Model-based fault diagnosis applied to an SI-Engine

Examensarbete utfört i Fordonssystem
vid Tekniska Högskolan i Linköping
av

Erik Frisk

Reg nr: LiTH-ISY-EX-1679

Model-based fault diagnosis applied to an SI-Engine

Examensarbete utfört i Fordonssystem
vid Tekniska Högskolan i Linköping
av

Erik Frisk

Reg nr: LiTH-ISY-EX-1679

Supervisor: **Mattias Nyberg**
Lars Nielsen

Examiner: **Lars Nielsen**

Linköping, September 29, 1996.

Abstract

A diagnosis procedure is an algorithm to detect and locate (isolate) faulty components in a dynamic process. In 1994 the California Air Resource Board released a regulation, called OBD II, demanding a thorough diagnosis system on board automotive vehicles. These legislative demands indicate that diagnosis will become increasingly important for automotive engines in the next few years.

To achieve diagnosis, redundancy has to be included in the system. This redundancy can be either hardware redundancy or analytical redundancy. Hardware redundancy, e.g. an extra sensor or extra actuator, can be space consuming or expensive. Methods based on analytical redundancy need no extra hardware, the redundancy here is generated from a process model instead. In this thesis, approaches utilizing analytical redundancy is examined.

A literature study is made, surveying a number of approaches to the diagnosis problem. Three approaches, based on both linear and non-linear models, are selected and further analyzed and complete design examples are performed. A mathematical model of an SI-engine is derived to enable simulations of the designed methods.

Key Words: Diagnosis, Analytical redundancy, SI-Engine, FDI, Eigenstructure, Parity equations, Robustness.

Acknowledgments

I wish to thank my supervisors Mattias Nyberg and Lars Nielsen. Thanks to Mattias for figure 6.1 and for the neverending stream of thoughts, comments and ideas.

Special thanks to Andrej Perkovic and Patrik Bergren for appendix A, Lars Eriksson for many useful comments, Victoria Wibeck for the loan of her computer, Greg and Tori for keeping me awake.

Notation

Abbreviations

AFD	Actuator Fault Diagnosis
CFD	Component Fault Diagnosis
DOS	Dedicated Observer Scheme
EGO	Exhaust Gas Oxygen
EGR	Ehaust Gas recirculation
FDI	Fault Detection and Isolation
GLR	Generalized Likelihood Ratio
GOS	General Observer Scheme
IFD	Instrumental(sensor) Fault Diagnosis
SI	Spark-Ignition
UIO	Unknown Input Observer

Contents

Chapter 1

Introduction

Diagnosis is a procedure to detect and locate faulty components in a dynamic process, e.g. an automotive engine. Why is there a need for diagnosis? The answer depends on the application. Some areas where diagnosis schemes can be of importance are:

- Chemical plants
- Nuclear plants
- Aero planes
- Automotive engines

The reason for diagnosing faults in the first three areas are that even *small* malfunctions can have disastrous, life threatening consequences. Here it is quite natural to want to detect and isolate a faulty component early before it can lead to plant failure.

This report concerns the last area and one of the main goals of a diagnosis scheme of an automotive engine is, apart from detecting life threatening failures, diagnosing faults in e.g. the emission control systems leading to greater volumes of pollutants in the outlet, [5].

1.1 Automotive engine diagnosis

In 1990 the American agency EPA (U.S Environmental Protection Agency) estimated that 60% of the total HC (Hydro Carbon) pollutants originated from 20% of the vehicles with malfunctions in their emission control systems, see [45]. This shows that a diagnostic procedure on board vehicles would probably be a major part of a solution to reduce vehicle emissions.

In 1988 CARB (California Air Resource Board) proposed OBD [1] I, a regulation stating that vehicles had to monitor the on-board computer, computer sensed components, the fuel metering system and the *exhaust gas recirculation* (EGR)[2]. In 1994 CARB released a new regulation, called OBD II, demanding an even more thorough monitor system.

[1] On Board Diagnostic

[2] More about these terms in section 6.1

The OBD II regulation states that a dashboard light MIL(Malfunction Indicator Light) should warn the driver when a fault has occured that causes pollutant emissions to exceed legislated limits by more than 1.5 times. It also states that a DTC (Diagnostic Trouble Code) is to be stored in the on-board computer to simplify repair. For in depth description of the regulation see [3].

There has also emerged a federal regulation similar to OBD II from the EPA. This indicates that a similar regulation for the European market soon will emerge.

Besides the legislative demands on automotive vehicles there are other factors indicating the importance of diagnosis. Examples of advantages with a well functioning diagnosis scheme are

- Repairability
  Repair can be simplified.
- Availability
  A diagnosis scheme can be able to determine the severity of a fault and determine when it is possible to drive the automobile to the workshop, this is called limp-home.
- Safety
  The personal safety in the vehicle is improved.
- Vehicle protection
  If faulty components are detected in an early stage, damage to the vehicle can be avoided.

Due to the legislative demands, it can be hard to incorporate new technology in the engine. A systematic design procedure reduces the effort to redesign the diagnosis scheme to be able to include the new component. A general and systematic diagnosis design procedure thus enhances the possibility to incorporate new technology.

Today the diagnosis tasks performed require great amount of effort. It is estimated that, today, approximately 40% of the entire software in the control unit are diagnosis related, [22].

1.2 Objectives

The objectives with this thesis work is to

1. Survey existing methods of model based fault diagnosis of dynamic systems that has a potential in automotive SI engines.
2. Develop a realistic model of an SI engine in a simulation tool. The model must contain features suitable for experiments with fault diagnosis.
3. Develop a diagnosis scheme by selecting methods and ideas from literature and apply them, in combination with own ideas, to the model.
4. In the simulation environment, evaluate the diagnosis scheme chosen in respect to robustness and other aspects.

1.3 Readers guide

In chapter 2 the diagnosis problem is defined. A survey over some approaches found in literature is done in chapter 3, methods to be further analyzed are also chosen. Two of these are described in detail in chapter 4 and 5. Chapter 5 is the mathematically most complex chapter in this report. For reader convenience appendix C is included where a few of the mathematical concepts used are defined.

In chapter 6, engine fundamentals are discussed and a physical model of the engine is derived. This model is later used in chapter 7 where diagnosis on the model is simulated. Simulink implementations of the model is found in appendix B. These are included for completeness and no explanation on how they work are submitted. For a complete description of Simulink, see [28].

Implementations of different methods are done in Matlab, especially in chapter 4 and 5. See [27] for detailed description of Matlab syntax.

Appendix A describes the laboratory facility used.

Chapter 2

The Fault diagnosis problem

In this chapter we will define the diagnosis problem and discuss why a *model* based approach is necessary for high performance diagnosis.

2.1 Problem formulation

A general diagnosis procedure for a dynamic system consists of several tasks. The following steps are suggested in literature, e.g. in [35].

- **Fault detection**
  Detect when a fault has occured. Special emphasis is laid upon *incipient*, or developing, faults rather than large step faults. This because incipient faults are harder to detect.

- **Fault isolation**
  Isolate the fault. Primarily to determine the faults origin but also the fault-type, size and time.

These two tasks are commonly referred to as FDI, Fault Detection and Isolation. FDI is sometimes referred to as diagnosis and the other way around and from now on in this report diagnosis is equivalent to FDI.

The system to be diagnosed often include a control loop which further complicates the problem. A control loop tend to hide or mask a faulty component or sensor making it even more important, in a controlled system, to detect incipient faults.

An important parameter in a diagnosis system is the false alarm rate, i.e. how often the system signals a fault in a fault-free environment, and probability for missed fault detection.

We speak about *faults* and *failures* in diagnosis. What do we mean by these words? In diagnosis literature there is a distinction between the two and the definition used can be written as in [35]:

Definition 2.1. *A failure suggests a complete breakdown of a process component while a fault is thought of as an unexpected component change that might be serious or tolerable.*

Obvious fault sources are actuators and sensors where the fault can be a bias or a drift. Other examples are actuator stuck-at faults. These are the types of faults that will be handled in this report.

In this paper we will investigate *model based* diagnosis, i.e. a diagnosis procedure that is founded on a model of the system to be diagnosed. Fault diagnosis and fault detection is not a new problem and before model based fault diagnosis, diagnosis were accomplished e.g. by introducing *hardware redundancy* in the process. A critical process component was then duplicated, triplicated (TMR [1]) or even quadrupled and then using a majority decision rule.

Hardware redundancy methods are fast and easy to implement but they have several drawbacks

- Extra hardware can be very expensive
- Introduces more complexity in the system
- The extra hardware is space consuming which can be of great importance, e.g. in a space shuttle. Also the components weight sometimes has to be considered.

Instead of using hardware redundancy, *analytical redundancy* can be utilized to reduce, or even avoid, the need for hardware redundancy. All methods examined in this report are founded on analytical redundancy. Analytical redundancy is in principle the relationships that exists between process variables and measured outputs. If an output is measured there are information about all variables that influences that variable in the measurement. If the relationships are known, by quantitative or qualitative knowledge, this information can be extracted and information extracted from different measurements can be checked for consistency against each other.

There are different types of analytical redundancy. If instead of measuring several outputs we feed the diagnosis procedure with output measurements at different times. If the system dynamics are known, we can from this time series extract fault information. This kind of analytical redundancy is called *temporal redundancy.*

One area where analytical redundancy based diagnosis will have problems replacing hardware redundancy is where the demands on fast response is very high, e.g. in an aircraft where human life depends on extremely fast response to component failure.

When the system model is given as analytical functions, analytical redundancy is sometimes referred to as *functional redundancy.* Even model based diagnosis is sometimes used synonymous with analytical redundancy, the correct relationship is however that a model based diagnosis scheme *utilizes* analytical redundancy.

2.2 Why model based diagnosis?

Why is there a need for a mathematical model to achieve diagnosis? It is easy to imagine a scheme where important entities of the dynamic process is measured and tested against predefined limits. The model based approach instead performs consistency checks of the

[1] Triple Modular Redundancy, see [12] for more information

process against a model of the process. There are several important advantages with the model based approach

1. Outputs are compared to their expected value on the basis of process state, therefore the thresholds can be set much tighter and the probability to identify faults in an early stage is increased dramatically.
2. A single fault in the process often propagate to several outputs and therefore causes more than one limit check to fire. This makes it hard to isolate faults without a mathematical model.
3. With a mathematical model of the process the FDI scheme can be made insensitive to unmeasured disturbances, e.g. in an SI-engine the load torque, making the FDI-scheme feasible in a much wider operating range.

There is of course a price to pay for these advantages in increased complexity in the diagnosis scheme and a need for a mathematical model.

Chapter 3

Approaches in literature

In this chapter we will look into the different approaches described in literature and briefly describe them. They will also be compared to each other and finally the approaches to be further investigated in this work will be selected.

The faults acting upon a system can be divided into three types of faults.

1. *Sensor (Instrument)* faults
   Faults acting on the sensors

2. *Actuator* faults
   Faults acting on the actuators

3. *Component (System)* faults
   A fault acting upon the system or the process we wish to diagnose.

A general FDI scheme based on analytical redundancy can be illustrated as in figure 3.1, an algorithm with measurements and control signals as inputs and a fault decision as output. If the system to be diagnosed is very large it can be necessary to include an inference mechanism to complement the isolation decision that very well can be an AI inference mechanism.

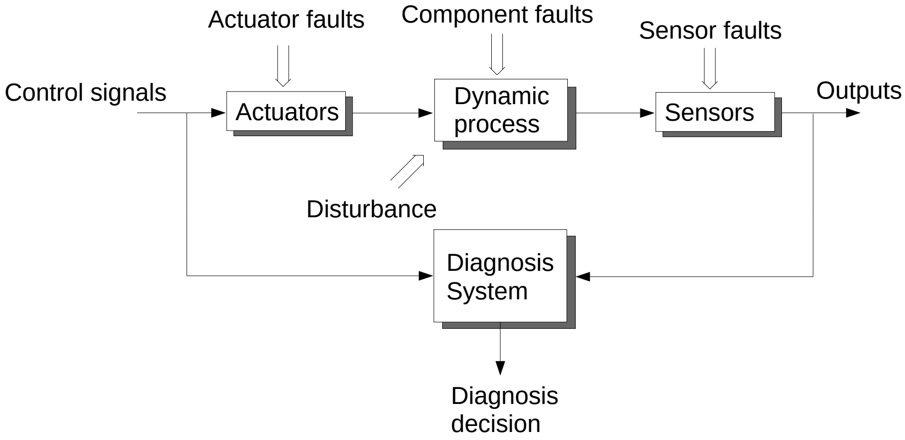

Figure 3.1. Structure of a diagnosis system

It is unrealistic to assume that all signals acting upon the process can be measured, therefore an important property of an algorithm is how it reacts upon these *unknown inputs*. It is also unrealistic to assume a perfect model, the modelling errors can be seen as unknown inputs. An algorithm that continue to work satisfactory even when unknown inputs vary is called *robust*. In some of the approaches described later in this chapter we have a possibility to achieve disturbance decoupling, i.e. make the isolation decision independent of unmeasured disturbances. Further discussions around robustness issues can be found in section 3.2.

There are many ways to categorize the different diagnosis schemes described in literature, but here we divide them into two groups: *knowledge based*, emerging from the computer science field of studies, and approaches based on *systems & control engineering*. The approaches based on systems & control engineering will, in the rest of the report, be shortened to control approaches. In this report we will concentrate on control engineering based approaches and therefore the discussions around knowledge based approaches are somewhat brief. This choice should only be seen as a way of limiting the scope of this work and not as knowledge based approaches are less important. More in-depth information about knowledge based approaches can be found in [44]. Approaches in both groups does however utilize analytical redundancy as was described in section 2.1.

3.1 Knowledge based approaches to FDI

This section gives a short introduction to the knowledge based approach to FDI. Here the word knowledge means that the knowledge known about the process and the faults acting upon the process is represented in a *knowledge base*. There is no need for the knowledge to be supported by analytical functions, the knowledge can be knowledge gathered by the engineers working with the process. The representation of the knowledge is an important issue here and is discussed in AI literature, e.g. in [39].

Knowledge based approaches is divided into *shallow diagnostic reasoning techniques* and *deep diagnostic reasoning techniques*.

3.1.1 Shallow diagnostic reasoning techniques

These approaches originates from applications where exact information about the process is hard to extract, e.g. in medical applications.

The most common way to implement a shallow reasoning diagnostic technique, is to use look-up tables, or a database, of process condition versus faults. This approach indicates that the look-up table becomes very large for even a moderately complex process where there is very little chance of identifying *all* faults and its corresponding system state. Therefore these approaches is not further investigated in this work, they are nevertheless interesting in a general perspective, e.g. because of their ability to incorporate knowledge not necessarily explainable. Also diagnosis schemes based on expert systems fits in this category.

3.1.2 Deep diagnostic reasoning techniques

The foundation of these techniques is a deeper model of the process than the look-up tables used in shallow knowledge based approaches.

There exists many different approaches to achieve diagnosis within the deep reasoning concept, two of these methods are

- Constraint suspension technique
- Governing equations technique

Constraint suspension technique

The constraint suspension technique uses constraints determined for all important entities in the process to be diagnosed. All entities, which if connected together forms the model of the process, has rules or constraints determining the relationships between in-out variables.

The main idea of the approach can be described as if the measured outputs of the sensors is consistent with the predicted value, a fault-free state is assumed. If there exists inconsistencies a fault is assumed present and a list of possible fault sources, i.e. entities, is determined by backtracking from the inconsistent output block and follow the dependency chain backwards. The possible fault sources is also called *candidates*. An example is given below:

If, in figure 3.2, output y_1 is inconsistent and y_2 is consistent with model prediction the candidate list is {1,2,4}. When the fault candidate list is determined each candidate, one at a time, is suspended, i.e. the model is assumed unknown and if there exist an output value of the candidate that explains all inconsistencies in the system that candidate is assumed as the source of the fault. A candidate can in itself be a set, i.e. it is possible that one fault alone cannot explain all inconsistencies in the system. If several sets of candidates can explain the inconsistencies the smallest set, i.e. the minimal set, is the most probable.

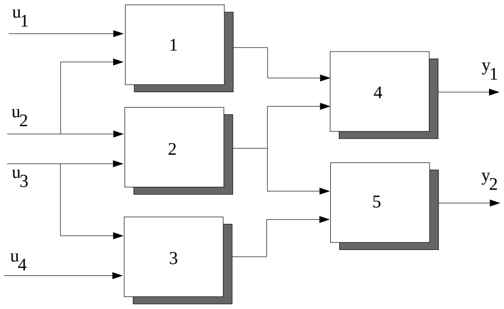

Figure 3.2.

Governing equations technique

This technique was primarily developed for chemical processes but is applicable if your model allows you to state equations describing constraints in the process and logical equations describing inconsistencies. If for example $F_{in} - F_{out} = 0$ describes static flow through a system. If the left hand side of the equation < 0, i.e. more flow out than in we can infer that

$$(F_{in}\text{-sensor too low}) \vee (F_{out}\text{-sensor too high})$$

If the left hand side of the constraint is > 0 we can infer that

$$(F_{in}\text{-sensor too high}) \vee (F_{out}\text{-sensor too low}) \vee (\text{system leak})$$

When defining a number of constraints as above we get a number of logical equations from whom it is possible to infer the fault origin. This can be done by a boolean logic inference system but to achieve a feasible system it is probably necessary to use a non-discrete inference system. A drawback with this approach is that it can be difficult to know when enough knowledge has been stated as logic formulas to diagnose any faults.

3.2 Systems & control engineering approaches to diagnosis

In control based approaches the diagnosis procedure is explicitly parted into two stages, the residual *generation* stage and the residual *evaluation* stage, as illustrated in figure 3.3. The residual evaluation can in its simplest form be a thresholding test on the residual, i.e. a test if abs($r(t)$) $> Threshold$. More generally the residual evaluation stage consists of a *change detection* test and a *logic inference system* to decide what caused the change. A change here represents a change in normal behavior of the residual.

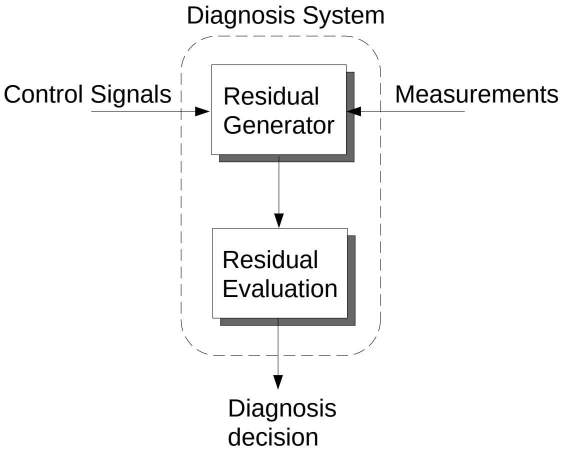

Figure 3.3. Two stage diagnosis system

The residual generation approaches can be divided into three subgroups, *limit & trend checking*, *signal analysis* and *process model based.*

- **Limit & trend checking**
  This approach is the simplest imaginable, testing sensor outputs against predefined limits and/or trends. This approach needs no mathematical model and are therefore simple to use but it is hard to achieve high performance diagnosis as was noted in section 2.2.

- **Signal analysis**
  These approaches analyses signals, i.e. sensor outputs, to achieve diagnosis. The analysis can be made in the frequency domain, [30], or by using a *signal* model, e.g. an ARMA-model. If fault influence are known to be greater than the input influence in well known frequency bands, a time-frequency distribution method as in [31] can be used.

- **Process model based residual generation**
  These methods are based on a *process* model and will be further investigated in this report. The process model based approaches are further parted into two groups, *parameter estimation*, and *parity space approaches.* These methods will be investigated further later in this section.

Before we can discuss the methods in this section we need to make some definitions. The approaches to be discussed here generates *residuals* which can be defined as

Definition 3.1 ***[Residual].*** *A residual (or parity vector)* $r(t)$ *is a scalar or vector that is* 0 *or small in the fault free case and* $\neq 0$ *when a fault occurs.*

The residual is a vector in the *parity space.* This definition implies that a residual $r(t)$ has to be *independent* of, or at least *insensitive* to, system states and unmeasured disturbances.

We will now concentrate on *linear* systems because they can be systematically analyzed, non-linear system will be briefly discussed in section 3.2.7. A general structure of a linear residual generator, can be described as in figure 3.4. The transfer function from the fault $f(t)$ to the residual $r(t)$ then becomes

$$r(s) = H_y(s)G_f(s)f(s) = G_{rf}(s)$$

What conditions has to be fulfilled to be able to detect a fault in the residual? In [4]

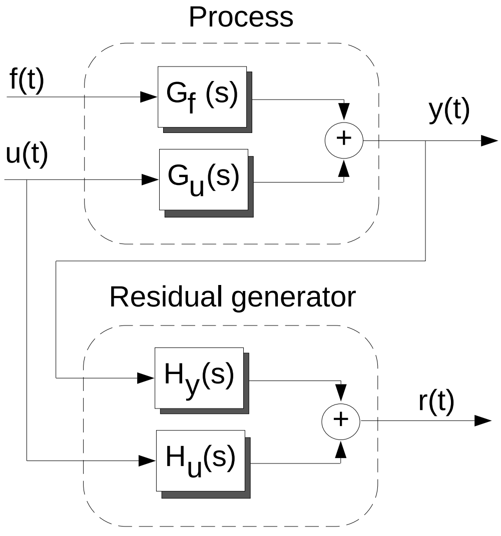

Figure 3.4. General structure of a linear residual generator

detectability has a natural definition. To be able to detect the i:th fault the i:th column of the response matrix $[G_{rf}(s)]_i$ has to be nonzero, i.e.

Definition 3.2 ***[Detectability].*** *The i:th fault is detectable in the residual if*

$$[G_{rf}(s)]_i \neq 0$$

This condition is however not enough in some practical situations. Assume that we have two residual generators with structure as in figure 3.4. When excited to a fault the residuals behave as in figure 3.5. Here we see that we have a fundamentally different behavior between $r_1(t)$ and $r_2(t)$ as $r_1(t)$ only reflects changes on the fault signal and $r_2(t)$ has approximately the same shape as the fault signal. Thus $r_1(t)$ can not be used in a reliable FDI application even though it is clear that $G_{r_1 f}(s) \neq 0$.

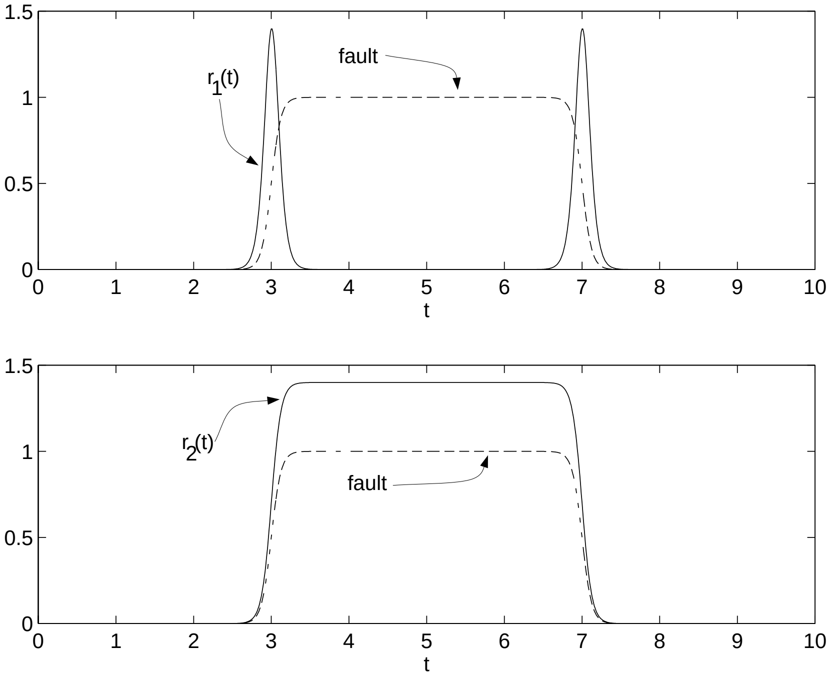

Figure 3.5. Example residuals

The difference between the two residuals in the example are the value of $G_{rf}(0)$. It is clear that residual 1 has $G_{r_1 f}(0) = 0$ while residual 2 have $G_{r_2 f}(0) \neq 0$. This leads to another definition in [4]

Definition 3.3 ***[Strong detectability].*** *The i:th fault is said to be* strongly *detectable if and only if*

$$[G_{rf}(0)]_i \neq 0$$

The above definitions show that it can be of great importance to perform a frequency analysis of the residual generator. What can be done if the designed residual generator has a response like $r_1(t)$? An easy solution can be to filter the residual, e.g. through a integrating filter.

3.2.1 Isolation strategies

If we now have strongly detectable residuals, how can isolation be achieved? In [33] two general methods are described

- Structured residuals
- Fixed direction residuals

Structured residuals

The idea behind structured residuals is that a bank of residuals is designed making each residual insensitive to different faults or subset of faults whilst remaining sensitive to the remaining faults, i.e. if we want to isolate three faults we can design three residuals $r_1(t)$,$r_2(t)$ and $r_3(t)$ to be *insensitive* to one fault each. Then if residuals $r_1(t)$ and $r_3(t)$ fire we can assume that fault 2 has occured.

