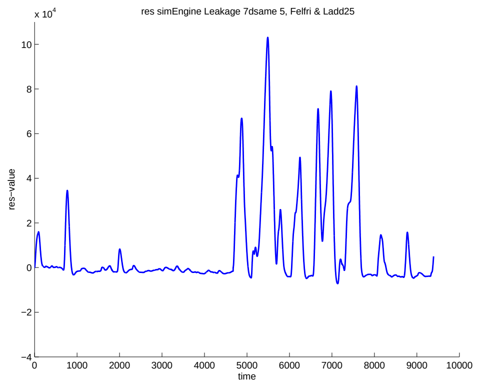
Residual C for a negative bias fault on the boost pressure sensor.



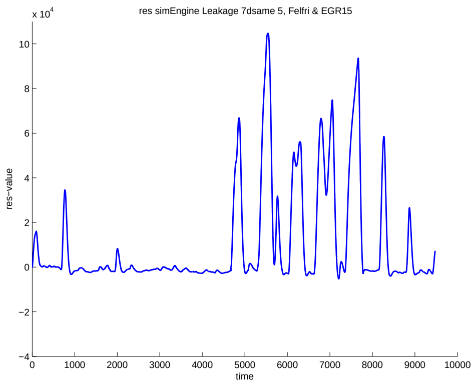
Residual C for a negative bias fault on the exhaust pressure sensor.

Figure 6.2. The four plots show residual C for a leakage on the boost-side, a leakage on the exhaust-side, a fault on the boost pressure sensor and a fault on the exhaust pressure sensor. It can be seen that a problem originating from the boost-side increases the value of the residual, while a problem originating from the exhaust-side decreases it.

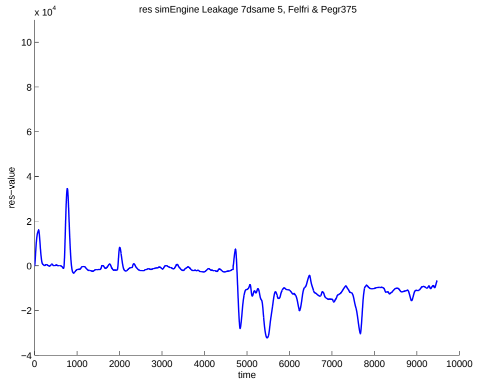
Another residual, see Figure 6.3, did not only change mean value, but it was also noticed that a fault on the boost pressure sensor caused the residual signal to take imaginary values. The residual did also seem to have different dynamic properties for different faults. This caused the residual to react more strongly during certain conditions.



Residual D for a 25 mm boost-side leakage.



Residual D for a 15 mm exhaust-side leakage.



Residual D for a bias fault on the exhaust pressure sensor.

Figure 6.3. The three plots show residual D for a leakage on the boost-side, a leakage on the exhaust-side and a fault on the boost pressure sensor. A fault on the exhaust pressure sensor caused the residual signal to take imaginary values and is therefore not shown.

6.4.2 Operating Conditions

Not very surprisingly, there were situations when some residuals worked better. By choosing to run the tests based on these residuals only during those conditions, the performance of the diagnosis system could be improved. Using these conditions, the problem that residuals did not react as foreseen to certain faults, could in some cases be handled and compensated for. In some cases, the signal could be enhanced or attenuated which helped the detection.

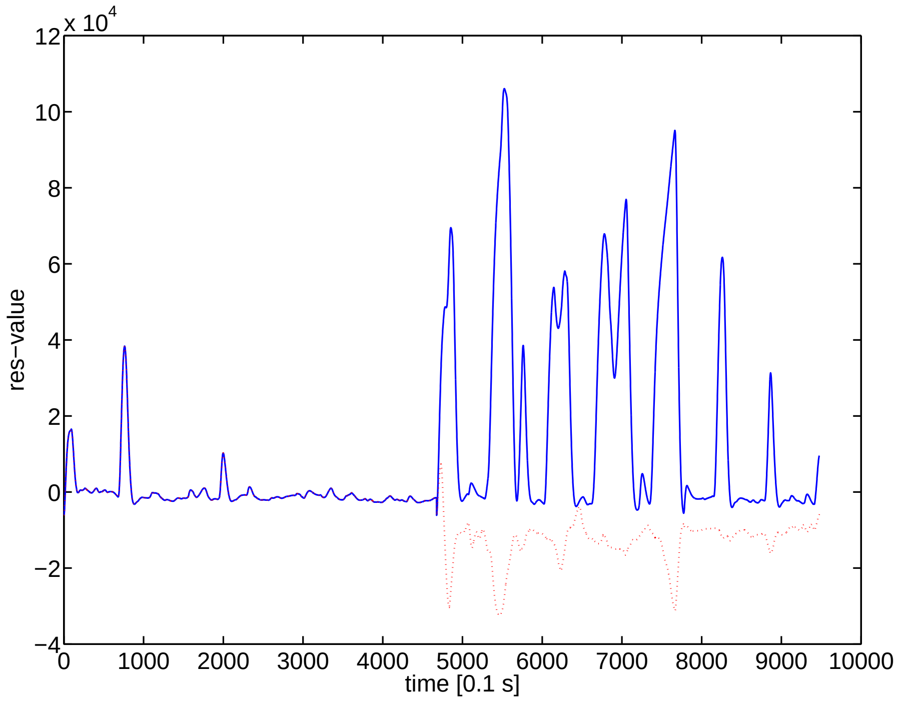


Figure 6.4. A residual that reacts differently to a fault in a pressure sensor (dashed line) and a leakage at the exhaust side (solid line). The first half is fault free measurement data, the second half is from two corresponding driving cycles for the two different faults. Notice that the base value of the leakage data seams to be zero with peaks during some periods, while the value for the pressure sensor data changes its mean value.

