BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING
FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

PREDICTIVE MAINTENANCE OF PNEUMATIC PISTONS
MOŽNOSTI PREDIKTIVNÍ ÚDRŽBY PNEUMATICKÝCH PÍSTŮ

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR Bc. Artyom Voronin
AUTOR PRÁCE

SUPERVISOR Ing. Martin Brablc
VEDOUCÍ PRÁCE

BRNO 2021

Assignment Master's Thesis

Institut:	Institute of Solid Mechanics, Mechatronics and Biomechanics
Student:	**Bc. Artyom Voronin**
Degree programm:	Applied Sciences in Engineering
Branch:	Mechatronics
Supervisor:	**Ing. Martin Brablc**
Academic year:	2020/21

As provided for by the Act No. 111/98 Coll. on higher education institutions and the BUT Study and Examination Regulations, the director of the Institute hereby assigns the following topic of Master's Thesis:

Predictive maintenance of pneumatic pistons

Brief Description:

With the ever–increasing degree of automation in the industry, a widespread effort to measure, record, and exploit information and signals related to the state of a given machine and its production quality, is becoming more relevant. Predictive Maintenance (PM) is a relatively new method, which builds on and further expands the ideas of the already established Fault Detection and Analysis (FDA). The purpose of this work is to demonstrate various approaches to Predictive Maintenance (e.g., signal–based and model–based) using the Matlab/Simulink software tools on a double–acting pneumatic piston as a case–study.

Master's Thesis goals:

1. Conduct research in the area of Predictive Maintenance, Fault Detection and Analysis, and related approaches and try to define their similarities and differences. Provide a practical demonstration for each of the approaches.
2. Create a simulation model of the demonstration device, including models of the sensors. Test different methods to create the model (e.g., software simulation, physical properties, black–box identification, etc.) and identify the models with real data.
3. Apply Predictive Maintenance techniques to a test dataset without using a simulation model.
4. Apply Predictive Maintenance techniques to a test dataset using a simulation model.
5. Evaluate the suitability of each approach for the application of PM and FDA.

Recommended bibliography:

PRITCHARD, Philip J. Introduction to Fluid Mechanics, 9th edition, Wiley, ISBN 978-1118921876.

NELLES, Oliver. Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Berlin: Springer, 2011. ISBN 978-364-2086-748.

LJUNG, Lennart. System identification: theory for the user. 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 1999. ISBN 978-0136566953.

VALÁŠEK, Michael. Mechatronika. Dot. 1. vyd. Praha: České vysoké učení technické, 1996. ISBN 80-010-1276-X.

NOSKIEVIČ, Petr. Modelování a identifikace systémů. Ostrava: Montanex, 1999. ISBN 80-722-50-0-2.

Deadline for submission Master's Thesis is given by the Schedule of the Academic year 2020/21

In Brno,

L. S.

prof. Ing. Jindřich Petruška, CSc. Director of the Institute	doc. Ing. Jaroslav Katolický, Ph.D. FME dean

Abstract

Tato práce se zabývá vytvořením simulačního modelu dvojčinného pneumatického pístu s mechanickou sestavou, včetně modelů snímačů, s následujícím odhadem parametrů a aproximací chování demonstračního zařízení. Dalším cílem je prezentace různých přístupů prediktivní údržby na datové sadě měřené na demonstračním zařízení. Na měřený datový soubor se aplikovaly signal-based techniky bez použití simulačního modelu a model-based metody, které vyžadují použití simulačního modelu.

Výsledkem této práce je ověření možnosti monitorování stavu zařízení pomocí nainstalovaných senzorů a vyhodnocení efektivity senzorů z hlediska přesnosti a finančních nákladů.

Summary

This thesis deals with creating a simulation model of a double-acting pneumatic piston with a mechanical assembly, including the sensors models, with the following parameter estimation and approximation to the behavior of a demonstration device. Another goal is the demonstration of various Predictive Maintenance approaches on a dataset measured on a demonstration device. Applying signal-based techniques to the measured dataset without using a simulation model and a model-based method that requires the use of a simulation model.