Structured residuals can also be generated with a bank of observers. Here we will present the structure for *instrument fault diagnosis* (IFD), the corresponding structure for *actuator fault diagnosis* (AFD) is trivial. There are two general structures for the observer bank, the *dedicated observer scheme* (DOS) or the *generalized observer scheme* (GOS). In DOS only one measurement is fed into each observer. The i:th observer are therefore only sensitive to sensor faults in the i:th sensor. DOS is illustrated in figure 3.6. Each observer in a GOS scheme on the other hand are fed by all *but* one measurement making the i:th residual sensitive to all but the i:th measurement. GOS is illustrated in figure 3.7. Since there always exists modelling errors and disturbances not modeled residuals are never 0 even in the fault free case. This can make some residuals fire that shouldn't and vice versa. Therefore it is more likely that a GOS-bank of residuals are more reliable than a DOS-bank in a realistic environment. This because that if one residual in a DOS-scheme happen to fire in a fault free case this immediately results in a bad fault decision. However in a GOS-scheme more than half of the residuals have to misfire (if we use a majority decision rule) to make a bad fault decision. If a residual pattern, i.e. a binary vector describing which residuals that have fired, doesn't

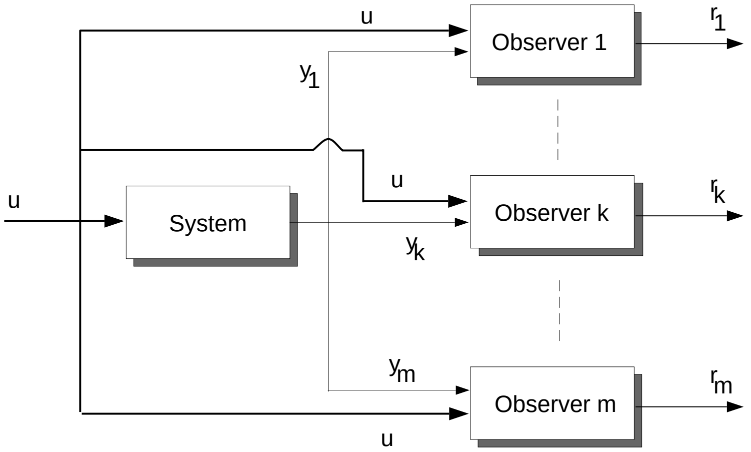

Figure 3.6. Dedicated Observer scheme for IFD

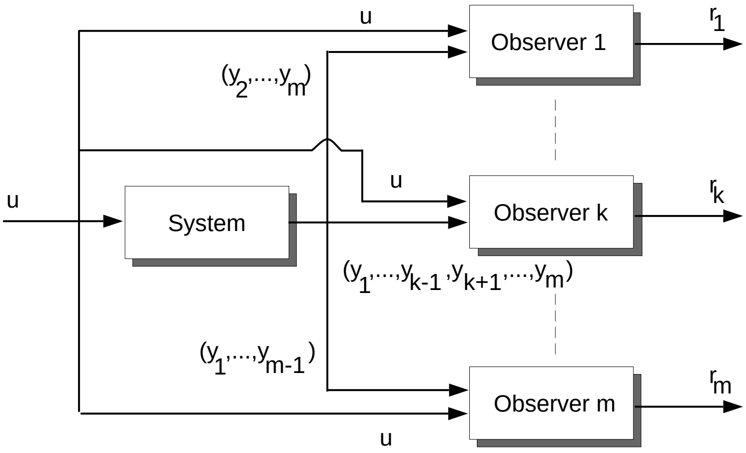

Figure 3.7. Generalized Observer scheme for IFD

correspond to any fault patterns a natural approach is to assume the faultpattern that has the smallest Hamming distance to the residual pattern. The Hamming distance is defined as the number of positions two binary vectors differ, e.g. $d((1,1,0),(0,1,1)) = 2$.

As always there is a price to pay for this increased reliability, or robustness, a GOS-scheme can only detect one fault at a time while a DOS-scheme can detect faults in all sensors at the same time. It is possible to extend a GOS scheme with extra sensors and residuals to achieve possibilities to detect and isolate multiple faults as in [16].

A thorough description of structured residuals are given in section 4.

Fixed direction residuals

This idea is the basis of the *fault detection filter* where the residual vector get a specific direction depending on the fault that is acting upon the system.

Figure 3.8 gives an geometrical illustration of this type of residuals when a fault of type 1 has occurred. The most probable fault can then be determined by finding the

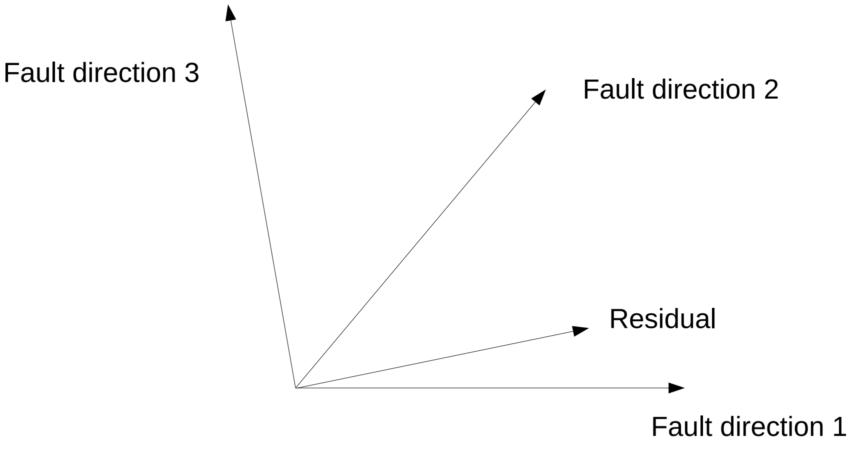

Figure 3.8. Fixed direction residuals

fault vector that has the smallest angle to the residual vector.

It can be noted that a DOS scheme can be viewed as a fixed direction residual generator with the basis vectors as directions. A GOS scheme can however not be viewed as a fixed directions residual generator as a residual there is confined to a subspace of order $n-1$ (if there are n residuals) instead of only a 1-dimensional subspace (the direction).

3.2.2 Robustness issues

One problem, as we have noticed earlier, is that unmeasurable signals often act upon the system plus the influence by modelling errors. This makes it hard to keep the false alarm rate at an appropriate level.

If it is known how the uncertainty influences the process, so called *structured* uncertainty, this information can be utilized to actively reduce or even eliminate their influence on the residuals. If it is not known how disturbances act upon the system there is little that can be done to achieve any decoupling. We actually don't produce any robustness, at best we can maximize the sensitivity to faults and minimize the sensitivity to disturbances over all operating points.

However it is possible to increase robustness in the fault evaluation stage, i.e. in the threshold selection step, e.g. by using *adaptive threshold levels* or *statistical decoupling* as described in section 3.2.6. This is also called *passive robustness*. It is not likely that one method can solve the entire robustness problem, a likely solution is one where disturbance decoupling is used side by side with adaptive thresholds.

3.2.3 Model structure

To proceed in the analysis of residual generation approaches we need an analytical model. In this report a state-representation of the model are used as

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t), u(t)) \end{aligned} \tag{3.1}$$

The linear (time-continuous) state representation

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3.2}$$

As we have noted earlier we have three general types of faults:

1. *Sensor (Instrument)* faults
   Modeled as an additive fault to the output signal.

2. Actuator faults
   Modeled as an additive fault to the input signal in the *system dynamics*

3. Component (System) faults
   Modeled as entering the *system dynamics* with any distribution matrix. Here it is seen that actuator faults only are a special case of component faults.

There are also uncertainties about the model or unmeasured inputs to the process, e.g. the load torque in an automotive engine. If these uncertainties are *structured*, i.e. it is known how they enter the system dynamics, this information can be incorporated into the model.

In the linear case and model uncertainties are supposed structured, the complete model becomes

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B(u(t) + f_a(t)) + Hf_c(t) + Ed(t) \\ y(t) &= Cx(t) + Du(t) + f_s(t) \end{aligned} \tag{3.3}$$

Where $f_a(t)$ denotes actuator faults, $f_c(t)$ component faults, $f_s(t)$ sensor faults and $d(t)$ disturbances acting upon the system. H and E is called the distribution matrices for $f_c(t)$ and $d(t)$.

3.2.4 Parameter estimation

As we noted in 3.2, process model based residual generators could be parted into two approaches parameter estimation and parity space approaches. A parameter estimation method, [18, 19] is based on estimating important parameters in a process, e.g. frictional coefficients, volumes or masses, and compare them with nominal values.

We first need to define the model structure to use. The process to be modeled typically consist of both *static relations* and *dynamics* relations, both *linear* and *non-linear.*

Theoretically there is no limit on the appearance of these relations, the parameter estimation could be done by e.g. a straightforward gradient-search algorithm. But to enable efficient estimation of model parameters here it is assumed that the model is linear in its parameters. A least squares solution are then easy to extract. Note that this in no way implies a linear model. The equation

$$y(t) = a_1\, x^2(t)$$

is linear in its parameter a_1 but is clearly non-linear.

With this assumption the model can be written as a linear regression model

$$y(t) = \varphi^T(t)\theta \tag{3.4}$$

where $\varphi(t)$ consists of inputs and old measured variables in a discrete model and output derivatives in a continuous model. θ are the model parameters to be estimated.

Example 3.1. For an ordinary linear differential equation

$$\begin{aligned} & y(t) + a_1\frac{dy(t)}{dt} + a_2\frac{d^2y(t)}{dt^2} + \ldots + a_n\frac{d^ny(t)}{dt^n} = \\ & = b_0u(t) + b_1\frac{du(t)}{dt} + b_2\frac{d^2u(t)}{dt^2} + \ldots + b_m\frac{d^mu(t)}{dt^m} \end{aligned}$$

we get

$$\begin{aligned}\varphi(t) &= \left[-\frac{dy(t)}{dt} \; -\frac{d^2y(t)}{dt^2} \; \ldots \; -\frac{d^ny(t)}{dt^n}\right. \\ &\qquad \left. u(t) \; \frac{du(t)}{dt} \; \frac{d^2u(t)}{dt^2} \; \ldots \; \frac{d^mu(t)}{dt^m}\right]^T \\ \theta &= [a_1 \, a_2 \, \ldots \, a_n \, b_0 \, b_1 \, \ldots \, b_m]^T\end{aligned}$$

Note that θ is the *model* parameters, not the *physical* parameters. θ can be written as a function of the physical parameters p as

$$\theta = f(p) \tag{3.5}$$

Note that it can be of great importance how in- and out-signals are chosen as we will see in the example below.

Example 3.2. Consider a simple linear system, a first order low pass RC-link. Here there are two physical parameters, the resistance R and capacitance C.

If the input and output voltages, u_1 and u_2 are chosen as in and out signals, the system gets

$$u_2(t) = -RC\dot{u}_2(t) + u(t) = \varphi^T(t)\theta = (-\dot{u}_2(t) \; u(t))\begin{pmatrix} RC \\ 1 \end{pmatrix} \tag{3.6}$$

In equation (3.6) we see that only one parameter appear in θ as RC. We can then conclude that the two parameters can not be estimated with this choice of input-output signals. If we instead considers the output current i_2 as output signal. The system then gets:

$$i_2(t) = -RC\dot{i}_2(t) + C\dot{u}(t) = \varphi^T(t)\theta = (-\dot{i}_2(t) \; \dot{u}(t))\begin{pmatrix} RC \\ C \end{pmatrix} \tag{3.7}$$

Here in (3.7) two parameters appear and both R and C is identifiable. In a practical problem there might not be a choice in in-out signals but the example shows that in a parameter estimation method, the in-out signal choice can be of great importance and should be analyzed.

Now when the model structure is defined we can outline the typical parameter estimation diagnosis method.

- **Data processing**
  With the help of the model and measured output data model parameters can be estimated, e.g. by minimizing the quadratic estimation error

$$V_N(\theta) = \sum_{i=0}^{N} \left(y(i) - \varphi^T(i)\theta\right)^2$$

Resulting in the well known solution

$$\hat{\theta} = \left[\varphi(t)^T \varphi(t)\right]^{-1} \varphi^T y$$

The LS-solution can easily be replaced by a RLS-estimator to achieve adaptability to a time varying process.

- **Fault detection**
  When an estimation of model parameters $\hat{\theta}$ is produced, an estimation of process parameters $\hat{p}$ can be extracted by inverting equation (3.5), this is also called *feature extraction.*

  $$\hat{p} = f^{-1}(\hat{\theta})$$

  Also $\Delta p = p_{nominal} - \hat{p}$ and σ_p can be extracted to be used in a statistical test whether a fault is acting upon the system or not.

  Δp and σ_p can be seen as residuals as they are small in the fault-free case. They are also in parameter estimation articles called *syndromes.*

- **Fault classification**
  If the statistical test mentioned above decides that a fault is present, isolation of the fault source is the final stage in a parameter estimation method.

The algorithm outlined above is an example of a typical algorithm, another approach is taken in [18] where the detection and classification steps are combined into one using a Bayes classification rule. The drawback with heuristic knowledge are that highly reliable training data, or experience is needed.

There exists another complication with the parameter estimation method. The $\varphi(t)$ vector often include time derivatives that are not measurable. In a realistic environment all measurement will be subjected to measurement noise which will make differentiating complicated. An ideal differentiator amplifies high frequency components and the typical measurement noise consists of high frequencies. One way to handle this problem are a state-variable filter approach described in [38].

3.2.5 Parity space approaches

The approaches described in this section are called parity space approaches because they generate residuals who are vectors in the parity space. The methods can be divided into open- and closed-loop approaches. In an open-loop approach there are, as the name suggests, no feedback from previously calculated residuals.

The idea behind closed-loop approaches, i.e. observer based approaches, are to use a state-estimator as a residual generator. Both structured residuals and fixed direction isolation methods is achievable with both open- and closed-loop design methods. There are a number of approaches suggested in literature, here we will address

- Parity equations from a state-space model
- State observers
- Fault detection filter

- Unknown Input Observer
- Eigenstructure assignment of observer

Note that these are *methods* to design the residual generator. Several of these designs may result in the same residual generator in the end as shown in [8].

Parity equations from a state-space model

An example of an open-loop implementation utilizing temporal redundancy. This method will be presented in detail in chapter 4.

State observers

If there are no uncertainties acting upon the system, a straightforward approach is to use a state estimator observer and compare the estimated outputs with the measured.

Consider the special case of IFD. Assume a linear system with additive sensor faults f_s as

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du + f_s\end{aligned} \tag{3.8}$$

A state observer for system (3.8) can be stated as

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + K(y - \hat{y}) \\ \hat{y} &= C\hat{x} + Du\end{aligned}$$

If $r = y - \hat{y}$ is used as the residual it can be written

$$r = y - \hat{y} = Cx + Du + f_s - C\hat{x} - Du = Ce + f_s$$

where e is the state estimation fault $e = x - \hat{x}$. The estimation error dynamics can be stated

$$\dot{e} = (A - KC)e - Kf_s$$

Assume f_s is a step from 0 to $F \neq 0$. Since $A_c = A - KC$ is a stable matrix, e will go towards a stationary value

$$e \to A_c^{-1}KF \text{ as } t \to \infty$$

As $r = Ce + f_s$ and e goes towards a non zero value the residual will be $\neq 0$ if $F \neq 0$. This result motivates the non-linear version of this residual generator that is used in a robust IFD scheme described in section 7.2.

Fault detection filter

The idea with the fault detection filter [8, 33] is, as was noted in earlier, to produce fixed direction residuals. The method is based on an observer of the form

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

Considering a fault in the $i:th$ actuator we get estimation error $e = x - \hat{x}$ dynamics as

$$\begin{aligned}
\dot{e} &= (A - KC)e + b_i f_a \\
e_y &= y - \hat{y} = Cx + Du - C\hat{x} - Du = C(x - \hat{x}) = Ce
\end{aligned}$$

Where b_i is th $i:th$ column in B. By a special choice of K it is possible to make e_y, i.e. the residual, grow in a specified direction when the $i:th$ fault has occured.

In [33] it is noted that the fixed direction approach uses up more of the design freedom compared to other observer based approaches described next who therefore supersedes the fault detection filter.

Unknown Input Observer

The unknown input observer residual generator as in [6, 7, 33, 35], is based on a generalized observer, the so called *Luenberger* observer rendering a residual generator as

$$\begin{aligned}
\dot{w}(t) &= Fw(t) + Ky(t) + Ju(t) \\
r(t) &= L_1 w(t) + L_2 y(t) + L_3 u(t)
\end{aligned}$$

Where w is an estimate of the transformed state vector Tx. Assuming a system as in (3.3) the error dynamics then gets

$$\begin{aligned}
\dot{e} &= T\dot{x} - \dot{w} = T(Ax + Bu + Bf_a + Hf_c + Ed) - Fw - Ky - Ju = \\
&= (TA - FT - KC)x + (TB - KD - J)u + Fe + TBf_a + THf_c + TEd - Kf_s \\
r &= L_1(Tx - e) + L_2(Cx + Du + f_s) + L_3 u = \\
&= -L_1 e + (L_1 T + L_2 C)x + (L_2 D + L_3)u + L_2 f_s
\end{aligned}$$

In the fault free, no disturbance case we require $r = 0$. The conditions can then be identified as

$$\begin{aligned}
FT - TA + KC &= 0 \\
J + KD - TB &= 0 \\
L_1 T + L_2 C &= 0 \\
L_2 D + L_3 &= 0
\end{aligned}$$

The conditions above must be upheld for any observer based residual generator, the unknown input observer is a method of finding all matrices in the generalized observer described above. Note that the disturbance influence can be eliminated already in the state estimate w by choosing $TE = 0$. In the eigenstructure observer described later, T is assumed to be the identity matrix, thus rendering a so called identity observer. Therefore there is no way of achieving disturbance decoupling in the state estimate, only in the residual r.

Eigenstructure assignment of observer

The eigenstructure assignment is an observer approach using an identity observer, i.e. $T = I$, to achieve disturbance decoupling in the residual. A detailed description of the eigenstructure approach is described in chapter 5.

3.2.6 Residual evaluation

Due to the uncertainties in the model used in the residual generator, measurement noise and/or that only approximate decoupling from unmeasured disturbances is achievable, residuals will not be 0 in the fault-free case. Therefore a non-zero threshold has to be selected. This is even more important in the case of unstructured uncertainties where exact disturbance decoupling in the residuals is impossible.

In [6] it is noted that when *deterministic* decoupling, i.e. decoupling of structured disturbances in the residuals, is not possible there is a possibility, if we know the statistical distribution of the residual, to use this knowledge and achieve robust FDI. This is called *statistical decoupling.*

One method who achieves statistical decoupling is the *GLR* (Generalized Likelihood Ratio) test where the $k:th$ residual is modeled as

$$r_k(t) = r_{0,k}(t) + G_k(p)f(t)$$

Where $r_{0,k}(t)$ is white noise with zero mean and G_k is the distribution matrix of the $k:th$ fault. p is the derivation operator, i.e. $\dot{y}(t) = p\,y(t)$.

A hypothesis test is then performed with the hypothesis

$$\begin{aligned} H_0 &: r_k = r_{0,k} \\ H_i &: r_k = r_{0,k} + G_{i,k}\, f_i \text{ the } \textit{i:th} \text{ fault has occured} \end{aligned}$$

The hypothesis decision can be made through a test of the likelihood ratio

$$L_i = \frac{Pr(r_1, \ldots, r_n | H_i, f_k = \hat{f}_i)}{Pr(r_1, \ldots, r_n | H_0)}$$

Where $Pr(\cdot)$ denotes the density function of the underlying stochastic process. The estimates $\hat{f}_i$ is calculated under the assumption that H_i is true. The decision is then based on the rule

$$\begin{aligned} L_i &> T_i : H_i \text{ is assumed, i.e. the } \textit{i:th} \text{ fault is assumed present} \\ L_i &< T_i : H_0 \text{ is assumed, i.e. no fault} \end{aligned}$$

The desired false alarm rate can be adjusted by choosing suitable thresholds T_i.

This approach can be easily illustrated on a one dimensional residual by figure 3.9. Assume the observed value of the residual is r. Assume H_0 is the density function of r under assumption H_0 and H_1 is the density function of r under the assumption H_1. We can directly see that H_0 is the most probable hypothesis. L_i is then an *estimation* of $\frac{v_1}{v_2}$. In this example L_i would be small as $v_1 < v_2$ and hypothesis H_0 would be assumed, just as expected.

Another more intuitive approach to robust residual evaluation is that of *adaptive thresholds.* Since the model used does not model the system perfectly, the residuals will fluctuate with changing inputs even in a fault-free situation. There might be situations where these fluctuations are so great so that no threshold level fulfills both satisfactory false alarm rate demands and missed detection probabilities.

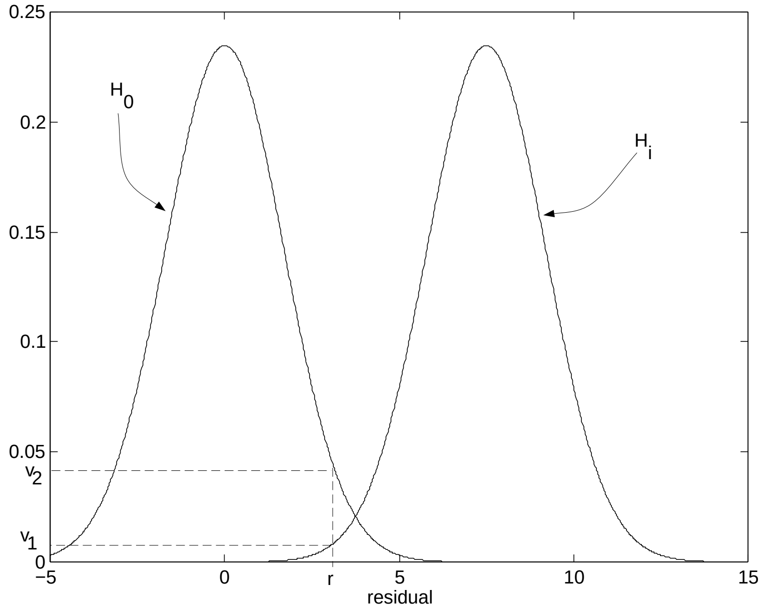

Figure 3.9. GLR illustration

The adaptive thresholds approach is as noted above based on the fact that the residuals tend to fluctuate *with* the input signals (unmeasured or measured). Examples of adaptive thresholds can be that the threshold level is scaled with the size of the input vector, i.e. $T_i(t) \propto ||u(t)||$, or time-derivative of the input vector, i.e. $T_i(t) \propto ||\dot{u}(t)||$.

In the end, we have to set the threshold levels. One simple approach is to observe the residuals in the fault free case and set the level to get the desired false-alarm rate. The residual evaluation rules used often get adapted to the application, e.g. by using time-limits on how long the residual can be above the Threshold before a fault is assumed etc. It is easy to imagine a number of ad hoc solutions to improve robustness, but a systematic approach based on Markov theory choosing the thresholds has been suggested in [46].

3.2.7 Non-linear residual generators

As noted, all previously described residual generators are *linear*. When applying a linear residual generator, based on a linearization of a non-linear system, modelling errors become dominant very quickly as the system deviates from the linearization point. One way to master this problem is to use a non-linear residual generator taking full advantage of the knowledge in the non-linear model. Non-linear residuals can be both closed-loop generators, [6], or open-loop generators [24]. Non-linear parity equations is described in [24] and used for automotive diagnosis in [9].

An example of a non-linear residual generator is given below.

Closed-loop approach

Consider a class of non-linear systems described by the differerential equations

$$\begin{aligned}\dot{x} &= f(x,u)+E_1f_1\\ y &= h(x,u)+E_2f_2\end{aligned}$$

A corresponding non-linear identity observer is given by

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x},u)+H(\hat{x},u)(y-\hat{y})\\ \hat{y} &= h(\hat{x},u)\end{aligned}$$

The error dynamics $\dot{e}=\dot{x}-\dot{\hat{x}}$ can then be calculated as

$$\dot{e}=f(x,u)-f(\hat{x},u)-H(\hat{x},u)(h(x,u)-h(\hat{x},u))+E_1f_1-H(\hat{x},u)E_2f_2$$

A Taylor expansion of $f(x,u)$ around $e=0$, i.e. $x=\hat{x}$ yields

$$f(x,u)=f(\hat{x},u)+\left.\frac{\partial f(x,u)}{\partial x}\right|_{x=\hat{x}}(x-\hat{x})+h.o.t$$

$f(x,u)-f(\hat{x},u)$ can then be written as

$$\begin{aligned}f(x,u)-f(\hat{x},u) &= f(\hat{x},u)+\left.\frac{\partial f(x,u)}{\partial x}\right|_{x=\hat{x}}\underbrace{(x-\hat{x})}_{e}+h.o.t-f(\hat{x},u)=\\ &= \left.\frac{\partial f(x,u)}{\partial x}\right|_{x=\hat{x}}e+h.o.t\end{aligned}$$

By the same line of reasoning we can state

$$h(x,u)-h(\hat{x},u)=\left.\frac{\partial h(x,u)}{\partial x}\right|_{x=\hat{x}}e+h.o.t$$

Assuming $||e||$ small enough to neglect higher order terms the expansions above results in error dynamics and a residual as

$$\begin{aligned}\dot{e} &= (\left.\frac{\partial f(x,u)}{\partial x}\right|_{x=\hat{x}}-H(\hat{x},u)\left.\frac{\partial h(x,u)}{\partial x}\right|_{x=\hat{x}})e+E_1f_1-H(\hat{x},u)E_2f_2\\ r &= \left.\frac{\partial h(x,u)}{\partial x}\right|_{x=\hat{x}}e-E_2f_2\end{aligned}$$

The observer gain $H(\hat{x},u)$ has to be designed so that $e=0$ becomes an asymptotically stable equilibrium. If there is any design freedom left, that freedom could be used to achieve approximate decoupling by using the expressions derived above. Note that it can be very hard to find the time-varying $H(\hat{x},u)$ for a general system. Non-linear observers based on the eigenstructure assignment approach is discussed briefly in section 5.6.