The peaks in Figure 6.4 were likely caused by certain operating conditions, and some work was done to find which. One particular operating condition was found for low demands on engine torque, i.e. when just rolling, which can be seen in Figure 6.5.

For a high engine torque the residual reacted to a bias fault of the pressure sensor on the exhaust-side, but not to a leakage. For low engine torque the residual hardly reacted on the fault in the pressure sensor but gave peeks for the leakage. By designing two tests based on the same residual, but only execute them during specific conditions, there would be two tests that would be sensitive to two different faults instead of one test being sensitive to all.

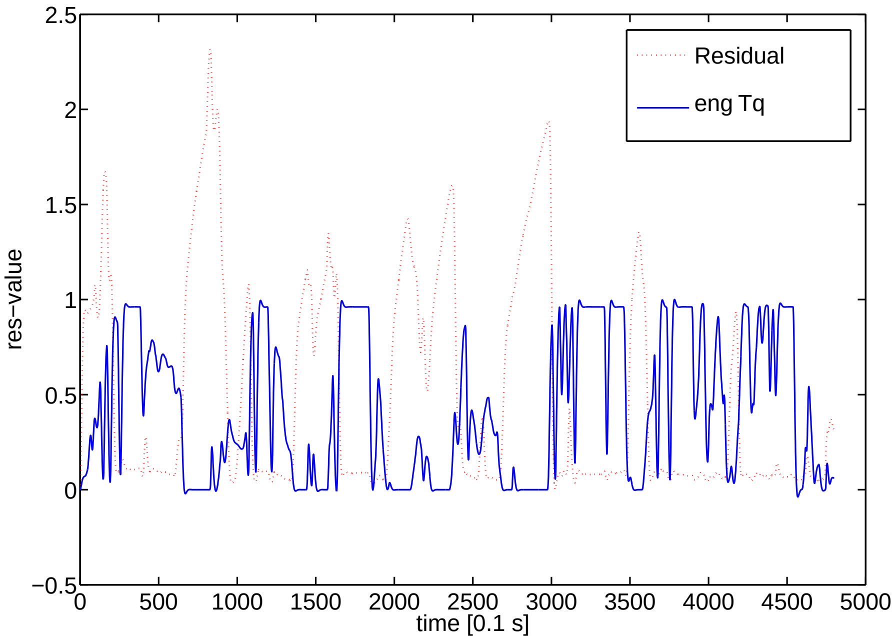


Figure 6.5. A plot indicating a connection between low engine torque (solid line) and high residual value (dashed line) for measurement data from a leakage on the exhaust-side.

A sliding mean processing for the residual when the engine torque was higher then a chosen percentage of its max value resulted in a small change for the leakage and a bigger change for the fault in the sensor. Applying the same thinking but the other way around, a CuSum for the residual when the engine torque was low would result in a significant value for the leakage, but a much smaller for the sensor error. See Figure 6.6 for an example of how the CuSum can be used to separate a fault in the exhaust pressure sensor and a leakage on the exhaust-side.

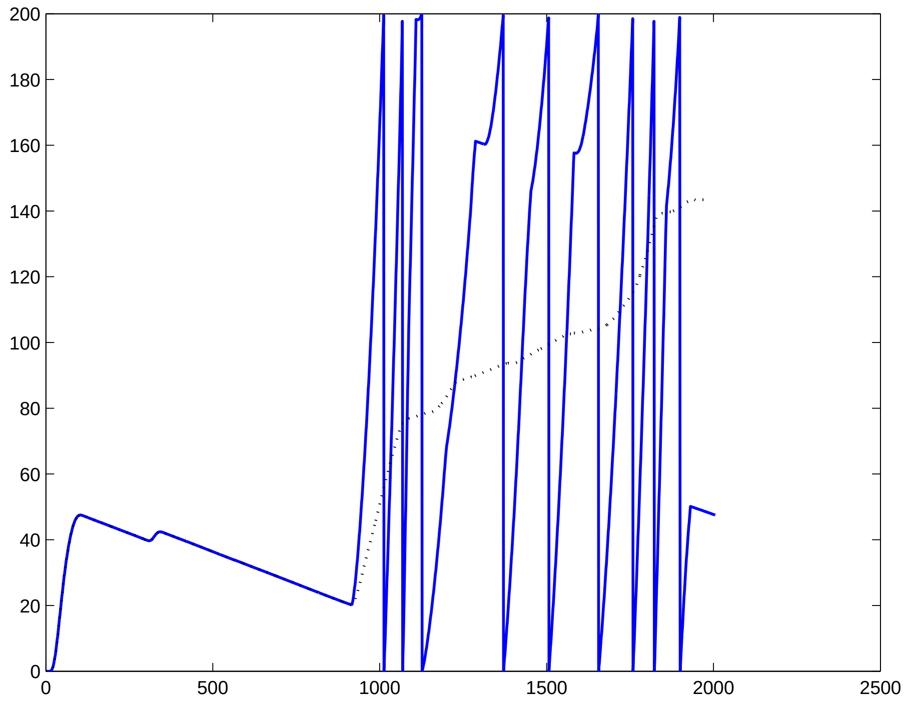


Figure 6.6. Graph showing the CuSum for a residual based on measurements for low engine torque. The curves are from leakage measurement data on the exhaust-side (solid line) and from a constant negative bias fault on the exhaust pressure sensor (dotted line).