The outcome of this work is the verification of the possibility of monitoring the device's condition state, using installed sensors, and evaluating the efficiency of the sensors in terms of accuracy/cost.

Klíčová slova

dvojčinný pneumatický válec, prediktivní údržba, identifikace a detekce poruch, zbývající doba použitelnosti, PdM, FDI, RUL

Keywords

double-acting pneumatic piston, predictive maintenance, fault detection and identification, remaining useful life, PdM, FDI, RUL

Bibliographic citation

VORONIN, A. *Predictive maintenance of pneumatic pistons* . Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2021. **??** pages, Master's thesis supervisor: Martin Brablc.

Rozšírený abstrakt

Úvod

Od začátku průmyslové revoluce, složitost výrobních strojů a sériových linek se postupně narůstala a tim vyžadovala neustálé monitorování stavu systémů, a to zejména z ekonomických důvodů. Na druhou stranu systémy vyžadujíci vysokou míru bezpečnosti jako letadla, kosmické lodě, automobilové systémy, jaderné reaktory a další vyžadují okamžité spuštění poplašného systému, lokalizování místa chyby a navíc možnost predikce poruchy. Tyto požadavky se staly předpokladem pro vznik identifikace a detekce poruch a prediktivní údržby.

Výrobní proces vždy zahrnoval prvky kontroly chyb a online monitorování. Od prvních metod detekce poruch, například vizuální inspekce, dnešní továrny přecházejí na automatizované systémy skládající se ze senzorů a výpočetní techniky k vyhodnocení poruch. Je potřeba monitorovat zařízení v reálném čase, aby nedošlo k poškození způsobené chybou nebo anomálií. Každá jednotlivá chyba může zapříčinit zpomalení výrobního procesu a tím i snížení zisku.

Algoritmy monitorování zařízení v reálném čase vytvořily Fault Detection and Analysis (FDA). Metody FDA ve většině případech nevyžadují strojové učení a dokáží detekovat poruchy pomocí základních algoritmů jako Fourierova analýza a algoritmy pro kontrolu trendů apod.

Vzhledem k množství údajů nahromaděných v posledních letech a rozšíření technologie ukládání dat jako cloudové služby a výpočetní efectivita, díky nim je možné používat pokročilejší algoritmy pro detekci poruch a analýzu. Pomocí technik klasifikace strojového učení je možné lokalizovat místo chyby. Další možnosti, ktere jsou k dispozici za použití velkého množství dat, je odhad zbývající doby použitelnosti (RUL) celého systému.

Tyto techniky vedly k prediktivni údržbé jakou je snaha optimálizací údržby.

Aktuální technický stav zařízení je vždy k dispozici podle informací extrahovaných z měřených signálů. Je možné použít aktuální stav systému pro odhad zbývající životnosti v jednotkach vzdálenosti nebo času. Odhadovaný zbytek životností dává možnost plánování údržby vhledem ke skutečnému stavu systému.

Tyto algoritmy pro odhad životnosti, detekce poruch, techniky modelování a identifikace systémů tvoří novou oblast prediktivní údržby.

Modelování systému umožňuje provádět experimenty a vyvíjet řešení offline před fyzickou implementacemi v hardwaru. Nedostupné nebo náročné měření lze nahradit generovanými daty ze simulačního modelu a nakonec simulační model pomáhá nasadit robustní algoritmus.

Tato práce poskytuje krátký úvod do detekce poruch a predikce metodiky údržby a základní terminologie. Kapitola 2 popisuje hlavní cíl a problémy v těchto oblastech, zaměřuje se na podobnosti a rozdíly mezi těmito dvěma přístupy.

Vývoj simulačního modelu dvojčinného pneumatického aktuátoru a porovnání s reálným vybavením pomocí různých přístupů je popsán v kapitolách 3, 4 a 5.

Následující kapitola 6 ilustruje prediktivní údržbu založenou na signal-based metodach využívajících různé senzory dostupné v demonstračním zařízení. Aplikace předzpracování, extrakce features a trénování klasifikačního modelu, senzory byly hodnoceny z hlediska funkčnosti, přesnosti a ceny.