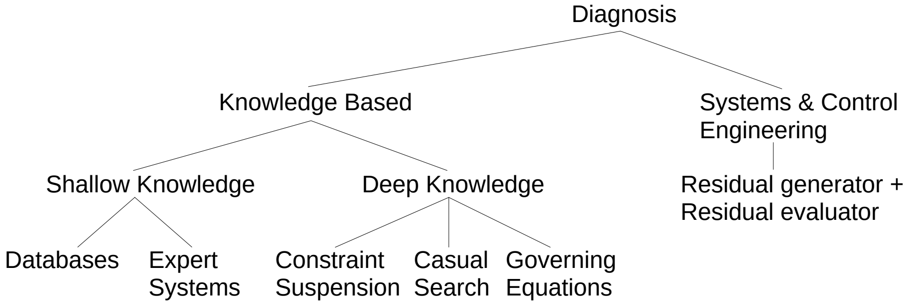

Figure 3.10. Categorization of FDI methods

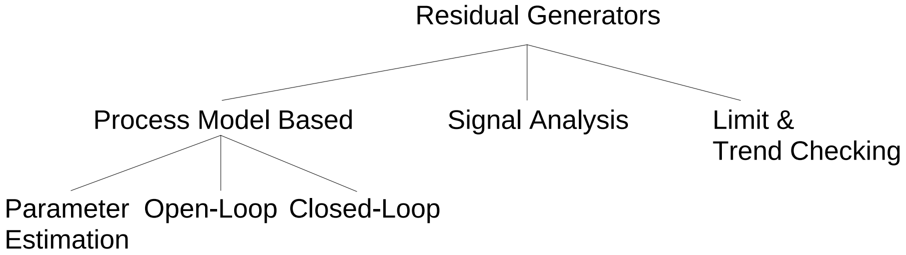

Figure 3.11. Categorization of residual generation methods

3.3 Summary of approaches in literature

To summarize the relationships between the different diagnosis methods described in this section, a tree-structure is presented in figure 3.10. The different residual generation methods are related as in figure 3.11. All these methods have their advantages and disadvantages and it is likely that in a complete diagnosis application several of these methods will be used. A comparison study between different methods is made in [20].

The presentation done here is in no way complete as there exists numerous of approaches, e.g. the neural network approach [1, 42].

3.4 Approaches to evaluate in this work

In this paper the residual generation stage is emphasized and in particular open- and closed-loop approaches are investigated further. This because they include the possibility to design a diagnosis scheme that is invariant to unmeasured structured disturbances, which exists in the automotive case in the road load.

Chapter 4

Parity equations from state-space model

In this section *structured parity equations from a state-space model* [8, 35] are examined in detail and a design example will be presented.

The parity equation strategy is an open-loop strategy that utilizes what is called *temporal redundancy* which is a type of analytical redundancy discussed in section 2.1. Temporal redundancy is sometimes referred to as *serial* redundancy.

The main idea with temporal redundancy are that given analytical knowledge on the process behavior it is possible to predict how process state and input signals affect future outputs. Considering a time window all information about any faults that may have occured during that time are present in the measurements.

This makes fault *detection* possible assuming that all signals acting upon the system are measurable. This is not always a realistic situation, therefore you need to make the diagnostic procedure invariant to unmeasurable inputs acting upon the system. And then to achieve fault isolation you also need to make the residuals insensitive to one or several of the other faults, achieving what is called structured parity equations. We will show that all this is possible by applying a multi-dimensional FIR-filter to the output estimate.

4.1 Residual generator

Restating the model given in equation (3.3), here a time-discrete form is used as it is more suited for this approach. First we consider the fault free, no disturbance case, i.e. $f_a = f_c = f_s = d \equiv 0$.

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{4.1}$$

It is not necessary to have the model on state-space form to develop the residual generator, it can just as well be developed using an input-output formulation of the model. The state-space form is chosen as it produces a clean notation.

Since we are going to utilize temporal redundancy we need an expression for the output based on previous states. The output at time $t+1, t+2, \ldots, t+s, s>0$ then becomes

$$\begin{aligned} y(t+1) &= CAx(t)+CBu(t)+Du(t+1) \\ y(t+2) &= CA^2x(t)+CABu(t)+CBu(t+1)+Du(t+2) \\ &\vdots \\ y(t+s) &= CA^sx(t)+CA^{s-1}Bu(t)+\ldots+CBu(t+s-1)+Du(t+s) \end{aligned}$$

Collecting $y(t-s), \ldots, y(t)$ in a vector yields

$$\mathbf{Y}(t) = \mathbf{R}x(t-s)+\mathbf{QU}(t) \tag{4.2}$$

where

$$\mathbf{Q} = \begin{pmatrix} D & 0 & \ldots & & 0 \\ CB & D & 0 & \ldots & 0 \\ CAB & CB & D & 0 & 0 \\ \vdots & \vdots & & \ddots & \\ CA^{s-1}B & CA^{s-2}B & \ldots & CB & D \end{pmatrix}$$

$$\mathbf{Y}(t) = \begin{pmatrix} y(t-s) \\ y(t-s+1) \\ y(t-s+2) \\ \vdots \\ y(t) \end{pmatrix} \quad \mathbf{U}(t) = \begin{pmatrix} u(t-s) \\ u(t-s+1) \\ u(t-s+2) \\ \vdots \\ u(t) \end{pmatrix} \quad \mathbf{R} = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^s \end{pmatrix}$$

Assuming k inputs and m measurements vector $\mathbf{Y}$ is $[(s+1)m]$ long and $\mathbf{U}$ is $[(s+1)k]$ long. Matrix $\mathbf{R}$ has dimensions $[(s+1)m \times n]$ and $\mathbf{Q}$ has $[[(s+1)m] \times [s+1]k]$. Note that $y(t)$ and $u(t)$ are vectors and not scalar values.

In equation (4.2), $\mathbf{Y}$, $\mathbf{U}$ and $\mathbf{Q}$ are known. Premultiplying with a vector w^T of length $[(s+1)m]$ and moving all known variables to the left side yields

$$r(t) = w^T(\mathbf{Y}(t)-\mathbf{QU}(t)) = w^T\mathbf{R}x(t-s) \tag{4.3}$$

As was described in section 3.2, equation (4.3) will qualify as a residual (parity relation) if the residual is invariant to state variables, i.e.

$$w^T\mathbf{R}x(t-s) = 0 \tag{4.4}$$

Given a vector w that satisfies (4.4) we have a residual generator where the left hand side of (4.3) is the computational form and the right hand side is the internal form. It can now easily be seen that this w can be seen as a multidimensional FIR filter. Rewriting (4.3) as

$$r(t) = [\underbrace{G_{y1}(q)\ \ldots\ G_{ym}(q)}_{G_y(q)}\ \underbrace{G_{u1}(q)\ \ldots\ G_{uk}(q)}_{G_u(q)}] \begin{pmatrix} \mathbf{Y}(t) \\ \mathbf{U}(t) \end{pmatrix}$$

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

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

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

Table 4.1. Example coding sets

To ease the notation we note that it is always possible to rearrange $\mathbf{Y}$ as

$$\mathbf{Y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{pmatrix} \tag{4.5}$$

It is now easy to see that

$$\begin{aligned} G_{y1}(q) &= w_1 q^{-s} + \ldots + w_s q + w_{s+1} \\ &\vdots \\ G_{ym}(q) &= w_{m(s+1)-s} q^{-s} + \ldots + w_{m(s+1)-1} q + w_{m(s+1)} \end{aligned}$$

A similar line of argument can be done for $G_{u1}(q), \ldots, G_{uk}(q)$. The point with this reasoning are that the residual generator becomes a multidimensional FIR filter of order $(s+1)m$ where the calculation burden for most practical purposes are small.

4.2 Isolation strategy

The next step after fault detection are fault isolation. For now we assume that we know how to make the residual invariant to faults and disturbances. details on how to achieve these invariance will be discussed in section 4.3.

With this assumption we can make a residual insensitive to one or several of the other faults and we can design a *bank* of residuals to achieve isolation. This is best explained by example. In table 4.1 three examples are presented and each row represents a residual, a 1 in position j on row i implies that fault f_j affects residual r_i. The different columns in the *coding sets* in table 4.1 is called the *fault code.* A coding set are a table that describes how different faults affect the residuals.

If for example in coding set *III* residuals r_1 and r_3 fire while r_2 don't, i.e. fault code $(101)^T$, it is probable that fault f_2 has occurred.

To detect a fault, no column can contain only zeros and to achieve isolation all columns must be unique. If these two requirements are fulfilled, the coding set is called *weakly isolating.*

To keep the false alarm rate at a low level, the thresholds making the residuals to fire are set high [8]. It is therefore more likely that a residual that should fire don't, i.e. a 1 is replaced by a 0, than the other way around, i.e. a 0 is replaced by a 1. To avoid mis-isolation, the coding set should be constructed as no two columns can get identical when ones in a column are replaced by zeros. A coding set that fulfills this requirement is called a *strongly isolating* set.

In figure 4.1 coding set I is non-isolating, II is weakly isolating and III is strongly isolating.

4.3 Residual invariance

Earlier we have assumed it possible to achieve invariance to unmeasured signals, here a method for achieving invariance is presented. If we drop the fault-free no disturbance assumption made in (4.1) the residual generator (4.3) transforms into

$$r(t) = w^T(\mathbf{Y}(t) - \mathbf{QU}(t)) = w^T(\mathbf{R}x(t-s) + \mathbf{QV}(t) + \mathbf{TN}(t) + \mathbf{S}(t)) \tag{4.6}$$

where
V is a vector of (unknown) actuator faults
N is a vector of (unknown) disturbances
S is a vector of (unknown) sensor faults
T relates to $\mathbf{N}(t)$ as **Q** relates to **U**(t). It can be seen that **T** has the same structure as **Q** with B changed to E and $D = 0$.

If we also want the residual (4.6) to be insensitive to the unknown disturbances or actuator faults we add the additional constraint:

$$w^T \left[\mathbf{T}\ \tilde{\mathbf{Q}}\right] = [0\ 0] \tag{4.7}$$

where $\tilde{\mathbf{Q}}$ are the **Q** matrix where only the columns in the B and D matrices corresponding to inputs to decouple are left.

If we want the residual to be insensitive to sensor faults we make sure that all w_i that appears in front of the sensor whose fault we wish to make the residual insensitive to are set to 0. This implies (s+1) zeros per sensor fault. If we have rearranged $\mathbf{Y}(t)$ as in (4.5) and want to make the residual insensitive to faults in the $i : th$ sensor w gets the structure:

$$w = (w_1,\ \ldots,\ w_{(i-1)s+i-1},\ 0,\ \ldots,\ 0,\ w_{i(s+1)+1},\ \ldots,\ w_{m(s+1)})^T$$

4.4 Diagnostic limits

Of course it is not possible to make the residual insensitive to an arbitrary number of disturbances and faults. We will now derive some of those limits.

What conditions must be fulfilled to make it possible to find a w that satisfies (4.4), (4.7) and then how many actuator/sensor faults are possible to decouple.

We first note that if we see disturbance as an (unknown) input we only need to consider actuator and sensor fault decoupling. Further we assume that the number of inputs, $n_u \leq n$ where n is the system order and n_u includes the number of disturbances acting upon the system.

This is a very reasonable assumption. If the assumption doesn't hold we can always rewrite our system in a way to uphold the inequality.

Consider the system

$$\dot{x}(t) = Ax(t) + \overbrace{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}}^{B} \begin{pmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{pmatrix}$$

Here we have 3 inputs and system order 2. We can define a new system with only two inputs $\tilde{u}_1(t), \tilde{u}_2(t)$ that are equivalent as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{pmatrix} = Ax(t) + \begin{pmatrix} u_1(t) + u_3(t) \\ u_2(t) + u_3(t) \end{pmatrix} = \\ &= Ax(t) + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1(t) + u_3(t) \\ u_2(t) + u_3(t) \end{pmatrix} = Ax(t) + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{u}_1(t) \\ \tilde{u}_2(t) \end{pmatrix} \end{aligned}$$

Denote the number of actuator faults and disturbances we want to decouple by s_u and the number of sensor faults by s_y. We note that

- To decouple the state influence on the residual, i.e. fulfill (4.4), we have to impose n constraints on w.
- When decoupling s_y outputs we set $s_y(s+1)$ elements in $w = 0$.
- To decouple s_u actuator faults we impose $s_u(s+1)$ if $D \neq 0$ and $s_u s$ if $D = 0$ constraints on w. The special case when $D = 0$ is easy to see when the last column in $\tilde{\mathbf{Q}}$ then becomes all zero.

In [8] s is chosen as $s = n$ if $D \neq 0$ and $s \geq n - s_u$ if $D = 0$. Summarizing and assuming $s = n$ if $D \neq 0$ and $s = n - s_u$ if $D = 0$, we can see that the number of constraints on w are:

$$n_c = \begin{cases} n + (s_u + s_y)(n+1) & \text{, if } D \neq 0 \\ n + s_u(n - s_u) + s_y(n - s_u + 1) & \text{, if } D = 0 \end{cases}$$

The w vector have as we earlier noted $[(s+1)m]$ elements and to ensure a solution other than the trivial $w = 0$ we need $(s+1)m > n_c$, i.e. an under determined equation system.

That is if $D \neq 0$

$$\begin{aligned} & (n+1)m > n + (s_u + s_y)(n+1) \\ \Rightarrow \qquad & s_u + s_y < m - \frac{n}{n+1} = m - 1 + \frac{1}{n+1} \end{aligned}$$

We also know that $n > 0 \Rightarrow \frac{1}{n+1} > 0$, which yields the upper limit on how many faults/disturbances we can decouple.

$$s_u + s_y = m - 1$$

If $D = 0$ we get

$$\begin{aligned} & (n - s_u + 1)m > n + s_u(n - s_u) + s_y(n - s_u + 1) = (s_u + s_y)(n - s_u + 1) + n - s_u \\ \Rightarrow \quad & s_u + s_y < m - \frac{n - s_u}{n - s_u + 1} = m - 1 + \frac{1}{n + 1 - s_u} \end{aligned}$$

We also know from the discussion above concerning an upper limit on number of inputs n_u that $n \geq n_u \geq s_u \Rightarrow \frac{1}{n+1-s_u} > 0$ which yields the upper limit on how many faults/disturbances we can decouple even here gets

$$s_u + s_y = m - 1$$

Chapter 5

The Eigenstructure assignment approach

In this chapter we will discuss the eigenstructure approach to FDI, as described in [33–36], and exemplify with an example to demonstrate how the theory can be used.

The eigenstructure approach is a closed-loop observer based method aiming to make the residual, not the state estimates, insensitive to disturbances. It can easily be extended to generate structured residuals to facilitate fault isolation. The eigenstructure of a matrix A is the set $\{\beta_i, v_i\}_{i=1\ldots n}$, where β_i are the eigenvalues and v_i the eigenvectors.

5.1 Residual generator

Assume a linear system as in (3.3), the residual generator is based on a straightforward state estimator, observer as

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + K(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t) + Du(t)\end{aligned}$$

From now on all time arguments will be dropped for notational simplicity.

Letting $e = x - \hat{x}$ we get estimation error dynamics as

$$\dot{e} = \dot{x} - \dot{\hat{x}} = \underbrace{(A - KC)}_{A_c} e + Bf_a + Hf_c + Ed - Kf_s$$

Now the residual generator can be formed. As in chapter 4 we premultiply the output estimation error to achieve insensitivity as

$$\begin{aligned}r(t) &= W(y(t) - \hat{y}(t)) = W(Cx + Du + f_s - C\hat{x} - Du) = \\ &= W(Ce + f_s) = WCe + Wf_s\end{aligned}$$

Going into the frequency domain we can now present the complete residual response as

$$\begin{aligned}r(s) &= WC(sI - A_c)^{-1}\left[Bf_a + Hf_c + Ed - Kf_s\right] + Wf_s = \\ &= WC(sI - A_c)^{-1}\left[Bf_a + Hf_c - Kf_s\right] + WC(sI - A_c)^{-1}Ed + Wf_s\end{aligned}$$

The disturbance decoupling condition can now be easily seen above as

$$G_{rd}(s) = WC(sI - A_c)^{-1}E = 0 \tag{5.1}$$

The problem is now how to find matrices W and K that fulfills (5.1).

Implementing a residual generator as was described above requires 4 matrix multiplications and 4 matrix additions to generate a residual. This computational burden is likely to be small in a practical situation. Another question is how to choose to dimension on the residual vector. The choice depends on the isolation strategy chosen and will be discussed below.

5.2 Isolation strategy

When using a closed-loop (observer) approach the isolation strategies available as described in section 3.2.1 are structured residuals and fixed direction residuals. When designing a structured residual bank there is no gain in choosing the residual dimension larger than 1, but if we use fixed direction residuals it is clearly seen that the dimension must be larger than 1 if more than two faults are to be isolated.

In this paper we will use structured residuals, utilizing either a GOS or a DOS observer scheme, therefore any residuals will be of dimension 1.

5.3 Residual invariance

We have earlier in this section only addressed disturbance decoupling. It can easily be seen that actuator faults can be thought of as unmeasured disturbances entering the system dynamics by the B matrix. To achieve actuator fault decoupling we enlarge the E matrix by columns in the B matrix.

Example 5.1. Consider the system

$$\dot{x} = Ax + \overbrace{[b_1\ b_2\ \ldots\ b_k]}^{B} \begin{pmatrix} u_1 + f_a \\ u_2 \\ \vdots \\ u_k \end{pmatrix} + Ed$$

If we want to consider the actuator fault f_a as a disturbance we can rewrite the system as

$$\dot{x} = Ax + B \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{pmatrix} + [E\ b_1] \begin{pmatrix} d \\ f_a \end{pmatrix}$$

We have now constructed a new E matrix and can achieve actuator fault decoupling by means of disturbance decoupling.

Component faults can by similar line of argument be thought of as disturbances.

When this is an observer approach, to achieve decoupling from sensor faults we only need to skip the feedback of a sensor, i.e. the observer is not driven by the sensor whose faults we want to decouple. Therefore will we only consider disturbance decoupling since all fault decoupling cases can be seen as special cases.

Before we can proceed and describe a method to find W and K to achieve the disturbance invariance we need some additional mathematical tools regarding eigenvectors and eigenvalues, i.e. the eigenstructure.

Lemma 5.1. *If matrix A has eigenvalues $\{\beta_i\}_{i=1\ldots n}$ then A^T has the same set of eigenvalues $\{\beta_i\}_{i=1\ldots n}$. This is equivalent to that the left and right eigenvectors of a matrix has the same set of eigenvalues.*

Lemma 5.2. *Assume A has right eigenvectors, $\{v_i\}_{i=1\ldots n}$ and left eigenvectors $\{l_i\}_{i=1\ldots n}$ corresponding to the eigenvalues $\beta_1, \ldots \beta_n$. As noted in lemma 5.1 left and right eigenvectors has the same set of eigenvalues. Then a given left eigenvector l_i (corresponding to eigenvalue β_i) is always orthogonal to the right eigenvectors v_j corresponding to eigenvalues $\beta_j \neq \beta_i$. i.e.:*

$$l_i^T v_j = 0, \text{ if } \beta_i \neq \beta_j$$

Proof. By definition we have

$$\begin{aligned} \beta_i v_i &= A v_i \\ \beta_i l_i^T &= l_i^T A \end{aligned} \tag{5.2}$$

By post multiplying (5.2) with v_j we get

$$\beta_i l_i^T v_j = l_i^T A v_j = \beta_j l_i^T v_j$$

Here it can easily be seen that

$$l_i^T v_j = 0, \text{ if } \beta_i \neq \beta_j$$

which ends the proof. □

This lemma can be extended to state that for anya diagonalizable matrix

$$l_i^T v_i \neq 0$$

where l_i and b_i corresponds to the same eigenvalue. This will be used in a proof later on.

The dynamics in (5.1) originates from $(sI - A_c)^{-1}$, i.e. the observer dynamics, this makes it interesting to analyze it further.

The matrix $(sI - A_c)^{-1}$, the so called resolvent, can be expanded in several ways. Different expansions will result in different design methods for the residual generators. We will here look into two expansions.

Lemma 5.3. *The resolvent $(sI - A_c)^{-1}$ can be expanded as*

$$(sI - A_c)^{-1} = \frac{I}{s} + \frac{A_c}{s^2} + \ldots + \frac{A_c^m}{s^{m+1}} + \ldots \tag{5.3}$$

Proof. Left multiplying (5.3) with $(sI - A_c)$ we get

$$\begin{aligned} I &= \frac{sI - A_c}{s} + \frac{(sI - A_c)A_c}{s^2} + \ldots + \frac{(sI - A_c)A_c^m}{s^{m+1}} + \ldots = \\ &= I - \frac{A_c}{s} + \frac{sA_c}{s^2} - \frac{A_c^2}{s^2} + \ldots + \frac{sA_c^m}{s^{m+1}} - \frac{A_c^{m+1}}{s^m} + \frac{sA_c^{m+1}}{s^{m+1}} + \ldots \end{aligned}$$

We see from above that every other term cancels out except for the first I term, i.e. the equality holds ending the proof. □

It can easily be seen that the decoupling condition (5.1) now transforms into

$$\begin{aligned} WCA_c^iE &= 0 \ , i = 1 \ldots n-1 \\ WCE &= 0 \end{aligned} \tag{5.4}$$

Note that i only goes up to $n-1$, this is a direct consequence of Cayley-Hamilton's theorem from which the lemma below follows.

Lemma 5.4. *A square matrix satisfies its own characteristic equation, i.e. if*

$$\det(\beta - A_c) = \beta^n + a_1\beta^{n-1} + \ldots + b_{n-1}\beta + b_n$$

then

$$A_c^n + a_1 A_c^{n-1} + \ldots + b_{n-1}A_c + b_n I = 0$$

This implies that for matrix of order n, A_c raised to any power $p \geq n$ can be written as a linear combination of $\{A_c^i\}_{i=1\ldots n-1}$

If it is possible to find a K and a W such that the rows of WC are *left* eigenvectors of A_c corresponding to eigenvalue 0 at the same time as $WCE = 0$ then decoupling according to (5.1) is achieved. We will show that under certain circumstances it is possible to place both eigenvalues *and* eigenvectors by a suitable choice of K.

As we have noted that $A_c = (A - KC)$ are the observer dynamics and placing eigenvalues in 0 yields a marginally stable system that very easily could become instable. The observer can however be designed in discrete time, when poles in 0 in a discrete system corresponds to a stable rather than a marginally stable system. The observer then corresponds to a so called dead-beat observer. The dead-beat observer has fast residual dynamics which is desirable but it also makes the observer sensitive to model faults as it uses all possible knowledge about the process to clobber the estimation fault.

Another expansion of $(sI - A_c)^{-1}$ where the eigenvalues of A_c can be selected arbitrarily, as long as the observer is stable, can be derived.

Lemma 5.5. *The resolvent $(sI - A_c)^{-1}$ can be expanded in its eigenstructure as*

$$(sI - A_c)^{-1} = \sum_{i=1}^{n} \frac{v_i l_i^T}{s - \beta_i} \tag{5.5}$$

where v_i and l_i are the right and left eigenvectors corresponding to eigenvalue β_i. v_i and l_i must be scaled so that

$$v_i^T l_i = 1$$

n is the order of matrix A_c.