Using these two tests, and a third test that used the knowledge of imaginary values of the residual for faults in the boost pressure sensor, a decision structure can be formed as:

	LE	LI	SPE	SPB
$T_3(resD\ LowTq)$	0	0	x	0
$T_4(resD\ HighTq)$	x	x	0	0
$T_5(resD\ Complex)$	0	0	0	x

6.5 Diagnosis System Performance

By using the residuals that were implemented, a hole with diameter 15 mm could be detected on the exhaust-side, and a hole of diameter 25 mm on the boost-side. A relatively common fault among the faults that might sometimes occur, namely a tube coming lose from its hold, should thus be easy to detect.

It is possible to detect smaller leakages, but to which extent is not clear. As can be seen in Figure 6.7, a hole with diameter 15 mm on the boost-side still causes a significant difference in the residual signal compared to the fault free case, which indicates that also this size of leakage can be detected. The hole with diameter 5 mm is very similar to the fault free case, indicating that this leakage is too small to be detected with this residual. Analogous on the exhaust-side, a hole of diameter 9 mm is likely to be detectable, while the leakage with diameter 3 mm is too small.

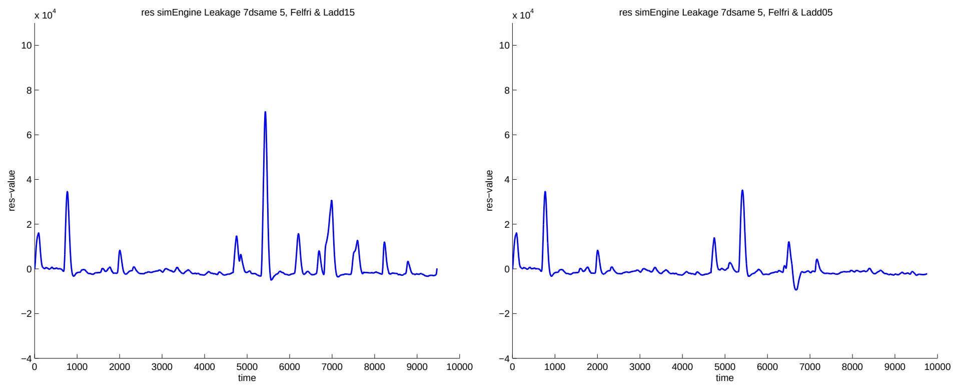


Figure 6.7. First half of the two plots is fault free measurement data, the second half is measurement data for a leakage on the boost-side. To the left, the leakage is from a hole of diameter 15 mm, to the right from a hole of diameter 5 mm. The plots indicates that the hole with diameter 15 can be detected, while the hole with diameter 5mm probably can not since it resembles the fault free case.

Using all MSO classes, each sensitive to a specific combination of faults, it was proven that all single faults could be detected and perfectly isolated from each other in theory. By using the MHS algorithm it was shown that a set of 12 residuals would be enough for perfect isolation of all 13 single faults. By using the influence structure for those MSO classes where a computation sequence was found, a more realistic theoretical isolability was obtained, see Figure 6.8. There are two single faults that most of the other single faults can not be isolated from. These faults are faults in the ambient pressure sensor, f_{spa}, and in the exhaust pressure sensor, f_{spe}. The ambient pressure sensor is considered one of the most robust sensors in the system, why this fault could be regarded as less likely in the event that a single fault needs to be pointed out.