Model-based techniky prediktivní údržby založené vyuziti simulačního modelu jsou popsány v kapitole 7. Pomocí simulační modelu lze určit zbytkové signály mezi naměřenými daty a simulacímí daty z výstupu modelu. Pomocí simulačního modelu lze vygenerovávat údaje o degradaci systemi a použit tyto data k odhadu zbývající životnosti.

Závěr

Cílem této práce bylo představít a ověřit metody detekcí poruch a techniky prediktivní údržby na dvojčinném pneumatickém pístu jako objekt případové studie.

Simulační model

Jedním z výstupů práce je simulační model dvojčinného pneumatického pístového systému postaveného na základě diferenciálních rovnic z pneumaticko-mechanické oblastí, modelováno a vyvíjeno pomocí softwaru Matlab/Simulink. Parametry simulačního modelu byli odhadnuty v nominalním stavu systému. Existuje však možnost odhodou parametry poruchového stavu a simulovat systém při poruše.

Vzhledem k dostupným naměřeným údajům a výrazně nelineární dynamice systému, simulační model vykazuje dobrou shodu s naměřeným daty. Na rozdíl od modelu vytvořeného pomocí knihovny Simulink/Simscape je výrazně méně výpočetně naročný při zachování numerické stability. Tato fakta jsou zásadní, pro odhad parametrů.

Simulační model byl použit k experimentování s chováním systému za různých podmínek, modelování poruchových situací a generování data pro design a vyvoj robustních algoritmů prediktivní údržby.

Signal-based PdM

Dalším výstupem je ověření možnosti klasifikace a detekce poruchového stavu pomocí technik prediktivní údržby, na zakladé signal-based metod.

Pokusy byly prováděny na datové sadě měřené na demonstračním zařízení pomocí osmi typů senzorů.

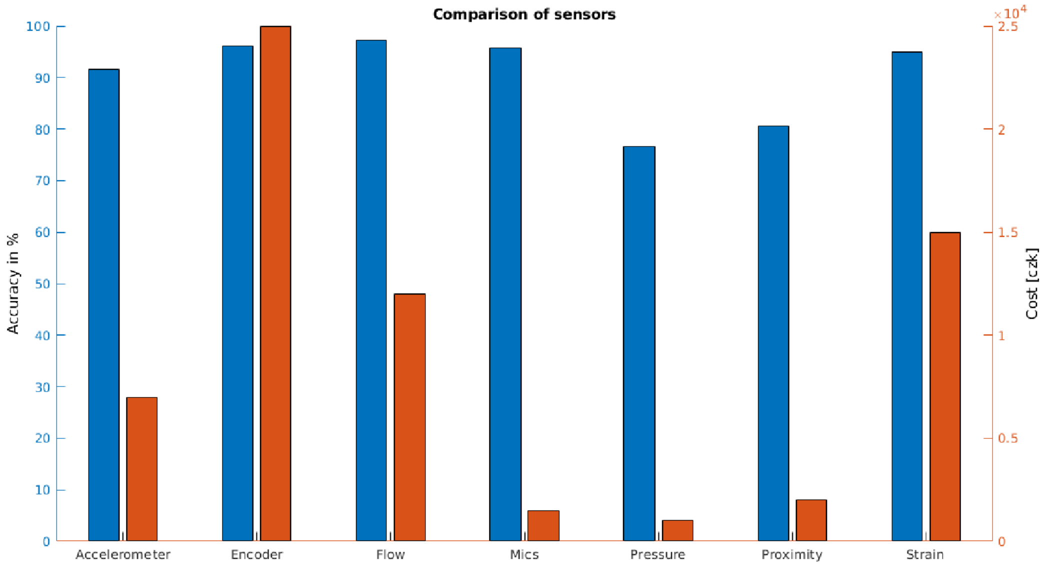
Signal-based metoda je založena na extrakci užitečných informací přímo ze signálu v časově-frekvenčních doménách. Každý senzor vyžadoval individuální přístup k předzpracování, extrahování features, hodnocení vlastnosti a vytváření klasifikačních modelů. Ale obecně lze doporučit minimální předběžné zpracování potřebné k uchování možných užitečných informací.