Proof. *Left multiply both sides of (5.5) with $(sI - A_c)$:*

$$\begin{aligned} I &= \sum_{i=1}^{n}(sI - A_c)\frac{v_i l_i^T}{s-\beta_i} = \sum_{i=1}^{n} s\frac{v_i l_i^T}{s-\beta_i} - A_c\frac{v_i l_i^T}{s-\beta_i} = \\ &= \sum_{i=1}^{n}(sI - \beta_i)\frac{v_i l_i^T}{s-\beta_i} = \underbrace{\sum_{i=1}^{n} v_i l_i^T}_{S} \end{aligned}$$

We now need to show that $S = I$, i.e

$$Sx = x, \forall x \in \mathbf{R}^n$$

If it is shown that the equality above is fulfilled for n linearly independent vectors $\{v_i\}_{i=1\ldots n}$ then $S = I$ since x can be written as a linear combination of any set of n linearly independent vectors. If we choose the set to be the n right eigenvectors we get

$$Sv_j = \sum_{i=1}^{n} v_i l_i^T v_j, j = 1, \ldots, n$$

From lemma 5.2 we know that

$$l_i^T v_j = \begin{cases} 0 & , \text{if } i \neq j \\ 1 & , \text{if } i = j \end{cases}$$

Using this result we see directly that

$$Sv_j = v_j, j = 1, \ldots, n$$

Which completes the proof. □

Theorem 5.1. *If $WCE = 0$ and all p rows of the matrix WC are left eigenvectors of A_c then (5.1) is satisfied. p is the dimension of the residual.*

Proof. The rows of matrix WC are left eigenvectors of A_c, i.e.:

$$WC = \begin{bmatrix} l_1^T \\ l_2^T \\ \vdots \\ l_p^T \end{bmatrix}$$

The decoupling condition are

$$WC(sI - A_c)^{-1}E = WC\sum_{i=1}^{n}\frac{v_i l_i^T}{s-\beta_i}E = 0$$

But according to lemma 5.2 and since all rows of WC are left eigenvectors, l_i, we have $WCv_i = 0,\ i = p+1, \dots, n$. The decoupling condition (5.1) can then be written as

$$WC(sI - A_c)^{-1}E = WC\sum_{i=1}^{p} \frac{v_i l_i^T}{s - \beta_i} E$$

But it was assumed that $WCE = 0$, i.e. $l_i^T E = 0,\ i = 1, \dots, p$ which yields

$$WC(sI - A_c)^{-1}E = 0$$

ending the proof. □

The procedure to find W and K can be summarized as

1. Compute W so that $WCE = 0$. This determines the left eigenvectors of the observer dynamics, according to lemma 5.1.
2. Determine the desired behavior of the residuals, i.e. where the observer poles should be placed
3. Find the corresponding K that generates the desired eigenstructure, i.e. the K that generates a matrix A_c with the desired eigenvalues and eigenvectors.

As we have seen, both expansions result in a solution that require us to find a K that places eigenvalues *and* eigenvectors for $A_c = (A - KC)$, a general procedure to achieve this will now be presented.

5.3.1 Direct eigenstructure feedback design

This problem is in [36] addressed as a control-feedback problem, not an observer problem. That is to find an L giving matrix $(A - BL)$ a suitable eigenstructure. The observer problem can be seen as the *dual* control problem, i.e. by replacing A by A^T, B by C^T and L by K^T we get an observer problem.

The problem kan now be formulated as to find an L that satisfies

$$(A - BL)v_i = \beta_i v_i \tag{5.6}$$

Where the eigenstructure $\{v_i, \beta_i\}$ are as close to the desired eigenstructure as possible. The necessary and sufficient conditions to find a *real* feedback matrix L that satisfies (5.6) are:

- The $v_i \in \mathbf{R}^n$ are linearly independent.
- $v_j^* = v_i$ whenever $\beta_i = \beta_j^*$ [1]

[1] * here denotes complex-conjugate

We know from basic control theory that if a system is controllable we can place the closed-loop poles wherever we want to, see [11]. We will show by example that if one has more inputs than "necessary", i.e. all inputs are not needed to make the system controllable, one also has some freedom in placing the eigenvectors. More inputs in the control problem relates to the dual observer problem as more measurements. This is quite natural, the more outputs we have, the more freedom we get when designing the observer.

Example 5.2. Consider the system

$$\dot{x} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} u$$

With the control law

$$u = -\begin{pmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{pmatrix} x$$

Here only the first input signal is necessary to make the system controllable, and thus making it possible to place all eigenvalues at arbitrarily positions, but we have an extra input signal.

Assume we want to place the closed-loop poles (eigenvalues) in -1 and -2. After some trivial but lengthy calculations we get a solution

$$\begin{aligned} l_{11} &= 6 - l_{12} - l_{22} \\ l_{21} &= -\frac{12 - l_{12} - 7l_{22} + l_{12}l_{22} + l_{22}^2}{l_{12}} \end{aligned}$$

And l_{12}, l_{22} can be chosen arbitrarily. If we choose $l_{12} = l_{22} = 1$ we get the closed loop matrix

$$A_c = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 4 & 1 \\ -6 & 1 \end{pmatrix} = \begin{pmatrix} -3 & -1 \\ 2 & 0 \end{pmatrix}$$

Which has the the desired eigenvalues -1 and -2 with the corresponding eigenvectors $v_1 = (1 \ -2)^T$ and $v_2 = (-1 \ 1)^T$.

Here we have seen that we have two parameters that we can choose freely and still achieve the desired closed-loop eigenvalues. This extra freedom can be used to shape the eigenvectors. If we instead chooses $l_{12} = 1$ and $l_{22} = 4$ we get the closed loop matrix

$$A_c = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -3 & 4 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 2 & -3 \end{pmatrix}$$

This also has eigenvalues -1, and -2 but here the corresponding eigenvectors are $v_1 = (1 \ 1)^T$ and $v_2 = (1 \ 2)^T$. Here we see that we got different eigenvectors with this other choice of l_{12} and l_{22}. We can however not choose the eigenvectors arbitrarily as we will show next.

We will now show that the subspace where the eigenvectors can be assigned are completely determined by the eigenvalue (and the system).

To derive a base for the subspace we first need to make some definitions. With each eigenvalue β_i associate two matrices $Q(\beta_i)$ and $S(\beta_i)$ as

$$\begin{aligned} Q(\beta_i) &= \left(\begin{array}{c|c} \beta_i I - A & B \end{array} \right) \in \mathbf{R}^{n\times(n+m)} \\ S(\beta_i) &= \left(\begin{array}{c} P(\beta_i) \\ T(\beta_i) \end{array} \right) \in \mathbf{R}^{(n+m)\times m} \end{aligned}$$

where the columns of $S(\beta_i)$ forms a basis for the null space of $Q(\beta_i)$,ie:

$$Q(\beta_i)S(\beta_i) = 0 \tag{5.7}$$

If we post multiply (5.7) by any vector e_i of length m and perform the matrix multiplication we get

$$\begin{aligned} Q(\beta_i)S(\beta_i)e_i &= \left(\begin{array}{c|c} \beta_i I - A & B \end{array} \right) \left(\begin{array}{c} P(\beta_i) \\ T(\beta_i) \end{array} \right) e_i = \\ &= (\beta_i I - A)P(\beta_i)e_i + BT(\beta_i)e_i = 0 \end{aligned} \tag{5.8}$$

We can rewrite (5.6) as:

$$(\beta_i I - A)v_i + BLv_i = 0 \tag{5.9}$$

Comparing (5.8) and (5.9) we identify:

$$v_i = P(\beta_i)e_i \tag{5.10}$$

That is, v_i is spanned by the columns in $P(\beta_i)$. We can also see that

$$Lv_i = T(\beta_i)e_i = z_i \tag{5.11}$$

Which we will use to calculate the feedback (observer) gain later.

The problems left are how to find the $S(\beta_i)$ given $Q(\beta_i)$ and find e_i given desired eigenvectors. We begin to address the first problem.

As noted earlier $S(\beta_i)$ were a basis for the null space for $Q(\beta_i)$. There are several ways to find a null space basis for a matrix, several of them including inverting matrices and thus making numerical issues important. In [36] a procedure that is based upon a Singular Value Decomposition (SVD) are presented.

Applying a SVD to $Q(\beta_i) \in \mathbf{R}^{n\times(n+m)}$ yields

$$Q(\beta_i) = U \overbrace{\left(\begin{array}{cccc} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{array} \right.}^{n} \overbrace{\left| \begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \end{array} \right)}^{m} W^T$$

And since W is an orthogonal matrix, the last m columns in the product $Q(\beta_i)W$ will be 0 since $W^T W$ is a diagonal matrix. We have then found a null space of $Q(\beta_i)$ with the m last columns of V as a base, i.e. $S(\beta_i)$ consists of the last m columns of V.

Now that we have found $S(\beta_i)$ we address the last problem, how to find the e_i that yields the corresponding eigenvector v_i that is closest to the desired.

Since eigenvectors only can reside in the subspace spanned by the columns in $P(\beta_i)$, normally the desired eigenvector is not possible. Then an *approximate* decoupling will have to suffice. But it is often the case, see [36], that only a few components of the desired eigenvector are specified, i.e. the rest of the components can take arbitrarily values. This freedom can be utilized rendering a LS-solution.

Assume that v_i^d is the desired eigenvector. The rows can always be reordered by a transformation matrix R as

$$Rv_i^d = \begin{pmatrix} v_i^\theta \\ x \\ \vdots \\ x \end{pmatrix}$$

where v_i^θ are the specified components and x are the components that can take on arbitrarily values.

A vector that lies in the prescribed subspace can be reordered in the same way and can the via (5.10) be written as

$$\begin{pmatrix} v_i^o \\ x \\ \vdots \\ x \end{pmatrix} = RP(\beta_i)e_i = \begin{pmatrix} J \\ J' \end{pmatrix} e_i$$

The e_i that minimizes $|v_i^o - v_i^\theta|$ in a least-squares sense can be obtained as the well known solution

$$e_i = (J^T J)^{-1} J^T v_i^\theta$$

Now that we have found e_i we can determine the feedback gain L from equation (5.11) as

$$LV = Z$$

And if $\{v_i\}_{i=1\ldots n}$ is an independent set, L can be determined as

$$L = ZV^{-1}$$

A full method for designing the observer has now been presented, there are however some limits to the described procedure. It is assumed that A is diagonalizable, i.e. the eigenvectors are linearly independent which isn't always the case. If all β_i are different then $\{v_i\}$ form an independent set, however in the first expansion of $(sI - A_c)^{-1}$ it was required to assign m eigenvalues at 0. This can in some cases lead to dependant eigenvectors. More on how to handle this can be found in [36]. The problem with dependant eigenvectors only arise in the first expansion, in the second expansion the eigenvalues could be selected arbitrarily, i.e. it is always possible to select eigenvalues so that no two eigenvalues are equal. If no two eigenvalues are equal the eigenvectors form an independent set.

5.4 Diagnostic limits

In [33] it is stated that the necessary and sufficient conditions to solve the disturbance decoupling problem is to find W, K and that the following inequality are satisfied.

$$\operatorname{rank}(E) \leq n - p \tag{5.12}$$

Where p is the dimension of the residual. When can we find a W and K? The constraints given in section 5.3.1 must be fulfilled. A sufficient condition can be stated as

$$\operatorname{rank}(E) \leq \operatorname{rank}(C) - 1 \tag{5.13}$$

Since $\operatorname{rank}(P(\beta_i)) = \operatorname{rank}(C)$ the dimension of the e_i vectors are $\operatorname{rank}(C)$. That is, since $P(\beta_i)$ spans the subspace where the eigenvector can reside, we have $\operatorname{rank}(C)$ parameters to influence the eigenvector. Now $\operatorname{rank}(E)$ determines the number of constraints placed upon the eigenvector. This results in a equation system that is under determined, i.e. we can achieve a non-zero solution, if inequality (5.13) is fulfilled.

If inequality (5.12) is not fulfilled we can at least achieve optimal approximate decoupling. The solution of course depends on how the optimization problem are stated. In [33] the problem is stated as to find a matrix E^* where $\operatorname{rank}(E^*) = n-p$, $WCE^* = 0$ and minimize

$$||E - E^*||_F^2$$

Where the $||\cdot||_F^2$ denotes the Frobenius norm.

The solution to the problem can be found by a SVD decomposition of E as:

$$E = U \operatorname{diag}(\sigma_1, \ldots, \sigma_n) V^T$$

where U and V are orthogonal matrices and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ are the singular values. Then E^* can be found as

$$E^* = U \operatorname{diag}(\sigma_1, \ldots, \sigma_{n-p-1}, 0, \ldots, 0) V^T$$

5.5 Example

To demonstrate the design procedure a residual generator will be designed that has disturbance *and* actuator fault decoupling.

Consider a generic strictly stable fifth order system

$$\begin{aligned} \dot{x} &= Ax + Bu + Bf_a + Ed \\ y &= Cx + Du \end{aligned}$$

Where

$$A = \begin{pmatrix} -1 & 0 & 1 & 0 & 1 \\ 1 & -2 & 0 & 0 & 1 \\ 0 & 1 & -3 & 1 & 0 \\ 0 & 0 & 1 & -4 & 2 \\ -1 & -1 & 0 & 1 & -5 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix} \quad C = I_5$$

$$D = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad E = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

If $\text{rank}(C) < n$ more conditions than outlined in section 5.4 has to be fulfilled by the eigenvector. If $\text{rank}(C) < n$ then a sufficient condition to find W and K are

$$\begin{aligned} & \text{rank}(E) + n - \text{rank}(C) && \leq \text{rank}(C) - 1 \\ \Rightarrow \quad & \text{rank}(E) && \leq 2\text{rank}(C) - n - 1 \end{aligned}$$

Therefore are $\text{rank}(C)$ intentionally chosen to n in this example since extra conditions only would unnecessarily complicate the example.

Assume that we want to design a scalar residual generator that is invariant to $d(t)$ and $u_1(t)$, i.e. to find a $W = l_1^T$ such that, according to theorem 5.1 the following is fulfilled

$$WC[E\ b_1] = [0\ 0]$$

where b_1 is the first column in B. This imposes 2 constraints on WC, i.e. on the left eigenvector to A_c that is chosen as a row in WC. If $l_1 = [l_{11}, \ldots, l_{15}]^T$ the constraints can be written

$$\begin{aligned} l_{11} + l_{15} &= 0 \\ l_{12} + l_{14} &= 0 \end{aligned} \tag{5.14}$$

The poles of the system are determined by the eigenvalues of the A matrix. We, quite naturally, want the observer dynamics to be faster than the system dynamics. The fastest system pole are $\beta_{min}(A) = -5.8662$. We can then choose observer poles to be $-10, -11, -12, -13, -14$. They are intentionally selected to different values to ensure linearly independent eigenvectors.

The eigenvector, l_1^T, corresponding to eigenvalue -10 are chosen as the row in WC. The procedure to calculate l_1 are described below in Matlab code

```
Q1 = [-10*eye(5,5)-A', C'];
[U, S, V] = svd(Q1);
L1 = V(:,6:10);
P1 = L1(1:5,:);
T1 = L1(6:10,:);
J = [P1(1,:)+P1(5,:);P1(2,:)+P1(4,:)];
e1 = [1;1;1;-inv(J(:,[4,5]))*(J(:,1)+J(:,2)+J(:,3))];
l1 = P1*e1;
z1 = T1*e1;
```

The J matrix represents the constraints in equation (5.14). Here also z_1 in equation (5.11) are calculated to enable calculation of the observer gain later. $l_2, \ldots, l_5$ are determined in a similar way corresponding to eigenvalues $-11, -12, -13, -14$, only here there are no constraints on the eigenvector.

The observer gain K and the matrix W are then calculated as

```
K = ([z1, z2, z3, z4, z5]*inv([l1, l2, l3, l4, l5]))';
W = l1';
```

Their numerical values (with 4 decimals precision) are

$$
\begin{aligned}
W &= \begin{pmatrix} -0.3704 & 0.4466 & 0.5715 & -0.4466 & 0.3704 \end{pmatrix} \\
K &= \begin{pmatrix} 10.9270 & -2.6920 & 2.3141 & -0.1604 & 1.4279 \\ 0.8315 & 7.7330 & -0.5795 & 0.7637 & 0.8228 \\ -0.3893 & -2.0097 & 9.8634 & 1.2675 & 0.2758 \\ -3.4944 & -9.3532 & 0.0476 & 11.5719 & 1.7417 \\ -2.4821 & -10.0029 & -3.5535 & 6.2237 & 4.9047 \end{pmatrix}
\end{aligned}
$$

To make sure that the design has been successful we calculate the eigenvalues end eigenvectors of A_c^T (the transponate because we are interested in *left* eigenvectors) with the Matlab commando `eig` as:

```
>> [eigvec, eigval] = eig((A-K*C)')

eigvec =

   -0.0000    0.0000    0.6021    0.5647   -0.3704
   -0.9864   -0.6918    0.0096   -0.0997    0.4466
   -0.0072    0.7076   -0.7950    0.6624    0.5715
   -0.0206   -0.0923    0.0719   -0.4621   -0.4466
    0.1626    0.1103    0.0087    0.1371    0.3704


eigval =

  -11.0000         0         0         0         0
         0  -13.0000         0         0         0
         0         0  -12.0000         0         0
         0         0         0  -14.0000         0
         0         0         0         0  -10.0000
```

Here we see that the calculated l_1 is indeed an eigenvector corresponding to eigenvalue -10 fulfilling equation (5.14).

To demonstrate the residual generator the system is simulated. The system is excited with the signals in figure 5.1. The dotted lines indicates the faulty signal, i.e. u_1 is subjected to a 10% fault at $t = 5$ and u_2 at $t = 7$. The system is also subjected to an unmeasured disturbance, here in the shape of a square wave as in figure 5.2. The calculated residual generator then produces a residual as in figure 5.3, note how the residual are insensitive to both disturbance and faults in actuator 1, but reacts to the fault in actuator 2 at $t = 7$.

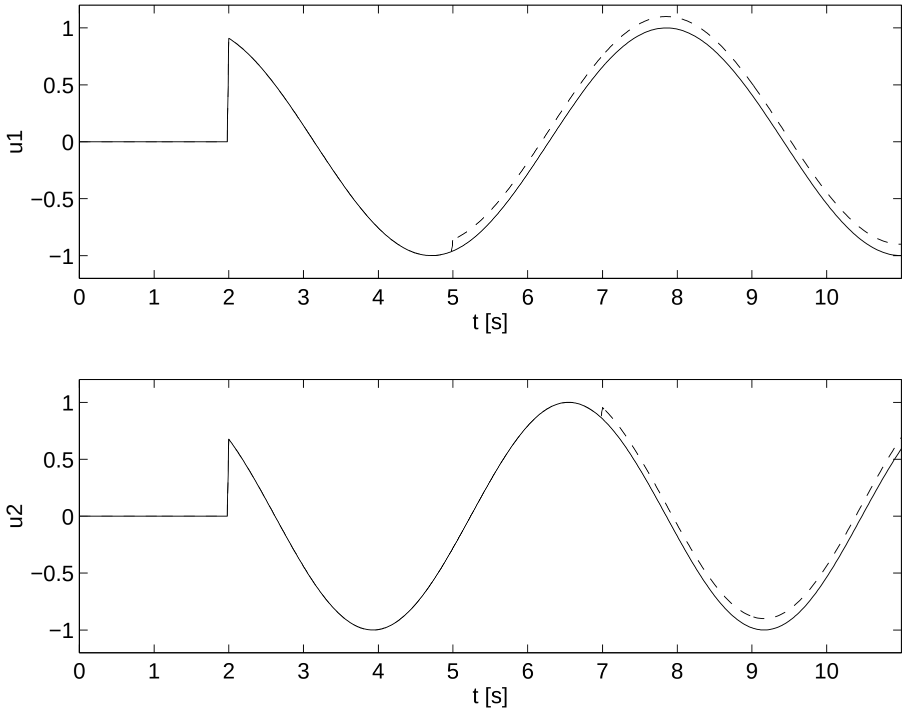

Figure 5.1. Input and fault signals

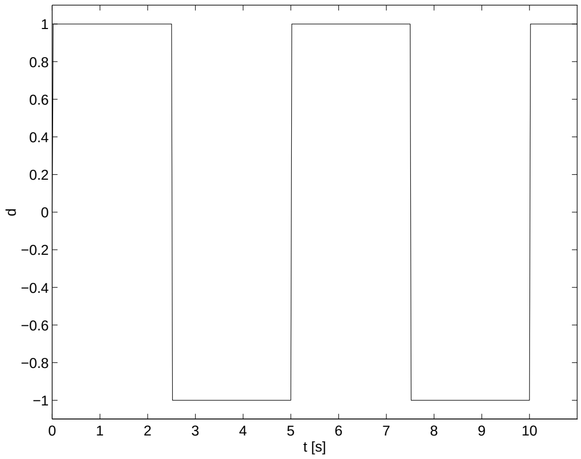

Figure 5.2. Disturbance signal

5.6 Nonlinear eigenstructure observers

In most applications it is not realistic to assume a linear model. In [33] a class of nonlinear systems are presented where the approach described in this section can be used. Decoupling is possible if the differential equations describing the system can be stated on the form

$$\begin{aligned}\dot{x} &= Ax + B(y,u) + E_1 d_1 + R_1 f \\ y &= Cx + E_2 d_2 + R_2 f + Du\end{aligned}$$

where d_1, d_2 are disturbance vectors and f are the fault vector

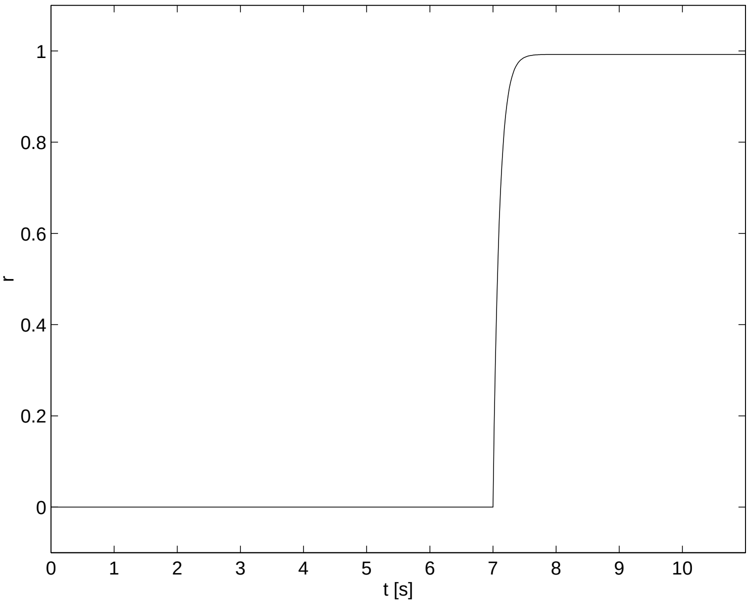

Figure 5.3. Residual

As the nonlinearity $B(y,u)$ only depends on measured variables, it can be compensated thus leading to a linear system. Another class of non-linear systems where the eigenstructure assignment approach has been shown to be applicable is when the differential equations can be written on the form

$$\begin{aligned}\dot{x} &= A(x) + B(x)u + E_1(x)d_1 + R_1(x)f \\ y &= C(x) + E_2 d_2 + R_2 f + Du\end{aligned}$$

Chapter 6

Modelling the SI-engine

As was established in section 2.2, a mathematical model is essential to accomplish high performance FDI. In this chapter we will discuss the mathematical model used to analyze the different diagnosis schemes.

First, a short introduction to *Spark Ignition*(SI) engines and a few concepts used are presented to make the modelling work done later in the chapter more easy to understand.

Then analytical expressions that builds the model are derived. Then we discuss measurements on a real SI engine, identification of model parameters, validation of the model.

To make diagnosis possible and realistic, several features has to be added, e.g. measurement noise and fault simulation.

6.1 SI-Engine fundamentals

A principle sketch of a SI-Engine is given in figure 6.1. Air flows into the engine past the throttle plate into the intake manifold. The air-speed depends on the sub pressure in the manifold and the throttle angle α. A closed throttle has $\alpha = 0°$ and a wide open throttle (WOT) has $\alpha = 90°$. When inlet valves on the cylinders are open, air flows into the cylinders and are at the same time mixed with fuel, compressed and ignited to produce a power stroke. The engine to be modeled here is a four-stroke engine, i.e. the engine cycle to produce one power stroke is split into four stages.

1. **Intake stroke**
   Starts with the piston at its top position (TC). The inlet valve is opened and the exhaust valve is closed. While the piston travels downward it draws fresh air-fuel mixture into the cylinder. When the piston reaches its lowest position (BC) the intake stroke ends.

2. **Compression stroke**
   Now both valves are closed and when the piston travels upward towards TC the mixture inside the cylinder is compressed to a fraction of its initial volume. Before the piston reaches TC combustion is initiated by ignition resulting in dramatically increased cylinder pressure.