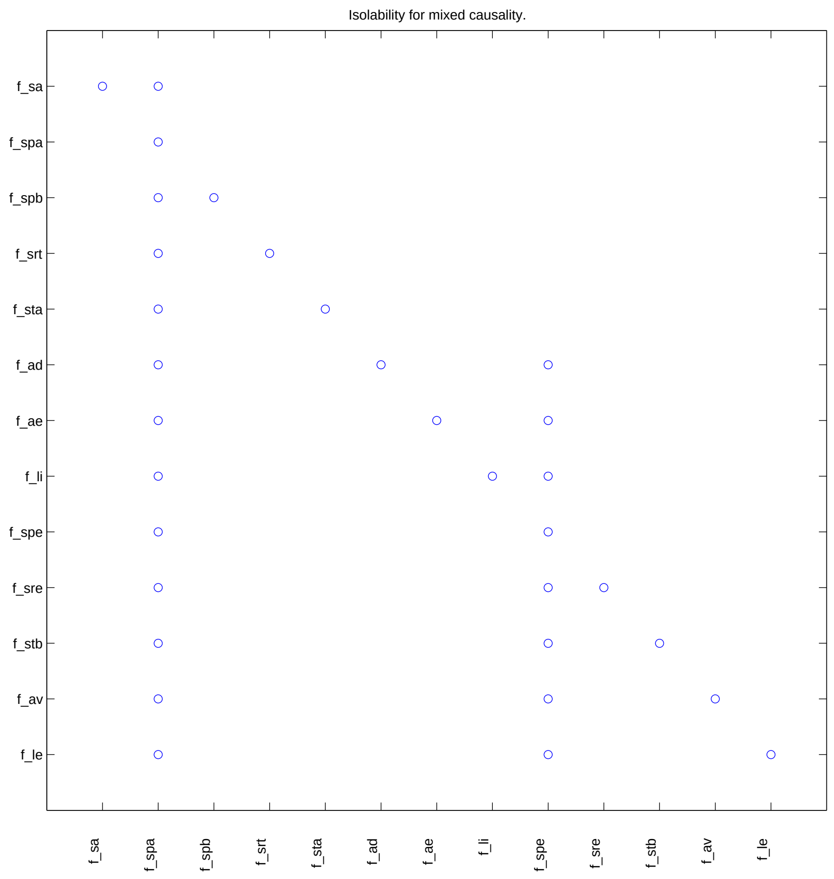


Figure 6.8. Theoretical fault isolation for mixed causality based on the residuals for which a computation sequence was found. A ring in the figure indicates that the fault on row i can not be isolated from the fault on column j. As it can be seen, most faults can not be isolated from faults in the ambient pressure sensor, f_{spa}, and faults in the exhaust pressure sensor, f_{spe}.

6.5.1 Actual Fault Sensitivity and Isolation

For leakages, and faults on the pressure sensors on the boost-side and the exhaust-side, faults could be implemented. However, other faults would be hard to implement and might even cause damage to the truck if active while driving. As a consequence, only measurements from faults in these pressure sensors and from the leakages were available when creating the decision structure.

Due to this lack of representative fault measurements, the only faults for which real detectability and isolability was investigated were a negative additive fault for the two pressure sensors and the largest leakage areas for the two regions.

With the two tests based on residual C and the three tests based on residual D, Section 6.4, the following decision structure was obtained:

	LE	LI	SPE	SPB
$T_1(resC\ up)$	0	x	0	x
$T_2(resC\ down)$	x	0	x	0
$T_3(resD\ LowTq)$	0	0	x	0
$T_4(resD\ HighTq)$	x	x	0	0
$T_5(resD\ Complex)$	0	0	0	x

The isolation structure generated from this decision structure is

	LE	LI	SPE	SPB
LE	1	0	0	0
LI	0	1	0	0
SPE	0	0	1	0
SPB	0	0	0	1

which shows that all these faults can be detected and isolated.

Since a leakage can be interpreted as a non constant bias on a presure sensor, these four faults are considered the hardest to isolate from each other. With the knowledge that this isolation is possible, the isolation structure for all the other single faults will probably be found to be good, after having investigated more fault behaviours.

Chapter 7

Analysis and Discussion

7.1 Measurements

The measurements were gathered on a test track, not in a test-bed. As a consequence, the measurements were generated in only remotely similar driving cycles compared to if they had been gathered in a test-bed. Due to this, it became harder to compare the behavior of residuals subject to different faults. However, there are two great advantages with the out-on-the-track measurements. First the availability. There are many more trucks than test-beds available which makes it easier to access equipment to generate measurements. Second the realistic aspect. Collecting measurements with a truck gives more realistic measurements, i.e. measurements more similar to actual driving conditions.

The hardware used for measurement acquisition, though mounted in the truck, could easily be dissembled and reassembled in a test-bed.

Inspite of extensive work, it proved hard to get relevant measurement data from faulty components. Some would be hard to implement, others might cause damage to the truck. For those that could be implemented, a infinity of fault sizes could be chosen. A selection of faults and fault sizes was therefore made. The selection was narrow, but a wider selection would have consumed more time than it would have improved the overall findings.

7.2 Evaluation of Method for Residual Generation

The method is thorough, generating all MSOs from the model and giving all possible residuals that can be generated from the MSOs with the available tools in the implementation. The result depends strongly upon the model used, why great care should be taken at the choice of model. The starting model and the resulting tests are things that the engineer needs to work with in order to form a good diagnosis system. The intermediate steps works more or less automatically with some minor special cases that needs to be taken into account. It will be hard to automate the selection process of test quantities since many special cases can be found, e.g. tests that proves unfavourable in one aspect might prove useful in one other and/or in combination with some other tests.

The implicit solutions found when solving equation system in order to find computation sequences, were discarded in this thesis. This was due to the fact that simple implementations of test quantities were desired and that there was no lack of found computation sequences. Implicit solutions can however be used in a computation sequences if numerical iteration, e.g. Newton-Raphson, is allowed in each time-step in the truck diagnosis system. This might however be too demanding for the computer on-board the truck since the solving time is non-deterministic for a specified accuracy. Investigation of worst case execution time is thus needed if implicit solutions are desired to be used.

Another solving tool than Maplesoft's MAPLE engine, which was in use in the MATLAB version available at the time of this thesis, might provide an alternative solution to the problem with complex equations connected to large SCCs, and possibly increase the speed of the process. According to Mathworks web page (acc. feb 2009), the MAPLE engine is replaced with a MuPAD ® engine from MATLAB v5.0 (R2008a+). This solving tool has not been tested in this thesis.