Tabulka 9.1 obsahuje srovnání senzorů ve dvou kategoriích, přesnost oveřená na testovacím datovém souboru a nákladéch. Graf 9.1 vizualizuje tyto údaje.

Překvapivě všechny senzory vykazovaly přesnost více než 75 %. Mikrofony nabízejí vynikající výkon z hlediska nákladů a přesnosti a jsou vhodné pro instalaci a údržbu.

Sensor	Acc	Encoder	Flow	Mics	Pressure	Proximity	Strain
Accuracy [%]	91.6	96.1	97.2	95.8	76.6	80.5	95.0
Cost [**czk**]	2x 3500	25000	6000	3x 500	1000	2x 1000	15000

Tabulka 1: Comparison of sensors from accuracy/cost perspective



Obrázek 1: Comparison of sensors from accuracy/cost perspective

PdM podle modelu

Další částí této práce byla aplikace model-based metody a využití simulačního modelu pro algoritmy prediktivní údržby. Tyto algoritmy jsou vhodné, pokud je těžké extrahovat užitečné informace přímo ze měřených signálů. V některých případech, pokud rozumíme dynamice systému, umíme využívat některé systémové proměnné jako indikátory stavu.

Extrakce features ve formě nelineárních koeficientu identifikačního modelu určeného z demonstračního zařízení, konkrétně s Hammerstein-Wiener modelem, nedal spolehlivé výsledky. Extrahované features nemají statistickou závislost a je nemožné předvídat typ poruchy použitím této metody na naměřených datech z pneumatického pístu.

Na druhou stranu residual estimation methoda pomocí simulačního modelu ukázala vynikající výsledky. Měřený signál polohy byl porovnán se signálem ze nominalním simulačním modelem. Tento zbytkový signál byl použit ke klasifikaci poruchového stavu a dosáhl 99 % na menší testovací datové sadě. Ale vzhledem k výsledkům získaným pomocí signal-based metody, použití residual estimation se může zdát zbytečná. V tomto konkrétním případé, z praktického hlediska, zlepšení výsledeku o několik procent nepřináší zásadní změny, ale doba výpočtu se významně zvyšuje.

Také byla ověřena možnost modelováni a simulace poruch senzorů pomocí simulačního modelu. Ve většíně připadéch je náročné sbírat chybová data způsobenými senzory vreálných podmínkách. Proto mohou být použity generováne data ze simulačního modelu a pří kombinaci s původní datovou sadou mohou vytvořit syntetický datový soubor.

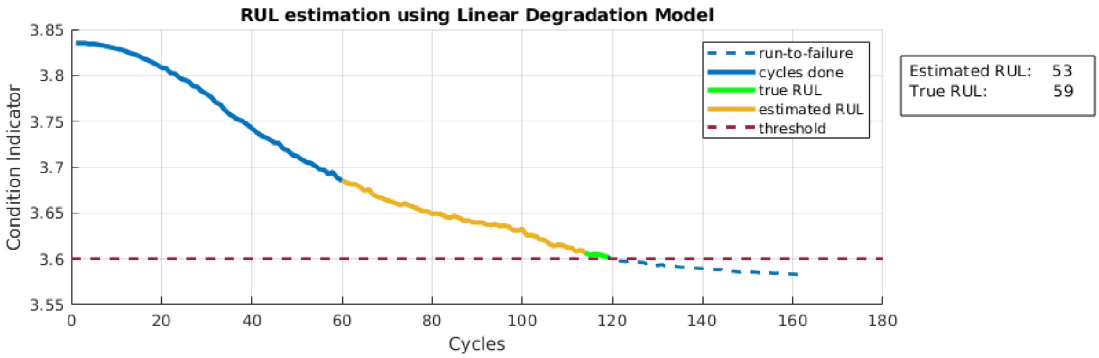
RUL

Jedním z hlavních cílů prediktivní údržby je odhadnout zbývající životnost. Původní datová sada neobsahuje záznam o historických datech, které ukazují degradační chování demonstračního zařízení.