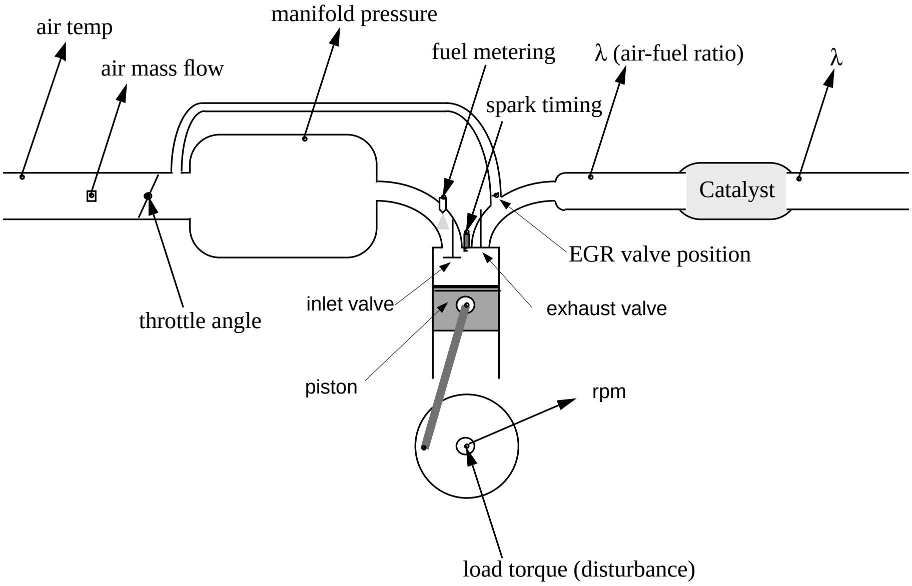

Figure 6.1. Principle sketch of the SI-engine

3. **Power stroke**
   Keeping both valves closed, the high pressure gases pushes the piston downwards and forces the crank-shaft to rotate. The developed work is about five times the work needed to compress the air-fuel mixture. When the piston reaches BC the power stroke ends.

4. **Exhaust stroke**
   As the exhaust valve is opened the remaining burned gases flows out of the cylinder, because the cylinder pressure is initially substantially higher than exhaust pressure. The piston also pushes the gases out of the cylinder as it moves upward towards TC to begin another four stroke cycle.

Note that during this four-stroke cycle, the crankshaft has rotated 720°.

The combustion process depends on the proportions of air and fuel in the cylinder mixture. The stoichiometric proportions of air and fuel is defined as when there is just enough oxygen for burning all the fuel. The stoichiometric value depends on the fuel used, but is normally between 14.57-14.70. In this thesis 14.67 will be used. The air/fuel ratio is denoted

$$(\mathrm{A/F}) = \frac{\dot{m}_a}{\dot{m}_f}$$

$\dot{m}$ represents *mass* flow, the subscript a indicates air mass-flow and the subscript f indicates fuel mass-flow. The parameter λ is the air/fuel ratio normalized with the stoichiometric value, i.e. in this thesis

$$\lambda = \frac{\frac{\dot{m}_a}{\dot{m}_f}}{14.67}$$

Modern vehicles has a catalytic converter to reduce the emissions from the engine. The efficency of the catalytic converter is highly dependant on λ. To achieve maximum efficency λ must be very near 1. λ can be measured with a *exhaust gas oxygen* (EGO) sensor. The EGO sensor has a sharp relay-like in-out characteristic, thus making it possible only to measure lean ($\lambda > 1$) or rich ($\lambda < 1$) mixtures. There is also a more expensive type of λ-sensor, the Universal Exhaust Gas Oxygen (UEGO), sometimes also called a linear oxygen sensor. The UEGO sensor can measure the actual λ and not just lean or rich mixtures. This sensor is used in later on in this chapter to compare model λ with real λ. In figure 6.1 a second λ-sensor is marked, after the catalyst. According to OBD-II the catalyst must be diagnosed. One way to make diagnosis of the catalyst possible is to use this additional λ-sensor.

In chapter 1 it was noted that the OBD II regulation stated that the EGR-system had to be monitored. Some engines use the concept of exhaust gas recycle (EGR). The EGR-valve is put out in figure 6.1. It is used to control emissions, a fraction of the exhaust gases is recycled and mixed with fresh air. The engine modeled here is not equipped with an EGR valve.

Another concept we will use when modelling the engine is the volumetric efficency η_{vol}. Volumetric efficency is an efficiency measure of the engines intake stroke and is defined as the air mass flow divided by the displaced air volume, i.e.:

$$\eta_{vol} = \frac{\dot{m}_a}{n\,\rho_{a,i}\frac{V_d}{2}}$$

Where $\rho_{a,i}$ is the inlet air density and V_d is the total cylinder volume. V_d is divided by two because on a four-stroke engine, half the total cylinder volume is displaced in one crankshaft rotation.

Another parameter that is used in the modelling work is η_{fc}, the fuel-conversion efficency. This efficency measure relates the output power to the power in the injected fuel as

$$\eta_{fc} = \frac{P}{Q_{HV}\dot{m}_f}$$

where P is the developed power, $\dot{m}_f$ is the injected fuel and Q_{HV} is the so called *heating value* of the fuel. Q_{HV} is a fuel dependant constant that defines the energy content in the specified fuel.

The material in this section was found in [15] and [37].

6.2 Physical model

The mathematical model used in this thesis is a “Mean-Value” model. Fast dynamic relationships, i.e. relationships that reaches equilibrium in a few engine cycles, are assumed static in this model and time developing processes are described by non-linear differential equations. A consequence of this simplification is that phenomena occuring during crank-shaft revolutions does not appear in the model, e.g. the crank-shaft revolution speed fluctuates in a real engine due to the two ignitions (on a four stroke, four cylinder engine) per revolution.

More information on mean-value modeling can be found in [14], and in-depth information about engine dynamics in [15].

The mathematical model consists of three major subsystems

1. Crank shaft dynamics
2. Fuel dynamics
3. Air dynamics

To get an overall view of the model and get a feel of how the three subsystems influences each other a block model is presented in figure 6.2. In a real engine λ influences the crank-shaft dynamics as it influences the fuel-conversion efficency η_{fc}. Here in this model λ is assumed near 1 and therefore no λ dependency in the crank-shaft dynamics. The model presented in this chapter also models the dynamics in the λ-sensor because the output of the λ-sensor is used in a control loop, adjusting the injected fuel to maintain λ near 1.

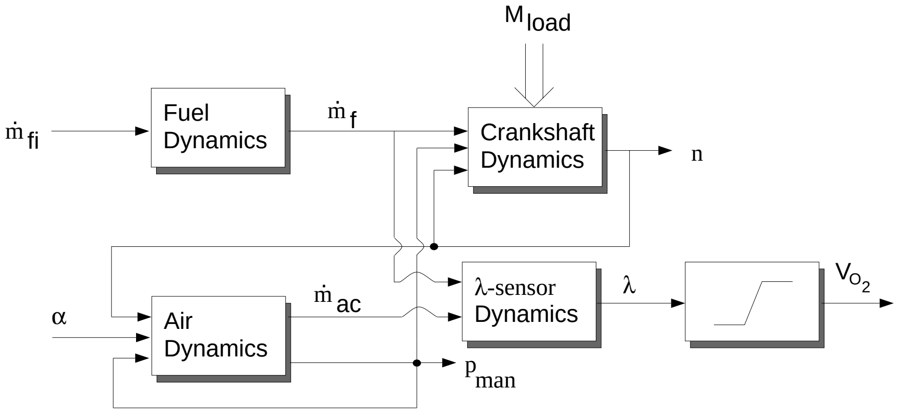

Figure 6.2. Block representation of SI-engine model

Table 6.1 summarizes most of the symbols and units used in the model. In the equations that follow factors 1/3600 and 1/60 appear. This is because the units on mass-flow and crank-shaft speed is not SI-units. The mass-flow unit is kg/h instead of m^3/s and the crank-shaft speed is rpm instead of rad/s.

Crank-shaft dynamics

The crank-shaft dynamics is modeled by Newtons Second Law for rotating masses. It can be stated as

$$M = I\dot{\omega}$$

where M is the momentum acting upon the rotating mass, ω the angular velocity and I the moment of inertia. For the crank-shaft the equation then becomes

$$I_{tot}\frac{2\pi}{60}\dot{n} = M_{gross} - M_{fric} - M_{load} \tag{6.1}$$

$$M_{gross} = \frac{\eta_{fc}\, Q_{HV}\, \dot{m}_f\, \frac{1}{3600}}{n\, 2\pi\frac{1}{60}} \tag{6.2}$$

$$M_{fric} = a_0 + a_1 n + a_2 n^2 + (a_3 + a_4 n)p_{man} \tag{6.3}$$

I_{tot}	Total moment of inertia $[kg\, m^2]$
n	Crank-shaft revolution speed [rpm]
M_{gross}	Engine developed momentum [Nm]
M_{fric}	Frictional momentum [Nm]
M_{load}	The load momentum [Nm]
η_{fc}	Fuel conversion efficency
Q_{HV}	Heating value [J/kg]
$\dot{m}_f$	Cylinder port fuel mass flow [kg/h]
p_{man}	Intake manifold air pressure [kPa]
p_{amb}	Ambient air pressure [kPa]
T_{man}	Intake manifold air temperature [K]
T_{amb}	Ambient air temperature [K]
V_{man}	Volume of intake manifold $[m^3]$
$\dot{m}_{at}$	Air mass flow past throttle plate [kg/h]
$\dot{m}_{ac}$	Cylinder port air mass flow [kg/h]
V_d	Displaced cylinder volume (entire engine) $[m^3]$
α	Throttle angle [degrees]
κ	Ratio of specific heats = 1.4 for air
τ_{ff}	Evaporation time-constant in fuel dynamics [s]
X	Fraction of injected fuel which is deposited on manifold as fuel film
$\dot{m}_{fi}$	Injected fuel mass flow [kg/h]
$\dot{m}_{ff}$	Fuel film mass flow [kg/h]
λ	Normalized A/F ratio
τ_λ	Time constant in λ-sensor dynamics [s]

Table 6.1. Symbols and units

Worth noticing is equation (6.3) describing the frictional momentum. Today, the frictional momentum is not yet completely understood and therefore there is no physical expression describing the friction. Instead we are forced to use a phenomenological model, the model used here in this thesis is equation (6.3) as suggested in [14].

Air dynamics

The model for the pressure in the intake manifold is obtained from the ideal gas law

$$pV = mRT$$

where p is pressure, V volume, m mass, R a gas constant and T is the temperature. For the engine we get

$$\begin{aligned}
\dot{p}_{man} &= \frac{RT_{man}}{V_{man}}\frac{1}{3600}\left(\dot{m}_{at}-\dot{m}_{ac}\right) && (6.4)\\
\dot{m}_{ac} &= \frac{V_d\,\eta_{vol}}{2\,R\,T_{man}}60n\,p_{man} && (6.5)\\
\eta_{vol} &= \frac{2\,R\,T_{man}\dot{m}_{ac}}{p_{man}\,V_d\,60n} = \\
&= \eta_{vn0}+\eta_{vn1}n+\eta_{vn2}n^2+\eta_{vp1}p_{man} && (6.6)
\end{aligned}$$

Like for the frictional momentum there is no well accepted physical model for η_{vol} and we are also here forced to use a phenomenological model. The model used in this thesis is equation (6.6), for more details see [14].

The air flow past the throttle can be closely approximated by the air mass flow of a compressible fluid through a converging nozzle. In its simplest form, this can be written as [15, 17]

$$\begin{aligned} \dot{m}_{at} &= f(\alpha, p_{man}) = \\ &= \underbrace{c_t \frac{\pi}{4} D^2 \frac{p_{amb}\sqrt{\frac{2\kappa}{\kappa-1}}}{\sqrt{RT_{amb}}}}_{K_{at}} \beta_1(\alpha)\beta_2(p_{man}) + \dot{m}_{at0} \end{aligned} \tag{6.7}$$

The functions β_1 and β_2 is given by

$$\beta_1(\alpha) = 1 - \cos(\alpha - \alpha_0) \tag{6.8}$$

$$\begin{aligned} \beta_2(p_{man}) &= \\ &= \begin{cases} \sqrt{\left(\frac{p_{man}}{p_{amb}}\right)^{\frac{2}{\kappa}} - \left(\frac{p_{man}}{p_{amb}}\right)^{\frac{\kappa+1}{\kappa}}} & \text{, if } \left(\frac{p_{man}}{p_{amb}}\right) \geq \left(\frac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \\ \sqrt{\frac{\kappa-1}{\kappa+1}\left(\frac{2}{\kappa+1}\right)^{\frac{2}{\kappa-1}}} & \text{, otherwise} \end{cases} \end{aligned} \tag{6.9}$$

In this thesis T_{man} is assumed constant room temperature (290 K) and p_{amb} is assumed normal air pressure (100 kPa).

Fuel dynamics

$$\ddot{m}_{ff} = \frac{1}{\tau_{ff}}(-\dot{m}_{ff} + X\dot{m}_{fi}) \tag{6.10}$$

$$\dot{m}_f = (1 - X)\dot{m}_{fi} + \dot{m}_{ff} \tag{6.11}$$

This is a fuel flow model as in [14] where part, $(1-X)\dot{m}_{fi}$, of the injected fuel mixes with the air directly and the rest strikes the manifold and becomes a puddle at the engine port, the fuel film.

The fuel in the fuel-film is evaporated off the heated intake manifold with a time-constant τ_{ff}.

λ-Sensor Dynamics

The λ-sensor dynamics is modeled by a first order system. λ is the real air-fuel ratio, and $\tilde{\lambda}$ is the output of the sensor dynamics. V_{O_2} is the output voltage of the sensor.

$$\begin{aligned} \dot{\tilde{\lambda}}(t) &= \frac{1}{\tau_\lambda}\left(-\tilde{\lambda}(t) + \lambda(t - \tau_d)\right) = \\ &= \frac{1}{\tau_\lambda}\left(-\tilde{\lambda}(t) + \frac{\frac{\dot{m}_{ac}(t-\tau_d)}{\dot{m}_f(t-\tau_d)}}{14.67}\right) \end{aligned} \tag{6.12}$$

$$V_{O_2} = \sigma(\tilde{\lambda}) \tag{6.13}$$

Equation (6.13) models the fact that the most common λ-sensor, the EGO-sensor, has a sharp relay-like in-out characteristic.

In equations (6.12) and (6.13) the time dependency has been explicitly noted, this because in all other model equations there is no time-delay, but here a time-delay τ_d is present. This exists because there is a significant time-delay between the inputs to the engine and the oxygen sensor output. The delay consists of three components

1. Hold-up time in the cylinders
2. Transport delay in the exhaust manifold and pipe
3. Reaction delay of the oxygen sensor

The delay is here modeled as $\tau_d \propto \frac{1}{n}$, but a more accurate model can be found in [9].

The output of the sensor is used in the λ controller mentioned earlier. As $\dot{m}_{ac}$ is a function of n and p_{man} according to equation (6.5) we can by feeding back n and p_{man} calculate the ideal fuel mass flow, i.e. the fuel mass flow that results in $\lambda = 1$ as

$$\dot{m}_f = \frac{\dot{m}_{ac}}{14.67}$$

As there exists fuel-dynamics, we can't control $\dot{m}_f$ directly, we can only control $\dot{m}_{fi}$. This plus measurement noise leads to a λ that isn't identically 1. The output of the λ-sensor is used with a PI-controller to control the air-fuel ratio.

6.2.1 State representation

From the modelling work done above it is possible to present a non-linear state form of the system. One natural set of state variables are as

$$\begin{aligned} x_1 &= n \\ x_2 &= p_{man} \\ x_3 &= \dot{m}_{ff} \\ x_4 &= \tilde{\lambda} \end{aligned}$$

These state-variables, except for x_4 as it is not modeled there, are the same as used in [14]. With $\alpha = u_1$, $\dot{m}_{fi} = u_2$ and $M_{load} = d_1$ we get

$$\begin{aligned} \dot{x}_1 &= \frac{60}{2\pi I_{tot}}\left\{\frac{\eta_{fc}Q_{HV}}{2\pi\, 60\, x_1}((1-X)u_2 + x_3) - M_{fric}(x_1, x_2) - d_1\right\} \\ \dot{x}_2 &= \frac{R\, T_{man}}{V_{man}\, 3600}\left\{(K_{at}\beta_1(u_1)\beta_2(x_2) + \dot{m}_{at0}) - \frac{V_d\, \eta_{vol}(x_1, x_2)}{2\, R\, T_{man}} 60\, x_1\, x_2\right\} \\ \dot{x}_3 &= \frac{1}{\tau_{ff}}(-x_3 + X\, u_2) \\ \dot{x}_4 &= \frac{1}{\tau_\lambda}\left\{-x_4 + \frac{1}{14.67}\frac{V_d\, \eta_{vol}(x_1, x_2)}{2\, R\, T_{man}} 60\, x_1\, x_2 \frac{1}{(1-X)u_2 + x_3}\right\} \end{aligned}$$

6.2.2 Model assumptions

Apart from the assumptions made when using a mean-value model there are some additional model assumptions made in this thesis, for clarity the most important assumptions made are listed below

- mean-value model assumptions
- phenomenological model for frictional momentum, equation (6.3)
- phenomenological model for η_{vol}, equation (6.6)
- p_{amb} assumed constant = normal air pressure 100 kPa
- T_{man} assumed = room temperature, 290 K
- The time displacement τ_d in equation (6.12) is $\propto \frac{1}{n}$

6.3 Measurements

Before the mathematical model is ready to be used in a simulation tool, e.g. Simulink, the different parameters in the model, e.g. V_{man} in equation (6.4) need to be determined numerically. This is accomplished by doing experiments on a real SI-engine. The parameters are then identified so that the model corresponds as, in some sense, close as possible to the real engine. Some of the parameters had already been determined before this work; or are physical constants only to be looked up in a book. The previously identified frictional model coefficients used in this thesis are

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 7.4081\,10^{1} \\ -6.1214\,10^{-3} \\ 2.3851\,10^{-6} \\ -3.5173\,10^{-1} \\ 6.7426\,10^{-5} \end{pmatrix}$$

Figure 6.3 shows the identified function. The other constants used in the model, not identified here, are listed in table 6.2.

Variable	Value
η_{fc}	0.40
τ_{ff}	0.3
τ_{λ}	0.15
τ_d prop. const	195
κ	1.4
X	0.2
R	0.2870
Q_{HV}	$4.3\,10^{7}$

Table 6.2. Model constants

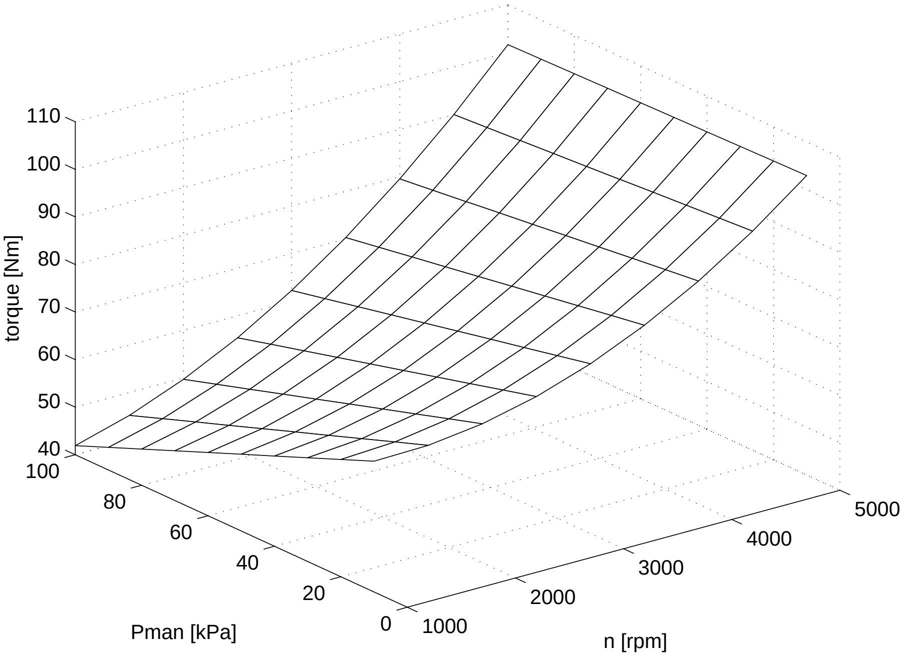

Figure 6.3. Engine friction

The parameters that needed to be identified was the parameters in equations (6.4)-(6.9), i.e. the equations describing air dynamics. That is, the following parameters had to be identified

- V_{man} in equation (6.4)
- $\eta_{vn0}, \eta_{vn1}, \eta_{vn2}$ and η_{vp1} in equation (6.6)
- K_{at}, α_0 and $\dot{m}_{at0}$ in equation (6.7)

It is essential for the success of the identification that the chain of actions leading to the actual identification is well thought through. The following steps had to be considered

1. Operating range of the model
2. Sampling frequency and anti-alias filtering

Model operation range

When developing a model for a system this complex it is hard to achieve a complete description of the system. The model can therefore only work satisfactory in a certain operating range, outside which there is no guarantee of model behavior. The model operating range is naturally the range where identification data has to be collected.

When establishing the model operating range it is important to bear in mind that a balance between two conflicting goals exists, as in almost all control problems, good model correspondence versus the size of the operating range.

In this work it isn't important that the model corresponds perfectly with the laboratory engine, the main goal is that the model behaves like *an* engine, not *the* engine. Therefore a rather large operating range has been chosen as

$$\begin{aligned} 1000 \leq & \quad n & \leq 4000 \\ 30 \leq & \quad p_{man} & \leq 100 \end{aligned}$$

Sampling frequency and anti-alias filter

The choice of sampling frequency, f_s and cut-off frequency, f_c, of the anti-alias filter are important choices.

Constraints/requirements on the anti-alias filter:

1. For simple filter design, first order RC-filters are to be used
2. Noise (assumed white) power attenuation at $f_s/2$ at least 20 dB

A first order filter has asymptotical power attenuation of 20 dB/decade and because of the mean-value model only frequencies below the largest crank-shaft rotation frequency $f_{max} = \frac{n_{max}}{60}$ are interesting. The following inequality can then be formulated

$$\begin{aligned} & f_s/2 > 10\, f_c > 10\, \frac{n_{max}}{60} \\ \Rightarrow \qquad & f_s > 20\, \frac{n_{max}}{60}\bigg|_{n_{max}=4000} > 1333Hz \end{aligned}$$

$n_{max} = 4000$ is deducted from the chosen model operating range. f_s was then chosen to 5000 Hz, an oversampling by 3.75 times.

The filter was finally constructed as $R = 274\ k\Omega$ and $C = 2.2\ nF$. This filter has $f_{c,3dB} = 264$ Hz and 20 dB attenuation at 2.6 kHz which approximately fulfills the conditions layed up for the filter.

The actual measurements was performed by a Matlab-script who measured all the desired working points 2 times each, this to be able to use the most reliable measurements. The Matlab-script used can be found in appendix D.

6.4 Identification

6.4.1 η_{vol} identification

η_{vol} is identified first since the η_{vol} model is used when identifying the other parameters. The model used, equation (6.6), is linear in its parameters and therefore a simple least-squares solution is possible.

The identification resulted in the parameters

$$\begin{pmatrix} \eta_{vn0} \\ \eta_{vn1} \\ \eta_{vn2} \\ \eta_{vp1} \end{pmatrix} = \begin{pmatrix} -8.2119\, 10^{-2} \\ -3.0125\, 10^{-5} \\ 1.0573\, 10^{-8} \\ 9.9272\, 10^{-3} \end{pmatrix}$$

Figure 6.4 shows the identified function.

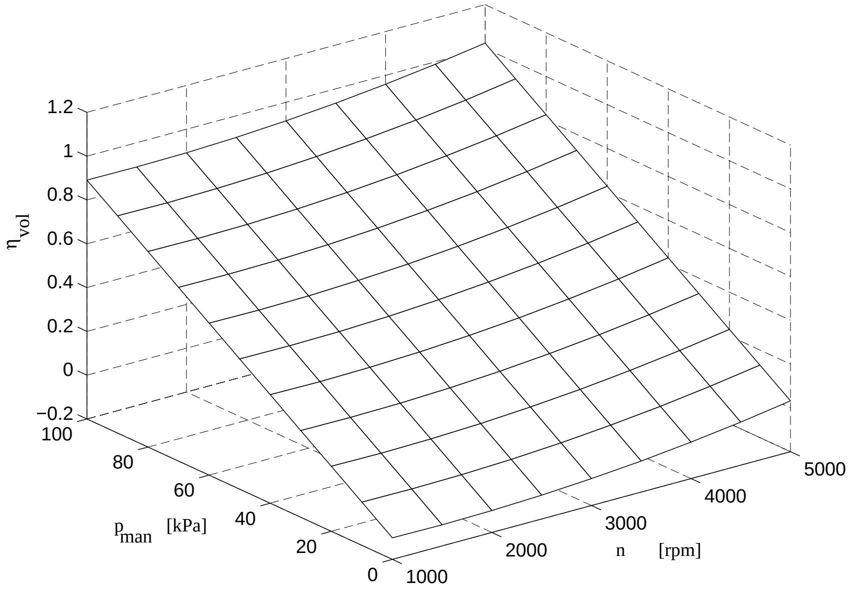

Figure 6.4. Volumetric efficency, η_{vol}, function

6.4.2 Air-flow past throttle identification

There are three parameters to be identified in equation (6.7), K_{at}, α_0 and $\dot{m}_{at0}$. Equation (6.7) is not linear in all parameters, it is non-linear in parameter α_0 and identification is therefore not as easy as above. The identification is done numerically by a steepest-descent search with a least-squares cost function.