7.3 Evaluation of Diagnosis System Performance

It is possible to detect the two kinds of leakages. Theoretically all faults can be detected and isolated. In reality the model is inaccurate to that degree that many residuals that should not react to fault do in fact react to them and vice versa. That made the process of choosing residuals in order to isolate faults much harder. Since most of the examined residuals did not behave according to the influence structure, there was no way to tell what the actual isolability would be without actually creating faults and finding the decision structure empirically, and then the actual isolability. Exactly which faults that can be isolated in reality is therefore unclear, but leakages and pressure sensor faults on exhaust- and boost-side can be isolated from each other.

If smaller leakages are desired to be detectable, the threshold needs to be lowered. This would increase the risk of false detection, and in some cases worsen the isolability. One way to enhance the detection, but still keep the isolation, could be to combine tests in specific ways. Take for example one test that has a very low risk for false detection and a high detectability but low isolation, and another test that has high risk for false detection but good isolability. The second test is not desirable to use due to the high risk of false detection. But if it is only used when the first test has detected a fault, the problem can be avoided. One other way could be to have two different thresholds for the same test, a lower to detect faults, and a higher to improve the isolation. During the data acquisition no warnings or fault codes were produced, indicating that the current tests are not influenced by leakages. If the existing diagnosis system is combined with one or some of the found residuals in this thesis, the combined diagnosis system would be able to detect leakages too.

Though no investigation about the potential gain of new sensors was made, a method was found treating this subject. The sensor placement method presented in [19] provides the means to perform a theoretical analysis of the improvements of a diagnosis system if additional sensor(s) would be introduced in the modeled system. Since the method is based on structural information, it is well applicable to the complex non-linear model used in this thesis. Another method [10], used on linear differential systems, performes analytical analysis of additional sensor(s). The benefit of analytical analysis of sensor placement is that it might handle models where structural methods fail.

7.4 Real Time Analysis

The method used to detect leakages on a (slow) stationary PC can be performed in real time in the MATLAB environment. That is, the value of a residual can be calculated for a series of measurements much faster then the time it takes to sample it. The question is of course if the processor on-board the truck can perform the same diagnosis in real time, knowing that it is slower than the stationary PC. To answer that question some additional studies are needed, but the following reasoning suggests a positive answer to the question.

The generation of residuals and tests is extensive and rather time consuming but the actual computation of a residual for a given observation is fast in comparison. One residual with integral causality needs about 200 lines of sequential code that could easily be compressed and/or reduced. The computations are mostly elementary such as addition and multiplication. The integration are treated with Euler forward with a fixed step length. No equations need to be solved. All in all, it is just a set of variables that should be calculated from an already known set of variables.

Chapter 8

Concluding Words

8.1 Conclusions

The methods and model used in this thesis provide means to generate residuals capable of detecting leakages, and some other faults, with measurements from the present set of sensors and actuators. The leakages can be isolated from the investigated faults in both the boost and exhaust pressure sensor, and vice versa. Further measurements are nevertheless needed to create reliable thresholds and to determine exactly which faults that can be detected and isolated. The resulting tests can probably be executed with the computation power available in a truck and is in that case suitable for on-board diagnosis.

To recapitulate the results of this thesis against its purpose, see the following results conclusions:

- In this work, a general method for model-based residual generation has been used to perform gas leakage detection in a diesel engine.

- The diagnostic tests generated with the chosen method has been evaluated with real measurement data, showing that leakages of a certain size can not only be detected, but also isolated from the faults that that are considered hardest to isolate from.

- Hardware and equipments has been created to acquire measurement data from leakages and some other faults.

- Improvements have been made to the implementation of the method, which speeds up the working process.

- The improvement of the diagnosis performance, resulting from extra sensors is not investigated, but propositions are made to how this could be investigated if wanted.

- Leakage diagnosis based on the methods in this thesis is probably suitable for implementation and execution in real time in a truck.

8.2 Future Work

This work has been done without active stimulation of the system. If forcing the EGR-vault to close during diagnosis, a special model could be made where the gas recirculation through the EGR was excluded, which would simplify the model and most likely the detection and isolation. Short periods with the EGR-vault closed can be found in some operating conditions while longer sequences probably need active control of the EGR-vault. Examining the impact of active diagnosis is thus a possible field to explore.

Implementation of more kinds of fault behaviors and in more components would be highly interesting in order to get a more accurate view of the actual detection and isolation performance. It would also be interesting to try different fault sizes in each component.

Regarding the implementation of the method used to generate residuals, the following topics can be looked into. One rather simple improvement would be a C/C++ function, restarting Matlab automatically after each freeze, continuing the process from where it froze. Another solving tool could be tested to examine the improvements of the solving speed, the stability and the number of solutions. For a better overall speed the residual generation method could be implemented in an other programming language.

Bibliography

[1] D. Antory. Application of a data-driven moitoring technique to diagnose air leaks in an automotive diesel engine, a case study. Technical report, Electrical Test for Advanced Architectures, International Automotive Research Center, Warwick Manufacturing Group, November 2005.

[2] G. Arrhenius and H. Einarsson. Automatic design of diagnosis systems using consistency based residuals. Master's thesis, Uppsala University, 2004.

[3] G. Biswas, M-O. Cordier, J. Lunze, L. Travé-Massuyès, and M. Staroswiecki. Diagnosis of complex systems: Bridging the methodologies of the fdi and dx communities. *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics.*

[4] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control.* Springer Verlag Berlin, 2003.

[5] 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.

[6] M-O. Cordier, P. Dague, F. Lévy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. Conflicts versus analytical redundancy relations: A comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics.*

[7] S.X. Ding, M. Zhong, T. Jeinsch, and B. Tang. Lmi-based integration of robust h control and rfd for lti systems. *Proc. IFAC World Congr., Barcelona, Spain*, page 494499, 2002.

[8] L. Eriksson. *Vehicular Systems.* Vehicular systems, ISY, 2007.

[9] E. Frisk. *Residual Generation for Fault Diagnosis.* PhD thesis, Linköpings Universitet, November 2001.

[10] E. Frisk, M. Krysander, and J. Åslund. Sensor placement for fault isolation in linear differential-algebraic systems. *Automatica*, 45(2):364–371, 2009.

[11] 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.

[12] J. Gertler. *Analytical Redundancy Methods in Fault Detection and Isolation; Survey and Synthesis.* IFAC Fault detection, Supervision and Safety for Technical Processes, 1991.

[13] T. Glad and L. Ljung. *Reglerteknik, Grunläggande teori.* Studentlitteratur, 4:3 edition, 2006. ISBN 978-91-44-02275-8.

[14] F. Gustafsson. *Adaptive Filtering and Change Detection.* John Wiley & Sons, 2000.

[15] J.B. Heywood. *Internal Combustion Engine Fundamentals.* McFraw-Hill, 1988.

[16] B. Jiang, M. Staroswiecki, and V. Cocquempot. Fault accommodation for nonlinear dynamic systems. *IEEE Transactions on Automatic Control.*

[17] J. Kingstedt and M. Johansson. Methods for residual generation using mixed causality in model based diagnosis. Master's thesis, Linköpings Tekniska Högskola, 2008. LiTH-ISY-EX–08/4882–SE.

[18] 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), 2008.

[19] 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.

[20] H. Niemann and J. Stoustrup. Integration of control and fault detection: nominal and robust design. *Proc. IFAC Safeprocess, Budapest, Hungary*, page 341346, 1997.

[21] M. Nyberg. Model based diagnosis of both sensor-faults and leakage in the air-intake system of an si-engine. Technical report, Vehicular systems Department of Electrical Engineering, 1999. SP-1419.

[22] M. Nyberg and E. Frisk. *Model Based Diagnosis of Technical Processes.* Bokakademin AB, 2008.

[23] Y. Peng and J. Reggia. *Abductive Inference Models for Diagnostic Problem Solving.* Springer-Verlag, New York, 1990.

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

[25] S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Academic, New York, 2 edition, 2003.

[26] J. Wahlström and L. Eriksson. Modeling of a diesel engine with vgt and egr including oxygen mass fraction. Technical report, Vehicular Systems Department of Electrical Engineering, September 2006. LiTH-ISY-R-2747.

[27] N. Weinhold, S.X. Ding, T. Jeinsch, and M. Schultalbers. Embedded model-based fault diagnosis for on-board diagnosis of engine control systems. In *2005 IEEE Conference on Control Applications*, pages 1206–1211, Toronto, Canada, August 2005.

Appendix A

Summary of Model Equations

Manifolds

$$\frac{d}{dt}p_{im} = \frac{R_a T_{im}}{V_{im}}\left(W_c + W_{egr} - W_{ei} - W_{li}\right) \tag{A.1}$$

$$\frac{d}{dt}p_{em} = \frac{R_e T_{em}}{V_{em}}\left(W_{eo} - W_t - W_{egr} - W_{le}\right) \tag{A.2}$$

Leakages

Leakage intake

$$W_{li} = \frac{A_{li,eff}\, p_{im} \Psi_{li}}{\sqrt{T_{im} R_a}} \tag{A.3}$$

$$\Psi_{li} = \sqrt{\frac{2\gamma_a}{\gamma_a - 1}\left(\Pi_{li}^{2/\gamma_a} - \Pi_{li}^{1+1/\gamma_a}\right)} \tag{A.4}$$

$$\Pi_{li} = \begin{cases} \Pi_{li,opt} & \text{if} \quad \dfrac{p_{amb}}{p_{im}} < \Pi_{li,opt} \\[2ex] \dfrac{p_{amb}}{p_{im}} & \text{if} \quad \Pi_{li,opt} \leqslant \dfrac{p_{amb}}{p_{im}} \leqslant 1 \\[2ex] 1 & \text{if} \quad 1 < \dfrac{p_{amb}}{p_{im}} \end{cases} \tag{A.5}$$

Leakage exhaust

$$W_{le} = \frac{A_{le,eff}\, p_{em} \Psi_{le}}{\sqrt{T_{em} R_e}} \tag{A.6}$$

$$\Psi_{le} = 1 - \left(\frac{1 - \Pi_{le}}{1 - \Pi_{le,opt}} - 1\right)^2 \tag{A.7}$$

$$\Pi_{le} = \begin{cases} \Pi_{le,opt} & \text{if } \dfrac{p_{amb}}{p_{em}} < \Pi_{le,opt} \\ \dfrac{p_{amb}}{p_{em}} & \text{if } \Pi_{le,opt} \leqslant \dfrac{p_{amb}}{p_{em}} \leqslant 1 \\ 1 & \text{if } 1 < \dfrac{p_{amb}}{p_{em}} \end{cases} \tag{A.8}$$