The identification resulted in the values

$$\begin{pmatrix} \dot{m}_{at0} \\ K_{at} \\ \alpha_0 \end{pmatrix} = \begin{pmatrix} 6.7557 \\ 6109.1 \\ 5.1222 \end{pmatrix}$$

The identified function is plotted in figure 6.5

6.4.3 V_{man} identification

To identify V_{man} in the dynamic equation (6.4), a response to a throttle step is measured. This is the same experiment as is described in [15].

To avoid differentiating the noisy measured p_{man} we integrate equation (6.4). We then get

$$p_{man}(T) - p_{man}(0) = \frac{R\,T_{man}}{V_{man}} \frac{1}{3600} \int_0^T \dot{m}_{at}(\tau) - \dot{m}_{ac}(\tau)\, d\tau \tag{6.14}$$

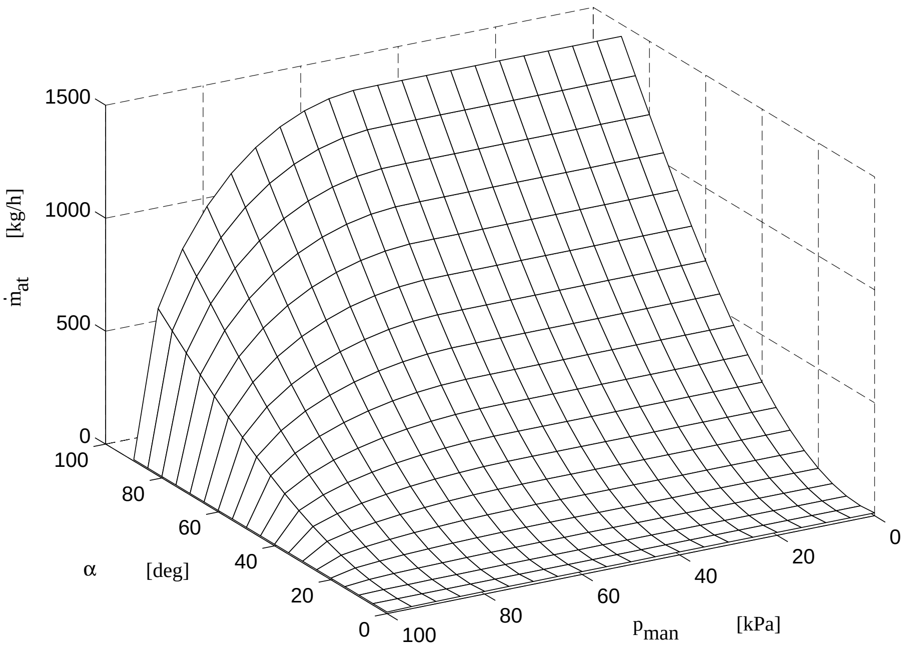

Figure 6.5. Air flow past throttle plate function

$\dot{m}_{at}$ is measured and $\dot{m}_{ac}$ is calculated with equation (6.5) and (6.6), the integral can then be approximated by a trapezoidal approximation. This resulted in the identified volume $V_{man} = 4.3$ liters.

6.5 Model validation

The derived and identified model now need to be implemented and then validated. The Simulink implementation can be found in appendix B. It is important to stress that the model here need not correspond perfectly with the laboratory engine to use the model for evaluation of diagnosis schemes, it only need to behave *like* a SI-engine where the influences of input signals affect the system approximately as in a real engine. Therefore will the validation of the model only consist of qualitative comparisons of step responses to see that time constants and general behavior etc. are approximately correct.

When validating air dynamics (since these are the dynamics that has been identified in this work) we see that in equation (6.7), if we let $\beta = \beta_1(\alpha)\beta_2(p_{man})$, the air mass flow past the throttle $\dot{m}_{at}$ is now a linear function in β. Plotting model vs. measurements renders figure 6.6, where the solid line is model behavior and the '+' marks are measurements. As can be seen, the linear relationship holds.

To compare the complete engine model and real engine behavior, a step in the throttle is made at $t = 0$ and important entities are measured and compared with corresponding model behavior. Results is plotted in figure 6.7, the dotted lines are model outputs and solid lines are real measured signals.

The decrease in crank-shaft speed after the initial speed increase due to the throttle step depends on a crank-shaft speed control algorithm, with the load torque as control signal, that is active controlling the speed back to 2500 rpm again. As seen, model

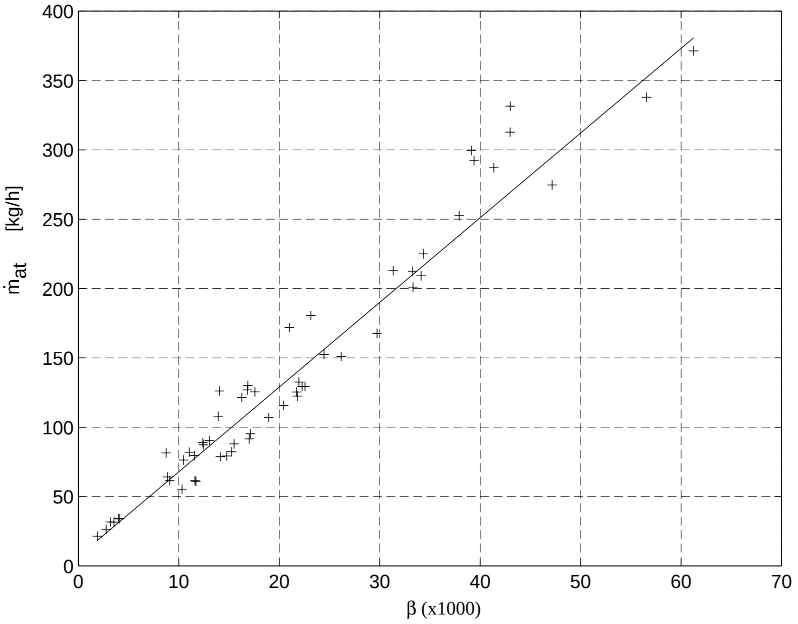

Figure 6.6. Air mass flow plotted against β

behavior behaves well enough to be used as a diagnosis simulation model.

6.6 Diagnosis adaptations of model

Measurement noise

Measurement noise is modeled as white noise added to sensor outputs. The noise power has been selected to make model outputs “look” like real measurements.

Fault simulation

In this thesis we will investigate FDI approaches to diagnose all three types of faults, actuator, component and sensor (instrument) faults. The fault sources chosen to analyze are:

- **Actuator faults**
  1. Fuel injector faults
  2. Throttle actuator fault
- **Component faults**
  1. Manifold leakage
- **Sensor faults**
  1. Rpm sensor faults
  2. p_{man}-sensor faults

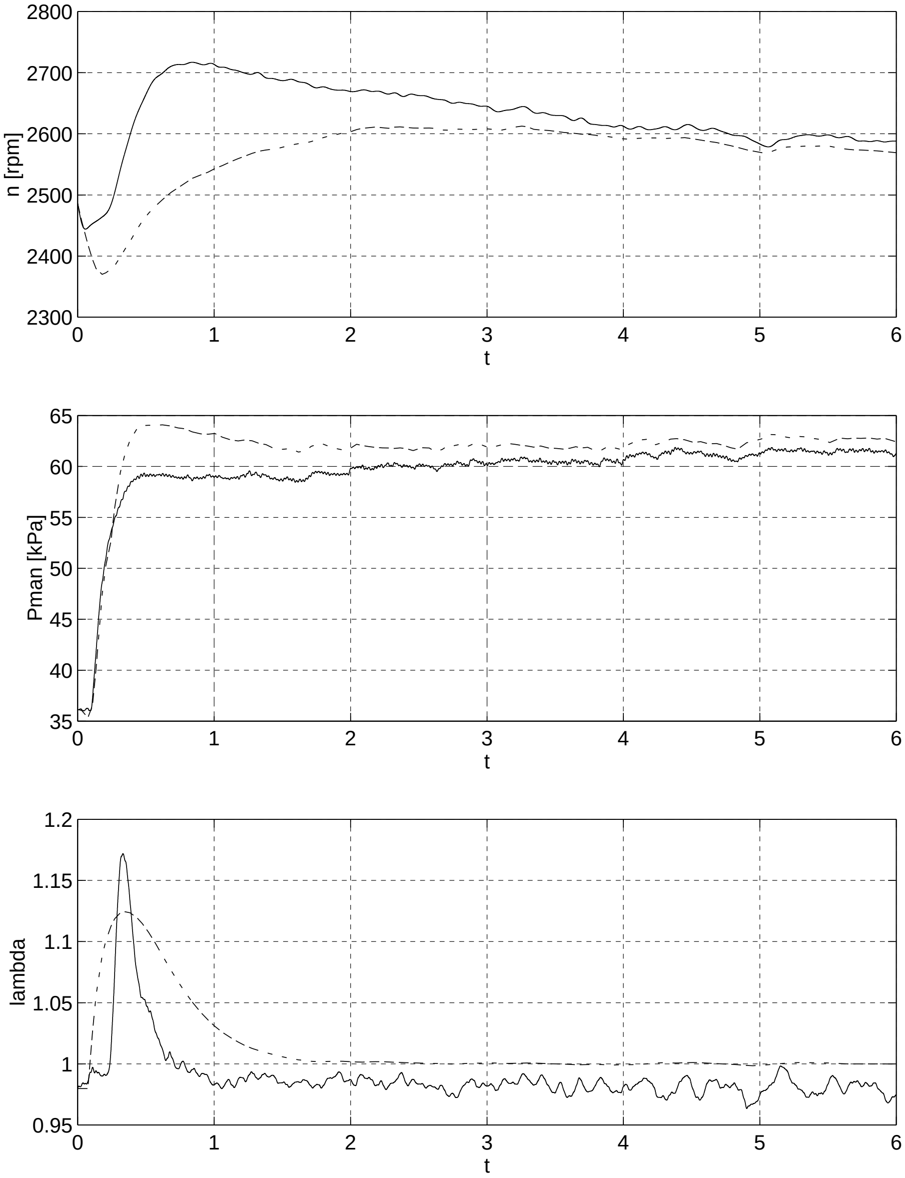

Figure 6.7. Engine and model response to throttle step

3. $\dot{m}_{at}$-sensor faults

There is actually no loss of generality when choosing these particular faults to analyze since all three types of faults described in chapter 3 are covered.

Actuator Faults

All actuator faults, i.e. fuel injector and throttle actuator faults, are here modeled as an additive disturbance entering the system *dynamics* in the same way as the actuator signal. This can be illustrated as in figure 6.8.

Since the approaches examined here in this thesis make no assumptions on the fault signal, e.g. this model also models multiplicative faults. If a 10% bias fault are to be

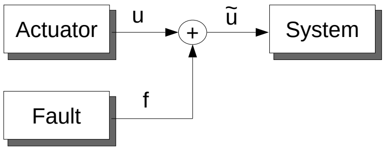

Figure 6.8. Actuator fault model

simulated, i.e. $\tilde{u} = 0.9u$ the fault signal can be set to $f = -0.1u$.

Sensor Faults

Sensor faults, i.e. rpm, p_{man} and $\dot{m}_{at}$-sensor faults, are also modeled as an additive disturbance and can be illustrated as in figure 6.9.

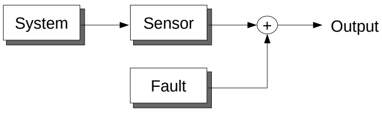

Figure 6.9. Sensor fault model

Component Fault

The component fault, manifold leakage, are here modeled as extra air flow into the manifold. Restating equation (6.4) with the fault element $\dot{m}_{leak}$ introduced results in:

$$\dot{p}_{man} = \frac{RT_{man}}{V_{man}} \frac{1}{3600} \left(\dot{m}_{at} - \dot{m}_{ac} + \dot{m}_{leak}\right) \tag{6.15}$$

Chapter 7

Diagnosis applied to automotive engines

In this chapter we will discuss diagnosis on automotive engines, and investigate the possibility to apply the methods previously described on the SI-engine model derived in chapter 6. We will investigate the possibility to use parity equations from state-space models, IFD observers and an approach based on eigenstructure observers.

In [5] ideal properties of a diagnostic procedure applied to an automotive engine are suggested and important examples are:

1. **Low computational properties**
   The lower the computational load the cheaper on-board processor can be used.

2. **Insensitive to unmeasured disturbances**
   In a realistic situation it is not uncommon that there are signals having significant influence on the process that are not measurable. On an automotive vehicle the load torque, e.g. if the vehicle is traveling down or uphill, is such a signal that can not be measured. To make diagnosis possible on such a process it is necessary to make the diagnostic algorithm insensitive to the unmeasured disturbance.

3. **Model robustness**
   Different vehicles behaves, quite naturally, slightly different, e.g. due to aging and production tolerances. Therefore are robustness to model faults highly desirable.

4. **No active diagnosis**
   It is called active diagnosis when the procedure assumes special, predefined, input signals acting upon the system. This will probably have negative influence on the process, e.g. the driving comfort in a car. Therefore diagnosis procedures relying on active diagnosis is not desired.

5. **Low false-alarm rate, low missed fault probability**
   Since legislative regulations state that a car manufacturer is fined if the on-board diagnosis system has a too high false-alarm rate or to high missed fault detection probability, these characteristics are important.

Variable	Value	
n	2600	[rpm]
p_{man}	64.5	[kPa]
α	28°	
$\dot{m}_{fi}$	9	[kg/h]

Table 7.1. Operating point

6. **Full operating range functionability**
   To fulfill OBD-II demands, the emission system must be diagnosed over the whole operating range. Therefore it is essential that the diagnosis system performs well in the whole operating range.

Diagnosis on automotive engines have been described in literature. The most common approach are different methods based on structured parity equations as described in [9, 10, 23]. Also the failure detection filter has been used in [41].

Other approaches and topics related to automotive diagnosis can be found in [5, 16, 21, 22, 25, 26, 29, 40, 43].

7.1 Parity equations from state-space model

Here a *linear* residual generator is designed as was described in chapter 4 creating structured residuals. The model used in the filter design is a linearization around a operation point as in table 7.1. The assumption made when converting the time-continuous model into a time-discrete model is that all input signals to the system is piecewise constant. This yields a 2-state representation as

$$\begin{aligned}\Delta x(t+1) &= A\Delta x(t) + B\Delta u(t) + Ed(t) + K_1 \begin{pmatrix} f_{a1}(t) \\ f_{a2}(t) \\ f_{c1}(t) \end{pmatrix} \\ \Delta y(t) &= C\Delta x(t) + D\Delta u(t) + K_2 \begin{pmatrix} f_{a1}(t) \\ f_{s1}(t) \\ f_{s2}(t) \\ f_{s3}(t) \end{pmatrix}\end{aligned}$$

$$A = \begin{pmatrix} -1.6688 & 4.1250 \\ -0.2926 & -15.8177 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 410.3077 \\ 55.6064 & 0 \end{pmatrix}$$

$$E = \begin{pmatrix} -23.3822 \\ 0 \end{pmatrix} \qquad K_1 = \begin{pmatrix} 0 & 410.3077 & 0 \\ 55.6064 & 0 & 5.3471 \end{pmatrix}$$

$$C = \begin{pmatrix} 1.0000 & 0 \\ 0 & 1.0000 \\ 0 & -0.6655 \end{pmatrix} \qquad D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 10.3995 & 0 \end{pmatrix}$$

$$K_2 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 10.3995 & 0 & 0 & 1.0000 \end{pmatrix}$$

	a_1	a_2	s_1	s_2	s_3	M_{load}	c_1
r_1	0	0	1	1	1	0	1
r_2	1	0	1	1	1	0	1
r_3	1	0	0	1	1	0	$\underline{\mathbf{0}}$
r_4	1	0	1	0	1	0	1
r_5	1	0	1	1	0	0	1
r_6	1	0	$\underline{\mathbf{0}}$	1	1	0	0

Table 7.2. Coding set

	a_1	s_2	s_3	c_1
r_1	0	1	1	1
r_4	1	0	1	1
r_5	1	1	0	1
r_6	1	1	1	0

Table 7.3. Reduced coding set

$$\text{where } \Delta x = \begin{pmatrix} \Delta n \\ \Delta p_{man} \end{pmatrix}, \Delta u = \begin{pmatrix} \Delta \alpha \\ \Delta \dot{m}_{fi} \end{pmatrix}, d = M_{load}, \begin{pmatrix} f_{a1} \\ f_{a2} \end{pmatrix} = \begin{pmatrix} \text{Throttle actuator fault} \\ \text{Fuel injector fault} \end{pmatrix},$$

$$f_{c1} = \text{Manifold leak and } \begin{pmatrix} f_{s1} \\ f_{s2} \\ f_{s3} \end{pmatrix} = \begin{pmatrix} \text{rpm-sensor fault} \\ p_{man}\text{-sensor fault} \\ \dot{m}_{at}\text{-sensor fault} \end{pmatrix}$$

The model derived in chapter 6 had 3 states (4 if the λ-sensor dynamics were included), but since there are no measurements revealing any information about the fuel-dynamics state we here have to assume fuel dynamics in perfect accordance with the model, hence only a two state model.

To isolate all 6 different type of faults we need 6 residuals, each independent of one fault each. All residuals should also be independent of the disturbance d. This is however not possible for this model, this can easily be seen as the disturbance d enters the system dynamics in the same way as faults in the $\dot{m}_{fi}$-sensor, f_{a2}. This means that any residual decoupling disturbance, automatically decouples any faults in the $\dot{m}_{fi}$-sensor. This is seen in the resulting coding set in table 7.2, the second column corresponding to a_2 is all zero. Note that this observation only holds for this linearization of the non-linear model, in the full non-linear case it might very well be possible to make a distinction between the two. It can also be seen that when decoupling s_1 you also decouple c_1 and vice versa indicated by the two underlined 0, this making the two columns identical making it impossible to isolate these two faults. Usually the rpm-sensor s_1 is very reliable. Therefore can the fault code for these two columns be assumed indicating a manifold leakage. As we now only have 4 faults left to diagnose, we only need 4 residuals. Removing the columns for a_2, s_1 and M_{load} and residuals r_2 and r_3 results in the reduced coding set in table 7.3 that is a strongly isolating coding set.

The time window, s, is chosen as in section 4 ($D \neq 0$) to $s = n = 2$. Matlab code to generate the first residual r_1, insensitive to load disturbances and faults in the rpm-sensor, can be written as

```
Q = [[D;C*B;C*A*B], [zeros(size(D));D;C*B], [zeros(size([D;C*B]));D]];
T = [[zeros(3,1);C*E;C*A*E], [zeros(3,1);zeros(3,1);C*E],
     [zeros(size([zeros(3,1);C*E]));zeros(3,1)]];
R = [C;C*A;C*A*A];
%%%%% Decoupling, d1 + actuator1 faults
Qtilde = [[D(:,1);C*B(:,1);C*A*B(:,1)], [zeros(size(D(:,1)));D(:,1);C*B(:,1)],
          [zeros(size([D(:,1);C*B(:,1)]));D(:,1)]];
Z = zeros(7,9);
Z(1:2,:) = R';
Z(3:4,:) = T(:,1:2)';
Z(5:7,:) = Qtilde(:,1:3)';
w_temp = Z(:,[1:4,6:7,9])\(-Z(:,5)-5*Z(:,8));
w1 = [w_temp(1:4);1;w_temp(5:6);5;w_temp(7)];
```

Residuals r_4, r_5 and r_6 are generated with similar code. This residual generator can now be tested, first by simulating faults on the linearized system to see ideal behavior. Figure 7.1 illustrates the simulation. Note how the step in load ($\approx$ uphill) affects the speed at $t = 2$. The lowest plot, the α-plot, illustrates how the assumed throttle angle is 28° but at $t = 5$ a 3° fault happens as indicated by the dotted line, also note how this (unwanted) increase in throttle angle affects the crank-shaft speed. Figure 7.2 shows the corresponding residuals. As expected (column 1 in table 7.3) r_4, r_5 and r_6 fires at $t = 5$ while r_1 does not. Note the invariance to the M_{load} step at $t = 2$.

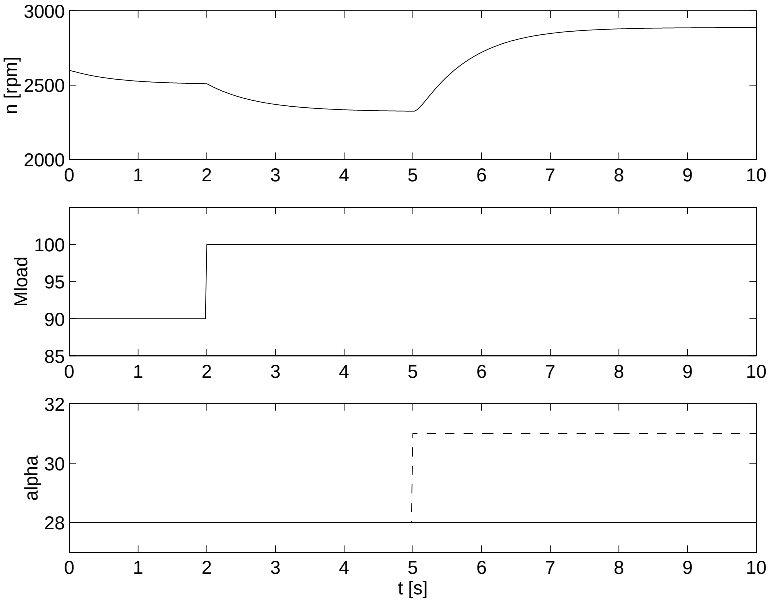

Figure 7.1. Linear throttle fault simulation

Performing a similar simulation on the full non-linear model instead, with M_{load} variations as in figure 7.3 and a 10% step fault in the throttle actuator at $t = 8$ results in the solid lines in figure 7.4.

Note how the system state now deviates from the linearization point. The residuals corresponding to the simulation is shown in figure 7.5. The dotted lines represents

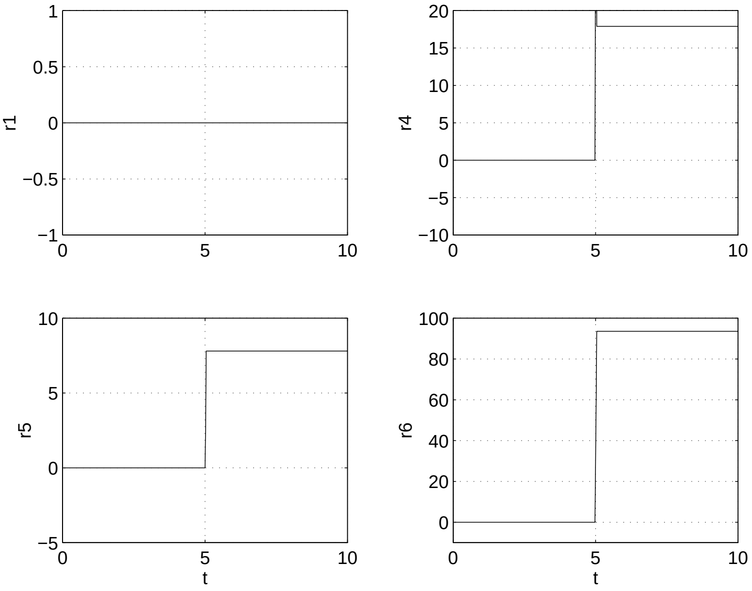

Figure 7.2. Residuals of linear throttle fault simulation

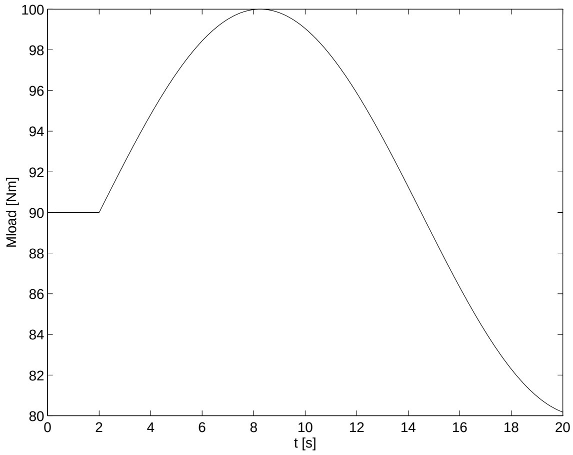

Figure 7.3. M_{load} variations in non-linear simulations

example thresholds chosen from fault-free simulations of the non-linear system around the operating point. We see that residuals r_4, r_5 and r_6 fires while r_1 don't, just as we wanted. Residual behavior is here approximately the same as in the ideal case, somewhat more oscillative. If we instead of a throttle actuator fault simulates a manifold leakage of 10 kg/h at $t = 8$ the system behaves as the dotted lines in figure 7.4. The corresponding residuals are shown in figure 7.6. According to the coding set residuals r_1, r_4 and r_5 should fire while r_6 should not. As we see this is fulfilled but all of residual responses are very near the selected thresholds, r_6 even “dips” below the treshold, indicating the large insecurity in the residuals. Also note that the system deviation from the linearization point is smaller in the second simulation than in the first simulation but still produced more insecure residuals. This is because the system is non-linear and no assumptions on residual “goodness” can be made on the basis of system state.