Cylinder flow

$$W_{ei} = \frac{\eta_{vol} p_{im} n_e V_d}{120 R_a T_{im}} \tag{A.9}$$

$$\eta_{vol} = c_{vol1} \sqrt{p_{im}} + c_{vol2} \sqrt{n_e} + c_{vol3} \tag{A.10}$$

$$W_f = \frac{10^{-6}}{120} u_\delta n_e n_{cyl} \tag{A.11}$$

$$W_{eo} = W_f + W_{ei} \tag{A.12}$$

Cylinder out temperature

$$\Pi_e = \frac{p_{em}}{p_{im}} \tag{A.13}$$

$$q_{in} = \frac{W_f q_{HV}}{W_{ei} + W_f}(1 - x_r) \tag{A.14}$$

$$x_p = 1 + \frac{q_{in} x_{cv}}{c_{va} T_1 r_c^{\gamma_a - 1}} \tag{A.15}$$

$$T_e = \eta_{sc} \Pi_e^{1-1/\gamma_a} r_c^{1-\gamma_a} x_p^{1/\gamma_a - 1} \left(q_{in} \left(\frac{1 - x_{cv}}{c_{pa}} + \frac{x_{cv}}{c_{va}} \right) + T_1 r_c^{\gamma_a - 1} \right) \tag{A.16}$$

$$\dot{T}_1 = \frac{x_r T_e + (1 - x_r) T_{im} - T_1}{\Delta t} \tag{A.17}$$

$$T_{em} = T_w + (T_e - T_w) e^{-\frac{h_{tot} \pi d_{pipe} l_{pipe} n_{pipe}}{W_{eo} c_{pe}}} \tag{A.18}$$

Sensors

Measuring states

$$y_{\omega_t} = \omega_t \tag{A.19}$$

$$y_{W_c} = W_c \tag{A.20}$$

$$y_{pim} = p_{im} \tag{A.21}$$

$$y_{pem} = p_{em} \tag{A.22}$$

Measurements as inputs

$$y_{tamb} = T_{amb} \tag{A.23}$$

$$y_{pamb} = p_{amb} \tag{A.24}$$

$$y_{tim} = T_{im} \tag{A.25}$$

$$y_{ne} = n_e \tag{A.26}$$

EGR-valve

$$W_{egr} = \frac{A_{egr} p_{em} \Psi_{egr}}{\sqrt{T_{em} R_e}} \tag{A.27}$$

$$\Psi_{egr} = 1 - \left(\frac{1 - \Pi_{egr}}{1 - \Pi_{egr,opt}} - 1 \right)^2 \tag{A.28}$$

$$\Pi_{egr} = \begin{cases} \Pi_{egr,opt} & \text{if} \quad \dfrac{p_{im}}{p_{em}} < \Pi_{egr,opt} \\[2ex] \dfrac{p_{im}}{p_{em}} & \text{if} \quad \Pi_{egr,opt} \leqslant \dfrac{p_{im}}{p_{em}} \leqslant 1 \\[2ex] 1 & \text{if} \quad 1 < \dfrac{p_{im}}{p_{em}} \end{cases} \tag{A.29}$$

$$A_{egr} = A_{egr,max} f_{egr}(u_{egr}) \tag{A.30}$$

$$f_{egr}(u_{egr}) = \begin{cases} c_{egr1} u_{egr}^2 + c_{egr2} u_{egr} + c_{egr3} & \text{if} \quad u_{egr} \leqslant -\dfrac{c_{egr2}}{2c_{egr1}} \\[2ex] c_{egr3} - \dfrac{c_{egr2}^2}{4c_{egr1}} & \text{if} \quad u_{egr} > -\dfrac{c_{egr2}}{2c_{egr1}} \end{cases} \tag{A.31}$$

Turbo

Turbo inertia

$$\frac{d}{dt}\omega_t = \frac{P_t\eta_m - P_c}{J_t\omega_t} \tag{A.32}$$

Turbine efficiency

$$\Pi_t = \frac{p_{amb}}{p_{em}} \tag{A.33}$$

$$P_t\eta_m = \eta_{tm} W_t c_{pe} T_{em}\left(1 - \Pi_t^{1-1/\gamma_e}\right) \tag{A.34}$$

$$\eta_{tm} = \eta_{tm,max} - c_m(BSR - BSR_{opt})^2 \tag{A.35}$$

$$BSR = \frac{R_t\omega_t}{\sqrt{2c_{pe}T_{em}\left(1 - \Pi_t^{1-1/\gamma_e}\right)}} \tag{A.36}$$

$$c_m = c_{m1}(\omega_t - c_{m2})^{c_{m3}} \tag{A.37}$$

Turbin mass flow

$$W_t = \frac{A_{vgt,max}p_{em}f_{\Pi_t}(\Pi_t)f_{vgt}(u_{vgt})}{\sqrt{T_{em}}} \tag{A.38}$$

$$f_{\Pi_t}(\Pi_t) = \sqrt{1 - \Pi_t^{K_t}} \tag{A.39}$$

$$f_{vgt}(u_{vgt}) = c_{f2} + c_{f1}\sqrt{1 - \left(\frac{u_{vgt} - c_{vgt2}}{c_{vgt1}}\right)^2} \tag{A.40}$$

Compressor

Compressor efficiency

$$\Pi_c = \frac{p_{im}}{p_{amb}} \tag{A.41}$$

$$P_c = \frac{W_c c_{pa} T_{amb}}{\eta_c}\left(\Pi_c^{1-1/\gamma_a} - 1\right) \tag{A.42}$$

$$\eta_c = \eta_{c,max} - \chi^T Q_c \chi \tag{A.43}$$

$$\chi = \begin{bmatrix} W_c - W_{c,opt} \\ \\ \pi_c - \pi_{c,opt} \end{bmatrix} \tag{A.44}$$

$$\pi_c = (\Pi_c - 1)^{pow_\pi} \tag{A.45}$$

$$Q_c = \begin{bmatrix} a_1 & a_3 \\ a_3 & a_2 \end{bmatrix} \tag{A.46}$$

Compressor mass flow

$$W_c = \frac{p_{amb}\pi R_c^3 \omega_t}{R_a T_{amb}} \Phi_c \tag{A.47}$$

$$\Phi_c = \sqrt{\frac{1 - c_{\Psi 1}(\Psi_c - c_{\Psi 2})^2}{c_{\Phi 1}}} + c_{\Phi 2} \tag{A.48}$$

$$\Psi_c = \frac{2c_{pa}T_{amb}\left(\Pi_c^{1-1/\gamma_a} - 1\right)}{R_c^2 \omega_t^2} \tag{A.49}$$

$$c_{\Psi 1} = c_{\omega \Psi 1}\omega_t^2 + c_{\omega \Psi 2}\omega_t + c_{\omega \Psi 3} \tag{A.50}$$

$$c_{\Phi 1} = c_{\omega \Phi 1}\omega_t^2 + c_{\omega \Phi 2}\omega_t + c_{\omega \Phi 3} \tag{A.51}$$

Appendix B

Example of a Residual

$function[residual, BM] = resEngineModelArea1(ustruct, BM)$

```
    % Variables
y_omegat=u_struct.y_omegat;
y_wc=u_struct.y_wc;
y_pim=u_struct.y_pim;
y_pem=u_struct.y_pem;
y_tamb=u_struct.y_tamb;
y_pamb=u_struct.y_pamb;
y_tim=u_struct.y_tim;
y_ne=u_struct.y_ne;
act_udelta=u_struct.act_udelta;
act_uegr=u_struct.act_uegr;
act_uvgt=u_struct.act_uvgt;

    % Parameters used in the model (actual values replaced with 'x')
R_A=x;
V_IM=x;
R_E=x;
V_EM=x;
GAMMA_A=x;
PI_LEOPT=x;
V_D=x;
C_VOL1=x;
C_VOL2=x;
C_VOL3=x;
N_CYL=x;
Q_HV=x;
X_R=x;
X_CV=x;
```

```
C_VA=x;
R_C=x;
NY_SC=x;
C_PA=x;
TSTATE=x;
C_PE=x;
H_TOT=x;
PI=x;
D_PIPE=x;
L_PIPE=x;
N_PIPE=x;
PI_EGROPT=x;
A_EGRMAX=x;
C_EGR1=x;
C_EGR2=x;
C_EGR3=x;
J_T=x;
GAMMA_E=x;
NY_TMMAX=x:
BSROPT=x;
C_M1=x;
C_M2=x;
C_M3=x;
A_VGTMAX=x;
K_T=x;
C_VGT1=x;
C_VGT2=x;
NY_CMAX=x;
Q1=x;
Q2=x;
Q3=x;
W_COPT=x;
POWPI=x;
PI_COPT=x;
C_PSI2=x;
C_PHI2=x;
C_WPSI1=x;
C_WPSI2=x;
C_WPSI3=x;
C_WPHI1=x;
C_WPHI2=x;
C_WPHI3=x;
C_F1=x;
C_F2=x;
T_W=x;
R_T=x;
```

```
    % Computations sequence
t_amb = y_tamb;
omega_t = y_omegat;
c_phi1 = C_WPHI1*omega_t^2+C_WPHI2*omega_t+C_WPHI3;
c_psi1 = C_WPSI1*omega_t^2+C_WPSI2*omega_t+C_WPSI3;
p_amb = y_pamb;
p_im = y_pim;
pi_c = p_im/p_amb;
psi_c = 2*C_PA*t_amb*(pi_c^((GAMMA_A-1)/GAMMA_A)-1)/R_C^2 ...
/omega_t^2;
phi_c = (-(-1+c_psi1*psi_c^2-2*c_psi1*psi_c*C_PSI2+c_psi1*C_PSI2^2) ...
/c_phi1)^(1/2)+C_PHI2;
w_c = p_amb*PI*R_C^3*omega_t*phi_c/R_A/t_amb;

    % Residual equation
residual=w_c-y_wc;
```

Appendix C

Abbreviations

Abbreviation	Explanation
ARR	Analytical Reduncancy Relations, p.16
CAN	Controler Area Network, p.40
CS	Computation Sequence, p.21
DX	Diagnostic, p.10
EDAQ	Electronic Data Acquisition, p.40
EGR	Exhaust Gas Recirculation, p.2
FDI	Fault Detection and Isolation, p.10
IC	InterCooler, p.1
MHS	Minimal Hitting-Set, p.18
MSO	Minimal Structurally Overdetermined, p.21
SCC	Strongly Connected Component, p.22
SM	Structural Model, p.19
VGT	Variable Geometry Turbine, p.2

Table C.1. Abbreviations and their full names.