In this section we have seen that structured parity equations approach function well

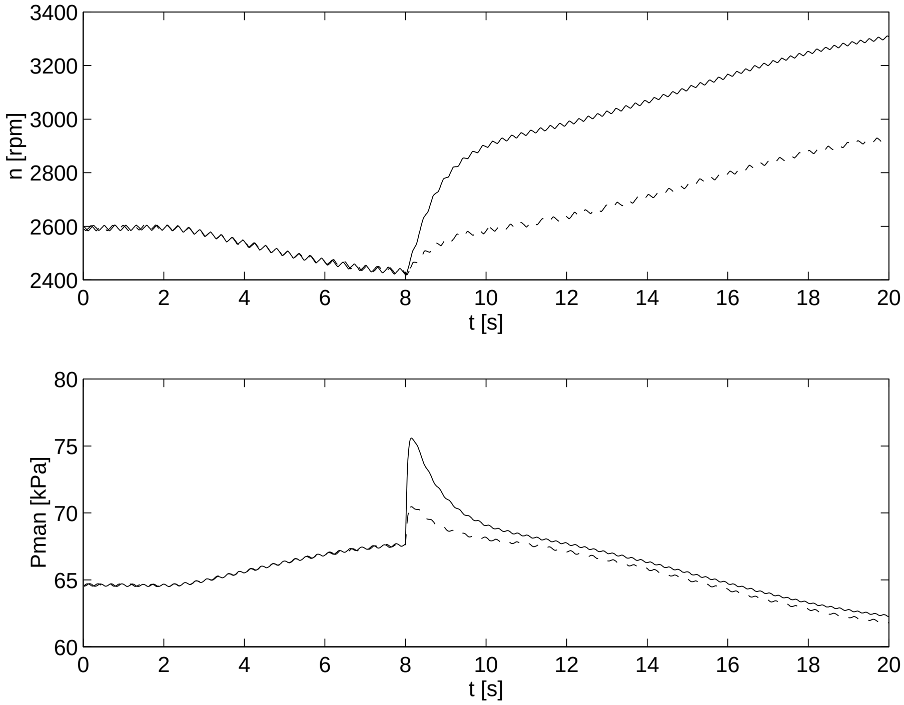

Figure 7.4. Non-linear simulations

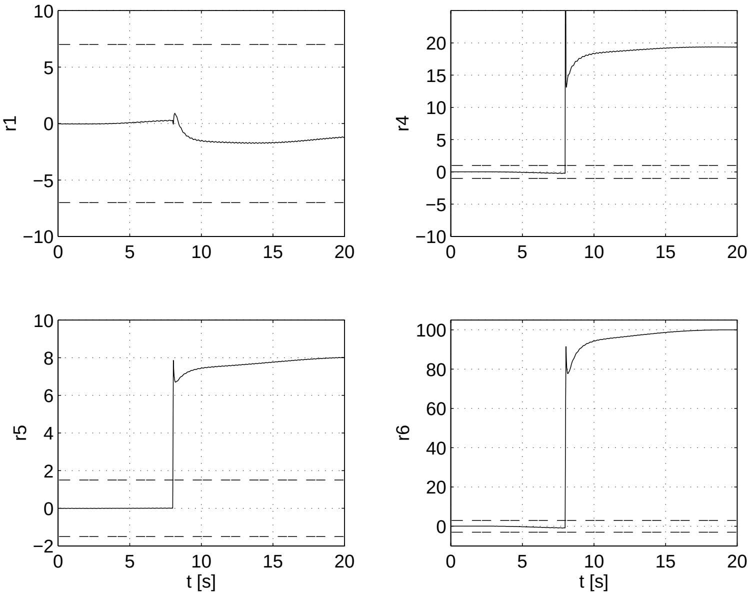

Figure 7.5. Residuals of non-linear throttle fault simulation

in the ideal linear case, but runs into severe problem when applied to the non-linear system. The solution seems to be to use some non-linear residual generators as discussed in section 3.2.7. But the approach described in this section is suitable to a system that is linear, or nearly so.

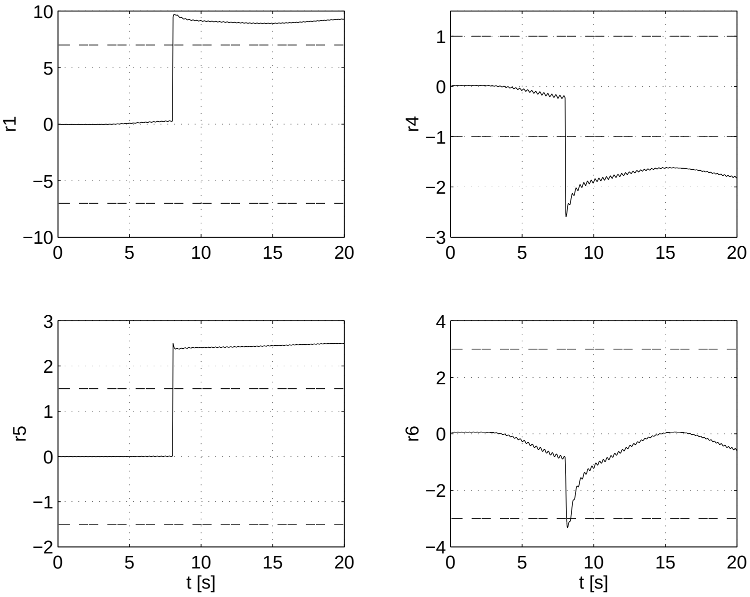

Figure 7.6. Residuals of non-linear manifold leakage fault simulation

7.2 Robust IFD with non-linear observers

This method is based on non-linear observers, creating a IFD-DOS scheme as described in section 3.2.1. The structure is illustrated in figure 7.7. The unmeasurable disturbance,

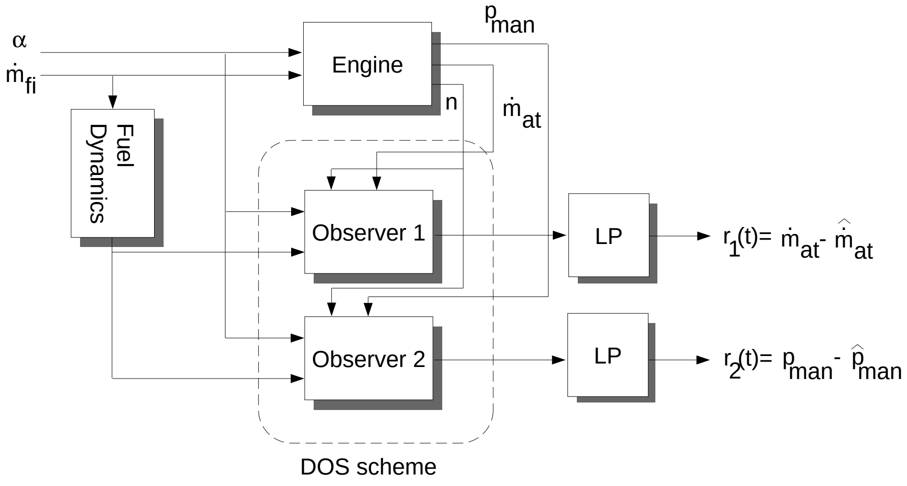

Figure 7.7. Non-linear IFD observer bank

i.e. the road-load, only affects the crankshaft-dynamics directly (see equation 6.1) and since the rpm sensor usually is very reliable we can consider the rpm-sensor output to be correct. With this assumption, also made in [9], we can achieve *robust* IFD, i.e. residuals insensitive to the road-load variations. We design two observers to monitor the two remaining sensors, $\dot{m}_{at}$ and p_{man} as

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x},u) + K_1(\dot{m}_{at} - \hat{\dot{m}}_{at}) \\ \hat{\dot{m}}_{at} &= h_1(\hat{x},u)\end{aligned}$$

and

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x},u) + K_2(p_{man} - \hat{p}_{man}) \\ \hat{p}_{man} &= h_2(\hat{x},u)\end{aligned}$$

Note how the first observer is only fed with $\dot{m}_{at}$ observations and the second only with p_{man} indicating that the first observer estimates states independently of any eventual faults in the p_{man} sensor and vice-versa.

The observer gains were calculated from a linearization of the non-linear model around the same point as in the previous section indicated by table 7.1. Since the observer uses the same model as used to simulate the engine, disturbances has to be introduced to see how it reacts in a non-ideal situation. Here white noise has been added to the sensor outputs. It is difficult to estimate the SNR (Signal Noise Ratio) in the lab-environment so the noise power has been chosen to make the signals "look" like the pre-filtered real measurements.

As in the previous section we assume fuel dynamics in perfect accordance with the model, hence the fuel-dynamics module in figure 7.7.

In all of the simulated experiments in this section a throttle step is made from $\alpha = 28°$ to $\alpha = 31°$ at time $t = 1$. $\dot{m}_{fi}$ is controlled so that $\lambda = 1$. The road-load varies according to figure 7.8. Figure 7.9 presents a simulation of a 10% fault in the p_{man}-sensor at $t = 4$.

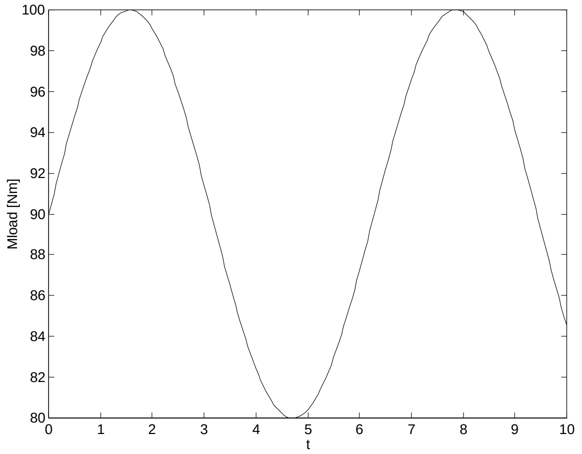

Figure 7.8. M_{load} variations

The dotted residual r_2 fires compared to the suggested threshold while the solid residual r_1 remains near 0. The throttle step at $t = 1$ is clearly visible in the rpm-plot, there is a distinct speed increase at $t = 1$. This step however does not show in the residuals.

If we instead simulates a 10% fault in the $\dot{m}_{at}$-sensor we get system behavior as in figure 7.10. Here the same throttle step is seen not to influence the residuals. The fault at $t = 4$ however influences the *solid* residual r_1, i.e. the other residual than in figure 7.9. These two residual generators are based upon a functioning and reliable speed sensor, if the speed sensor should be affected by a fault, the isolation properties of r_1 and r_2 are cancelled as shown in figure 7.11 where the rpm-sensor has been subjected to a 10% fault at time $t = 4$. As seen, both residuals show non-zero behavior even when both p_{man} and $\dot{m}_{at}$ sensor are fault free.

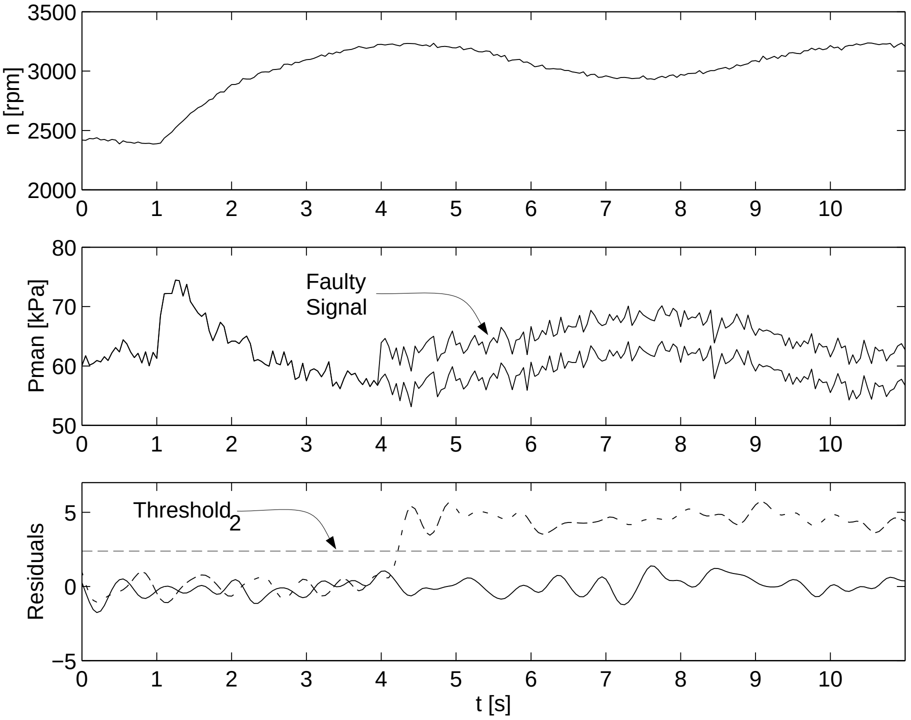

Figure 7.9. Simulation of p_{man} sensor fault

As seen in this section the approach described handles sensor faults well, and since the full non-linear model is used, an improved model directly results in more reliable diagnosis. But this under the condition that no faults in either rpm-sensor or actuators are present. This implies that this approach can be used, but not stand alone, the actuators need to be diagnosed to be able to rely on the sensor diagnosis.

7.3 Eigenstructure Diagnosis

Here we assume we want to diagnose all faults, i.e. even the rpm-sensor and therefore we need disturbance decoupling. In section 5.4 it was noted that the inequality

$$\text{rank}(E) \leq n - p$$

has to be fulfilled to be able to solve the decoupling problem. Here we want to design a GOS scheme, therefore we set the residual dimension $p = 1$. rank(E) is the number of independent signals we want to decouple, here rank$(E) = 2$ since we want to decouple the road load disturbance and one fault. The model derived in chapter 6 is a three state model, but since no information about the fuel-dynamics state can be inferred from any of the measurements there is no gain in using a three state model. We are limited to a two-state model, i.e. $n = 2$. Examining the limit inequality again we see that with this model the inequality can not be fulfilled, i.e. the eigenstructure can not be used with this model. However if we were to develop a better model, or a sensor revealing information about the fuel-dynamics state the eigenstructure assignment could very well be used. Also if we made the same assumptions regarding the rpm-sensor as earlier a eigenstructure approach might be feasible.

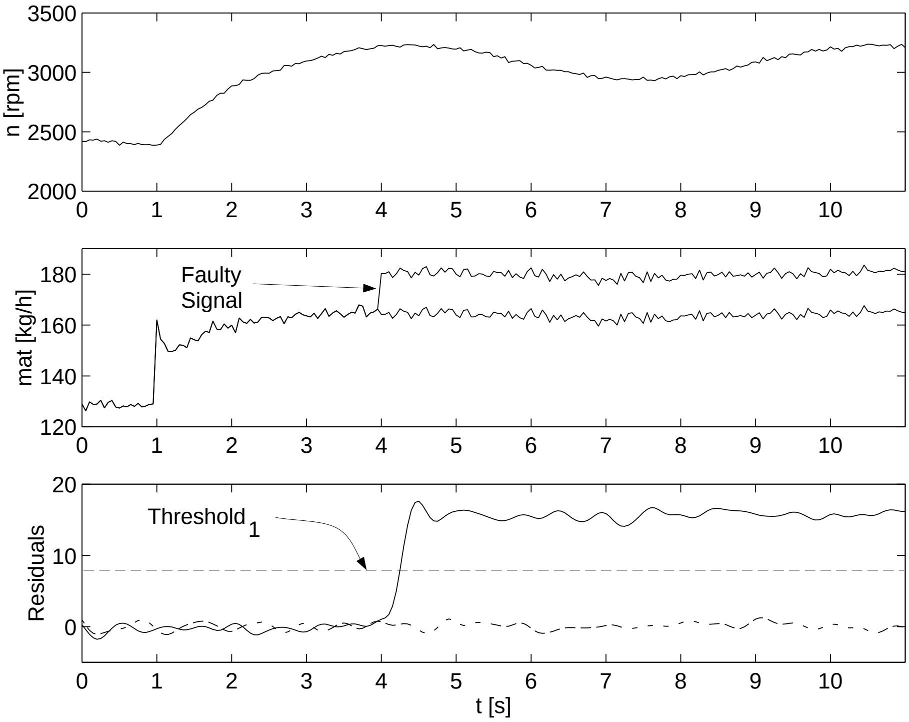

Figure 7.10. Simulation of $\dot{m}_{at}$ sensor fault

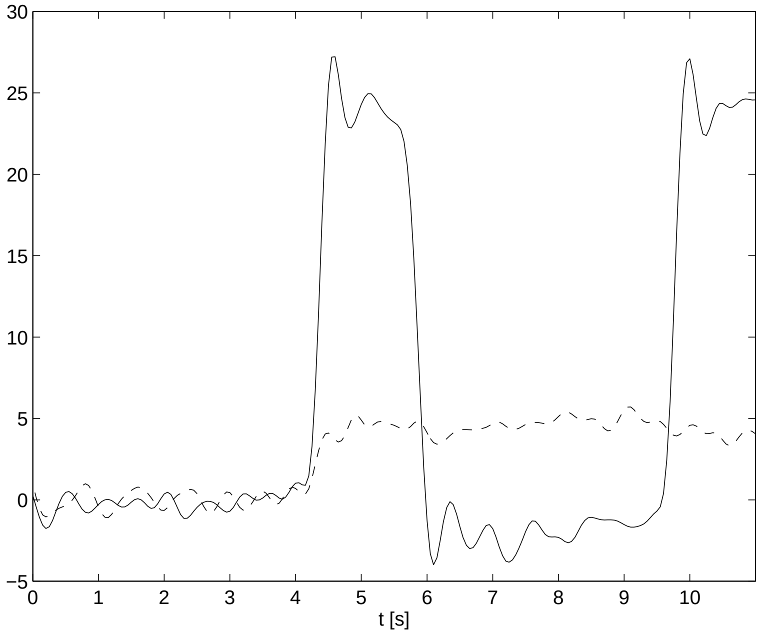

Figure 7.11. Residuals of rpm-sensor fault

Chapter 8

Conclusions and extensions

8.1 Conclusions

In this report it was noted that with the OBD-II regulation, diagnosis will become increasingly important for automotive engines in the next few years. It was also concluded that some form of model-based diagnosis is crucial for the success of a modern high-performance diagnosis system.

The mathematical model used in this thesis is a small mean-value model derived for the complete SI-engine system. The modelling work resulted in a 4-state nonlinear model. This model, although quite small, performs like an engine in a very wide operating range and is well suited for diagnosis experiments.

A literature survey was made and a number of methods to perform diagnosis are examined. Many of them are well suited for diagnosing faults in linear systems. Three methods are chosen for further detailed analysis.

- Parity equations from a state-space model
- Eigenstructure observer
- Non-linear IFD observer

Two of the chosen methods are linear, one open- and one closed-loop, and one non-linear. The linear methods both have the ability to achieve insensitivity to the unmeasured M_{load}. However the methods are linear and the SI-engine is a highly non-linear system. From simulations it was shown that the linear methods with designs based on linearization of the non-linear system run into severe problems when subjected to the full non-linear process. The simulations also show that it was not possible to determine a reliability measure of the diagnosis based on process state. Intuitively, a reliability measure of the linear diagnosis would be how far the process state has deviated from the linearization point. But due to the non-linear process, no such conclusions can be made.

The third non-linear method could achieve the same M_{load} insensitivity under the assumption that the rpm-sensor is fault-free, a quite realistic assumption as the rpm-sensor on a real automotive engine is highly reliable. This approach works well in the whole operating range but a major limitation with this approach is that it is only able to diagnose sensor faults. It could however be used as a part of a complete diagnosis system.

A general approach to the non-linear problem is to use a diagnosis method that relies on a non-linear model, just as the third method. There exists non-linear approaches to FDI, but no general theory exists and therefore more research has to be done investigating the non-linear problem.

8.2 Extensions

This report has given an overall survey of the diagnostic field with emphasis on linear residual generators. To be able to design a complete diagnostic system, a more deep understanding of the linear but more importantly the non-linear field is required.

Important areas that should be investigated further are

- Linear residual generators
- Non-linear residual generators
- Residual evaluation
- Integrating Knowledge based and Control Engineering approaches

Linear residual generators

In this thesis the eigenstructure observer approach were investigated. However with the UIO, [7], more design freedom exists as the observer need not be an identity observer. How this extra freedom influences the diagnosis problem ought to be investigated further.

Other extensions in the linear field can include methods based on stable factorization of transfer functions [33], multiple fault diagnosis [16] or an approach where diagnostics are seen as as a one-step procedure instead of the 2 stage approach used here [2].

Non-linear residual generators

As noted in the report, linear residual generators generally performs poorly on non-linear processes. Methods of designing residual generators for non-linear processes are highly desirable. Methods that ought to be analyzed are for example feedback-linearization, non-linear parity equations [9, 24] or a gain scheduling approach.

Also parameter estimation methods might be a way of handling non-linear processes.

Residual evaluation

In this report, the residual evaluation step has only been a simple threshold test of the residual. A further investigation of residual evaluation methods to increase the robustness of the diagnosis procedure are desirable. E.g. GLR, *Marginalized Likelihood Ratio*(MLR)[13] and approaches based on fuzzy logic [33] can be of importance.

Integrating Knowledge based and Control Engineering approaches

There are advantages with both knowledge based approaches and approaches based on control engineering. Ways to integrate methods from both fields to utilize their respective advantages should be investigated as they might result in more reliable diagnosis procedures.

References

[1] D. Barschdorff. Comparison of neural and classical decision algorithms. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 409–415, Baden-Baden, Germany, 1991.

[2] G. Bloch, D. Theillol, and P. Thomas. Simultaneous detection, location and identification of faults for dynamic systems. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 47–51, Espoo, Finland, 1994.

[3] California's OBD II regulation (section 1968.1, title 13, california code of regulations), resolution 93-40, july 9. pages 220.7 – 220.12(h), 1993.

[4] J. Chen and R.J Patton. A re-examination of fault detectability and isolability in linear dynamic systems. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 567–573, Espoo, Finland, 1994.

[5] M.H. Costin. On-board diagnostics of vehicle emission system components: Review of upcoming government regulation. In *IFAC Fault Detection, Supervision and Safety for technical processes*, pages 497–601, Baden-Baden, Germany, 1991.

[6] P.M. Frank. Enhancement of robustness in observer-based fault detection. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 99–111, Baden-Baden, Germany, 1991.

[7] P.M. Frank and J. Wünnenberg. *Robust fault diagnosis using unknown input observer schemes*, chapter 3. In Patton et al. [32], 1989.

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

[9] J. Gertler, M. Costin, X. Fang, R. Hira, Z. Kowalalczuk, M. Kunwer, and R. Monajemy. Model based diagnosis for automotive engines - algorithm development and testing on a production vehicle. *IEEE Trans. on Control Systems Technology*, 3(1):61–69, 1995.

[10] J. Gertler, M. Costin, X. Fang, R. Hira, Z. Kowalalczuk, and Q. Luo. Model-based on-board fault detection and diagnosis for automotive engines. *Control Engineering Practice*, 1(1):3–17, 1993.

[11] T. Glad and L. Ljung. *Reglerteknik, Grundläggande teori.* Studentlitteratur, Lund, Sweden, 2nd edition, 1989.

[12] B. Gudmundsson and T. Hallberg. Feltoleranta digitala system. Kompendium ISY, 1992.

[13] Fredrik Gustafsson. The marginalized likelihood ratio test for detecting abrupt changes. *IEEE Transactions on automatic control*, 41(1):66–78, January 1996.

[14] E. Hendricks. Mean value modelling of spark ignition engines. *SAE–Technical Paper Series*, 1(900616), 1990.

[15] John B. Heywood. *Internal Combustion Engine Fundamentals.* McGraw-Hill series in mechanical engineering. McGraw-Hill, 1992.

[16] P.L. Hsu, K.L. Lin, and L.C. Shen. Diagnosis of multiple sensor and actuator failures in automotive engines. *IEEE transactions on vehicular technology*, 44(4):779–789, November 1995.

[17] Formelsamling - grundläggande mekanisk värmeteori och strömmningslära. LiTH, IKP, 581 83 Linköping, Sverige.

[18] R. Isermann. *Process fault diagnosis based on dynamic models and parameter estimation methods*, chapter 7. In Patton et al. [32], 1989.

[19] R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 43–55, Baden-Baden, Germany, 1991.

[20] R. Isermann. Integration of fault detection and diagnosis methods. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 575–590, Espoo, Finland, 1994.

[21] J.A. Jeyes. Diagnostics - the potential for improving vehicle safety and reducing pollution. In *Proceedings of the IMechE - Automotive Diagnostics*, pages 65–76, Birdcage Walk, London, November 1990.

[22] R. Jurgen. *Automotive electronics handbook.* McGraw Hill, 1995.

[23] V. Krishnaswami, G.C. Luh, and G. Rizzoni. Fault detection in IC engines using nonlinear parity equations. In *Proceedings of the American Control Conference*, pages 1581–1584, Baltimore, Maryland, 1994.

[24] V. Krishnaswami and G. Rizzoni. Non-linear parity equation residual generation for fault detection and isolation. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 305–310, Espoo, Finland, 1994.

[25] S.H.Y. Lai. Engine system diagnosis using vibration data. *Computers and Industrial Engineering*, 25(1-4):135–138, 1993.

[26] G.C. Luh and G. Rizzoni. Identification of a nonlinear MIMO IC engine model during I/M240 driving cycle for on-board diagnosis. In *Proceedings of the American Control Conference*, pages 1581–1584, Baltimore, Maryland, 1994.

[27] The MathWorks Inc. *Matlab - User's guide*, 1992.

[28] The MathWorks Inc. *Simulink - User's guide*, 1992.

[29] G.F. Mauer. On-line performance diagnostics for internal combustion engines. In *Int. Conf. on Ind. Electronics, Control, Instrumentation and Automation*, volume 3, pages 1460–1465, 1992.

[30] D. Neumann. Fault diagnosis of machine-tools by estimation of signal spectra. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 147–152, Baden-Baden, Germany, 1991.

[31] P.M. Olin and G. Rizzoni. Robust fault detection. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 259–264, Baden-Baden, Germany, 1991.

[32] R. Patton, P. Frank, and R. Clark, editors. *Fault diagnosis in Dynamic systems.* Systems and Control Engineering. Prentice Hall, 1989.

[33] R.J Patton. Robust model-based fault diagnosis:the state of the art. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 1–24, Espoo, Finland, 1994.

[34] R.J. Patton and J. Chen. Optimal selection of the unknown input distribution matrix in the design of robust observer for fault diagnosis. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 229–234, Baden-Baden, Germany, 1991.

[35] R.J Patton and J. Chen. A review of parity space approaches to fault diagnosis. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 65–81, Baden-Baden, Germany, 1991.

[36] R.J Patton and S.M. Kangethe. *Robust fault diagnosis using eigenstructure assignment of observers*, chapter 4. In Patton et al. [32], 1989.

[37] Andrej Perkovic and Patrik Berggren. Cylinder individual lambda feedback control in a SI engine. Master's thesis, Vehicular Systems, Linköpings University, 1996.

[38] K. Peter and R. Isermann. Parameter-adaptive pid-control based on continuous-time process models. In *Adaptive Systems in Control and Signal Processing*, pages 241–246, Glasgow, UK, 1989. IFAC.

[39] E. Rich and K. Knight. *Artificial Intelligence, 2nd ed.* McGraw-Hill Inc., 1991.

[40] G. Rizzoni, P.M. Azzoni, and G. Minelli. On-board diagnosis of emission control system malfunctions in electronically controlled spark ignition engines. *Proc. of the American Control Conference*, pages 1790–1795, 1993.

[41] G. Rizzoni and P.S. Min. Detection of sensor failures in automotive engines. *IEEE Trans. on Vehicular Technology*, 40(2):487–500, 1991.

[42] T. Sorsa and H.N. Koivo. Application of artificial neural networks in process fault diagnosis. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 423–428, Baden-Baden, Germany, 1991.

[43] R. Stobart. Observers in engine control systems: What is the potential? *Colloquium on Automotive Applications of Advanced Modelling and Control*, pages 5/1–5/3, 1994.

[44] S.G. Tzafestas. *System fault diagnosis using the knowledge-based methodology*, chapter 15. In Patton et al. [32], 1989.

[45] A. Unger and K. Smith. The OBD II system in the Volvo 850 turbo. 1(932665), 1993.

[46] B.K. Walker. *Fault detection threshold determination using Markov theory*, chapter 14. In Patton et al. [32], 1989.

Appendix A: Laboratory Facility and Engine Specifications

This appendix will describe the laboratory equipment, at the Division of Vehicular Systems at Linköping University, including engine, dynamometer,measurement equipment and computers.

Engine Specifications

The engine is a SAAB 2.3 L spark ignition standard engine equipped with extra sensors for measurement of in cylinder pressure, ionization currents, air mass flow etc. The electronic control unit is called Selma and is developed by Mecel AB. It controls, among many other things, the fuel injectors and spark advance of each cylinder of the engine. Selma is equipped with a CAN–bus[1] interface making it possible to send and receive information during engine tests. The general specifications of the engine are:

Engine type:	4 cylinder, four stroke, 16 valve engine with double overhead camshafts and double balance shafts.
Displacement:	2.3 liters (2290 cm^3).
Bore:	90 mm.
Stroke:	90 mm.
Firing order:	1–3–4–2.
Maximum engine power:	150 bhp (110 kW).
Maximum engine torque:	212 Nm.
Weight:	≈ 160 kg.
Serial number:	B2341.4N10M219569.

Dynamometer

To simulate different driving conditions, engine speed and engine load, we need a dynamometer (brake). There are different types of brakes and the one used in the laboratory is a Dynasyn NT 85 servo motor/generator from Schenck. It can operate under conditions up to engine torques of 150 Nm.

Computers

Besides Selma we use three standard PCs when the engine is running. All three computers and Selma communicate via the CAN–bus, see Figure A.1. The first computer, Hillman, contains a real time system using RTKernel software. It's on this computer the different controllers are executed, e.g. a crank-shaft speed controller. From the second computer, Minx, reference values for throttle angle and engine speed can be set with help of a graphical user interface with slide bars.

[1] Controller Area Network

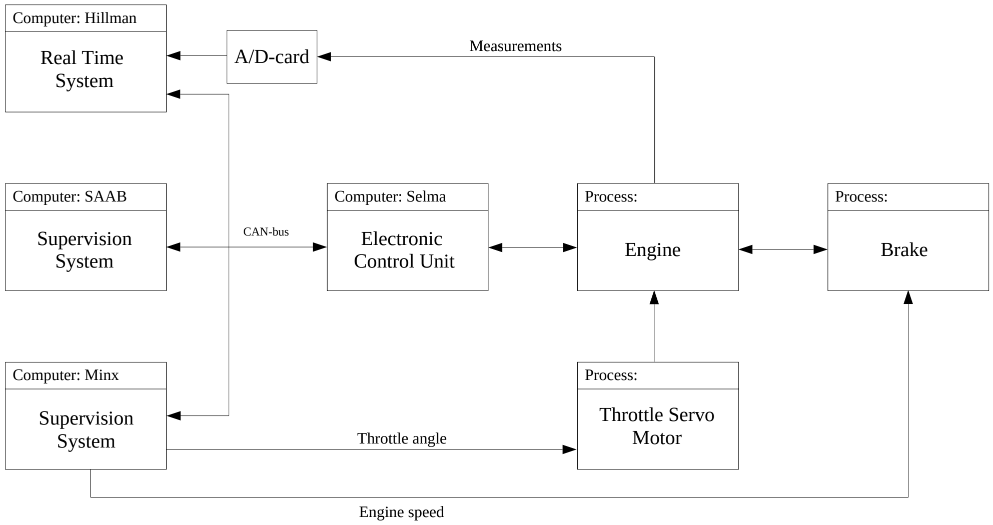

Figure A.1. Hardware setup.

Sensors

5 sensors are used in this report and the measured values enter Hillman via a 12 bit A/D conversion card (type RTI–815 from Analog Devices) with a resolution of 4.88 mV as indicated in figure A.1. The 5 sensors are:

- **rpm sensor**
  An rpm-sensor was built before any engine-speed measurements could be done by Erik Frisk and Mattias Nyberg.
- **Intake manifold pressure sensor**
  The pressure sensor used is a KRISTAL pressure transmitter 4285A2
- **Torque sensor**
  The torque sensor used is a Hottinger Baldwin Messtechnik GMBH of type MBL 40
- **Air mass-flow sensor**
  A BOSCH Hot-Film Air-Mass Sensor of type HFM2C - 4.7 is used
- **λ-sensor**
  The sensor for measuring lambda are either the standard EGO sensor located 80 cm downstream the exhaust manifold or an UEGO sensor of type TL–7111–W1 with electronic controller TC–6000 from NGK. The UEGO sensor is located at the same position as the standard EGO sensor.

Appendix B: Simulink Implementations

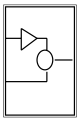

Figure B.1. Subsystem

Engine Model

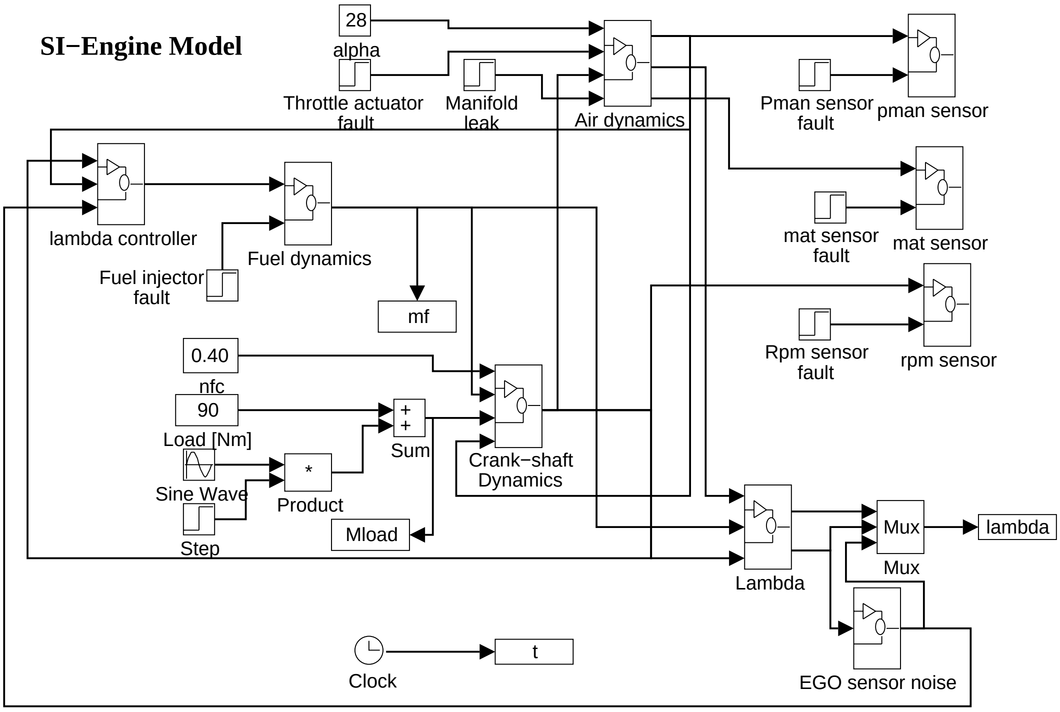

Figure B.2. SI-Engine Model

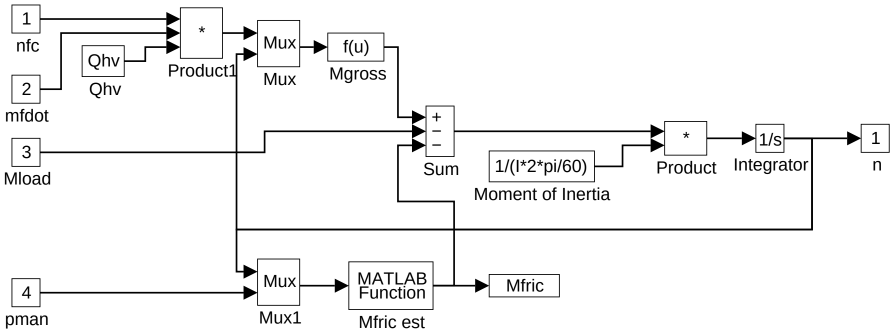

Figure B.3. Crank-shaft Dynamics

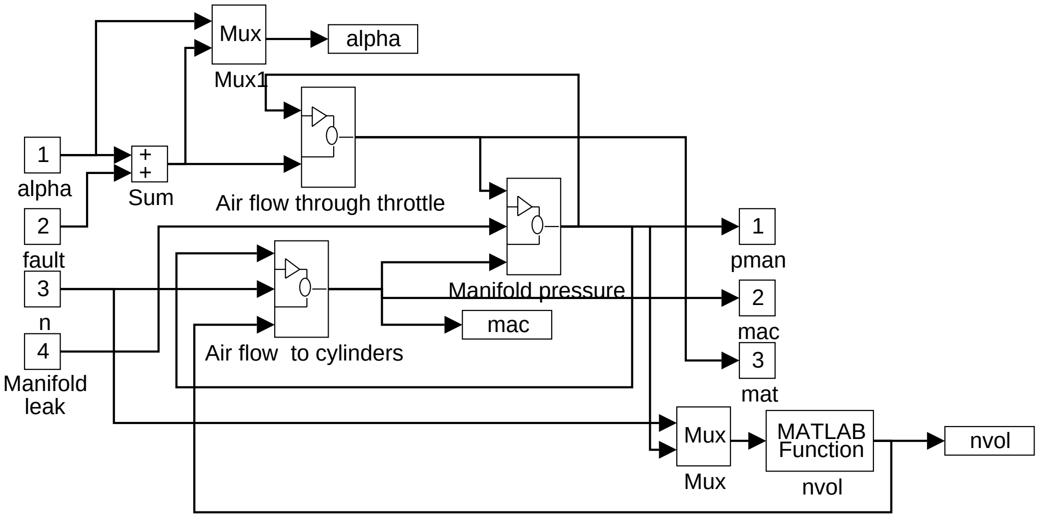

Figure B.4. Air Dynamics

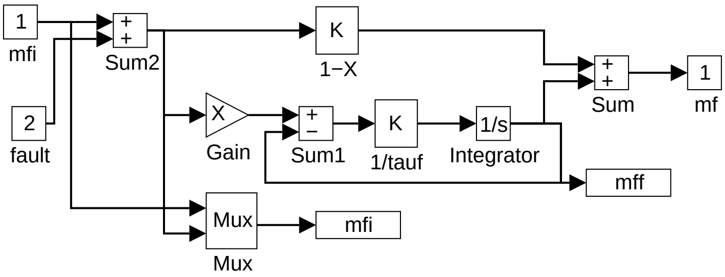

Figure B.5. Fuel dynamics

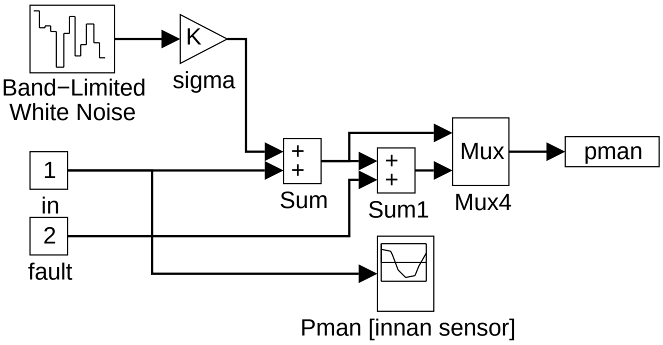

Figure B.6. Pman sensor

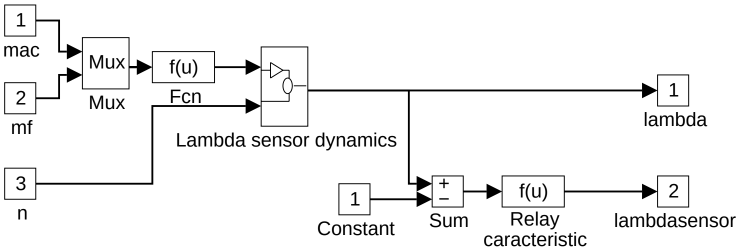

Figure B.7. Lambda

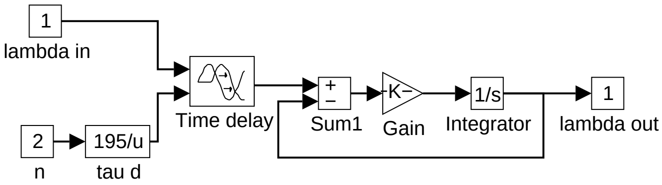

Figure B.8. Lambda sensor dynamics

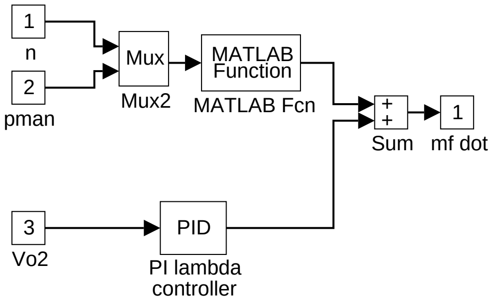

Figure B.9. Lambda controller

IFD Observers

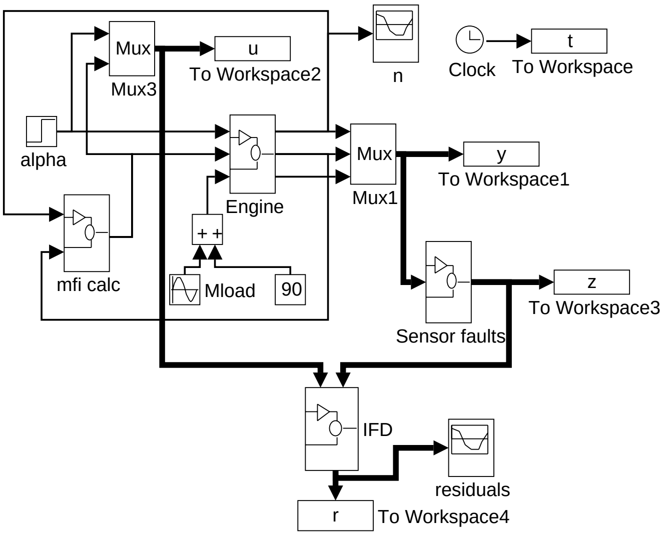

Figure B.10. IFD observers

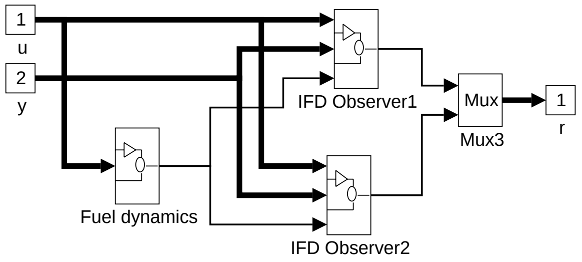

Figure B.11. IFD

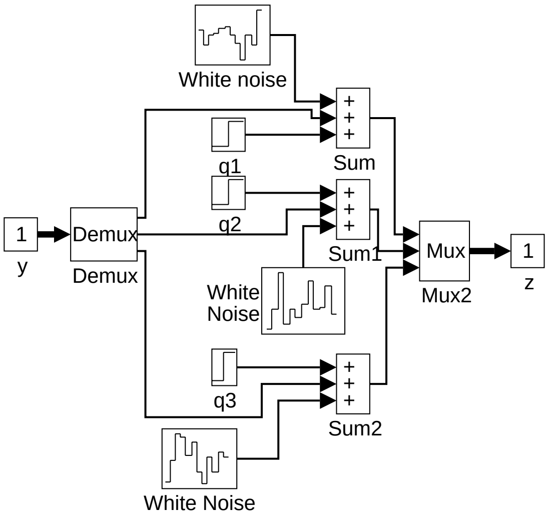

Figure B.12. Sensor faults

Appendix C: Mathematical definitions

Matrix rank

Let $A \in \mathbf{R}^{m \times n}$.

- The row rank of A is the number of linearly independent rows in A
- The column rank of A is the number of linearly independent columns in A

It can be shown that row rank = column rank and is denoted $\text{rank}(A)$. Matrix A is said to have *full rank* if $\text{rank}(A) = \min(m, n)$.

Left eigenvectors

A vector l is called a *left* eigenvector to matrix A if there exists a β so that

$$l^T A = \beta l^T$$

The constant β is called the corresponding eigenvalue.

An eigenvector v that multiplies with A from right, i.e.

$$Av = \beta v$$

is called a right eigenvector or just eigenvector.

Left eigenvector l to matrix A is right eigenvector to matrix A^T as

$$\begin{aligned} & (l^T A)^T = (\beta l^T)^T \\ \Rightarrow \quad & A^T l = \beta l \end{aligned}$$

Diagonalizable matrices

A matrix A is said to be diagonalizable if it can be written

$$A = TDT^{-1}$$

Where D is a diagonal matrix and T is the transformation matrix. If the eigenvectors of a matrix is *linearly independent* the matrix is diagonalizable. A set of eigenvectors corresponding to different eigenvalues are linearly independent.

Singular Value Decomposition (SVD)

If $Z \in \mathbf{R}^{m \times n}$, then there exists *orthogonal* matrices

$$\begin{aligned} U &\in \mathbf{R}^{m \times m} \\ W &\in \mathbf{R}^{n \times n} \end{aligned}$$

such that

$$\begin{aligned} Z &= UDW^T \\ D &= \text{diag}(\sigma_1, \ldots, \sigma_q) \end{aligned}$$

where $q = \min(n, m)$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_q$. The values $\sigma_1, \ldots, \sigma_q$ is called the *singular values.*

If $m < n$, the matrix D becomes

$$D = \left(\begin{array}{cccc|c} \multicolumn{4}{c}{\overbrace{\hphantom{\sigma_1 \quad \sigma_2 \quad \ddots \quad \sigma_m}}^{m}} & \overbrace{\hphantom{0}}^{n-m} \\ \sigma_1 & & & & 0 \\ & \sigma_2 & & & 0 \\ & & \ddots & & \vdots \\ & & & \sigma_m & 0 \end{array} \right)$$

For example, if

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \end{pmatrix}$$

the SVD becomes

$$U = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1.4142 & 0 \end{pmatrix} W = \begin{pmatrix} 0 & 0.7071 & -0.7071 \\ 1 & 0 & 0 \\ 0 & 0.7071 & 0.7071 \end{pmatrix}$$

Frobenius norm

The Frobenius norm is defined as taking the square root out of the squared sum of all matrix elements, i.e.

$$A \in \mathbf{R}^{n \times m}, ||A||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^2}$$

Or equivalently

$$||A||_F = \sqrt{\text{trace}(A^T A)}$$

Appendix D: Matlab measurement script

```
function [mat1, pman1, n1, mat2, pman2, n2, actualRate] =
  measure( nMin, nMax, pMin, pMax, numWPoints, fs );
% nMin        - Minimum rpm               [rpm]
% nMax        - Maximum rpm               [rpm]
% pMin        - Minimum manifold pressure [kPa]
% pMax        - Maximum manifold pressure [kPa]
% numWPoints  - Number of WP
% fs          - Sampling frequency        [Hz]

if nargin ~= 6
  disp('Wrong number of arguments');
  break;
end

hillman('exec "engine.ini"');
hillman('start brake');
hillman('make r1(PIpman)');
hillman('r1.yport=4');
hillman('r1.uport=0');
hillman('r1.Ti=5');
hillman('r1.Tt=5');
hillman('start r1');

nRange    = [nMin:(nMax-nMin)/(numWPoints-1):nMax];
pManRange = [pMin:(pMax-pMin)/(numWPoints-1):pMax];
mat1  = zeros( numWPoints, numWPoints );
pman1 = zeros( numWPoints, numWPoints );
n1    = zeros( numWPoints, numWPoints );
mat2  = zeros( numWPoints, numWPoints );
pman2 = zeros( numWPoints, numWPoints );
n2    = zeros( numWPoints, numWPoints );

[y, t, actualRate] = sample([0], fs, .1);
for i = 1:numWPoints
  for j = 1:numWPoints
    % Enter WP(i,j)
    disp( sprintf('pman = %d, n = %d\n', pManRange(i), nRange(j)) );
    hillman( sprintf('r1.ref = %d', pManRange(i)) );
    hillman( sprintf('port(18) = %d', nRange(j)/1000) );


    % Wait trainsients, delay-kommando!!!
    delay(5);

    hillman( 'suspend r1' );
    delay(15);
    % Measure 1 sec samples and take mean-value
    y = sample( [0 1 2], fs, 1 );

    hillman( 'resume r1' );

    mat1(i, j)  = mean( fsma(y(:,1)) );             % [kg/h]
    pman1(i, j) = mean( fspman(y(:,2)) )*100;       % [kPa]
    n1(i, j)    = mean( fsw(y(:,3)) ).*(60/(2*pi)); % [rpm]
  end;
end;

for i = numWPoints:-1:1
  for j = numWPoints:-1:1
    % Enter WP(i,j)
    disp( sprintf('pman = %d, n = %d\n', pManRange(i), nRange(j)) );
    hillman( sprintf('r1.ref = %d', pManRange(i)) );
    hillman( sprintf('port(18) = %d', nRange(j)/1000) );


    % Wait trainsients, delay-kommando!!!
    delay(5);

    hillman( 'suspend r1' );

    delay(15);
    % Measure 1 sec samples and take mean-value
    y = sample( [0 1 2], fs, 1 );

    hillman( 'resume r1' );

    mat2(i,j)  = mean( fsma(y(:,1)) );
    pman2(i,j) = mean( fspman(y(:,2)) )*100;
    n2(i,j)    = mean( fsw(y(:,3))).*(60/(2*pi) );
  end;
end;


hillman( 'killall' );
```