Běžným problémem při údržbě pneumatických pístů je netěsnost vzduchu z komory, kde je umístěn píst. Tato situace byla modelovaná na simulačním modelu a generovaná data byla použita pro RUL odhad.

Vygenerovaná datová sada obsahuje 25 simulací s různou dynamikou poruch. Každá simulace zahrnuje jiný počet cyklů v závislosti na dynamice selhání. Každý cyklus obsahuje 10-sekundové měření odezvy systému. V experimentu byl jako předmět zájmu vybrán signál průtoku. Z signálu průtoku, byl vypočítán parametr shape factor, který byl použit jako indikátor stavu.

Výsledkem je možnost odhadnuti zbývající životnosti na generovaném degradačním datovém souboru pomocí residual similarity, pairwise similarity a linear degradation modelu. Předpovídané výsledky jsou uspokojivé (obr. 2).



Obrázek 2: Linear degradation model performance

Další vývoj

Pro další vývoj a zlepšení vysledků by bylo vhodné odhadnout parametry systému po částech. S důrazem na pracovní charakteristiku škrticích ventilů a tlumičů s příspůsobením.

Vhodným rozvojem by mohlo byt provedení měření poruchy úniku vzduchu a sběr historické údaje o degradaci skutečného pneumatického pístu. Následně vyhodnocení dynamiku poruchy způsobené únikem vzduchu, ověření možností odhadu zbývající životnosti pomocí snímače průtoku.

Mohla by to být zajímavá případová studie k ověření možnosti odhadu RUL pomocí mikrofonů. Pokud jsou signaly z dostupných senzorů nedostačujicí lze provádět měření tlaku v komoře. Tlak v komoře je přímo závislý na úniku vzduchu z komory, jako uvedené v rovnici 8.2. Příklad změn tlaku způsobených únikem vzduchu ze simulačního modelu je znázorněn na obrázku 8.8.

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This master's thesis is my own work and contains nothing which is the outcome of work done in collaboration with others.

Artyom Voronin

Brno . . . .

I want to express my gratitude to my supervisor, family and friends. Thank you.

Artyom Voronin

Contents

1 Introduction

Since the beginning of the industrial revolution, the complexity of production machines and serial lines has gradually increased and requires constant monitoring of the conditions of the systems for economic reasons. On the other hand, critical systems such as aircraft, spacecraft, automotive systems, nuclear reactors, and others require immediate alarm on fault, localize occurred fault, and even more predict possible future faults. These requirements have become prerequisites for Fault Detection and Analysis and Predictive Maintenance fields.

The production process always included elements of fault control and online monitoring. From the first methods of fault detection, such as visual inspection, today's factories move to automated systems consisting of sensors and computing units to evaluate the faults. Sometimes it is critical to monitor processing equipment in real-time to prevent damage caused by fault or anomaly. Every single fault can cause a slowing down of the production process and thus reducing the profit [6].

Device real-time monitoring algorithms have formed the Fault Detection and Analysis (FDA) field. FDA methods, in most cases, do not require machine learning techniques and can detect failures, using fundamental algorithms from Fourier analysis and trend checking algorithms to more complex techniques such as Gaussian Mixture Models [9].

Due to the amount of data collected in recent years and the expansion of data storage technology as cloud services and computation efficiency, it has become possible to use more advanced algorithms for fault detection and analysis. Using classification machine learning techniques, it is possible to isolate where does the fault occur. Another option that becomes available with a large amount of data is to estimate the remaining useful life (RUL) of the entire system. These techniques have led to predictive maintenance as an effort for optimal maintenance solutions. The current technical condition of the equipment is always available by information extracted from measured signals. It is possible to use current system conditions to estimate remaining useful life in time or distance measurements such as days, kilometers, or cycles. Estimated residual lifetime gives an option to plan maintenance concerning actual system conditions [10].

These remaining useful life estimation algorithms, the fault detection methods and system modeling and identification techniques form a new predictive maintenance field.

System modeling allows providing experiments and developing solutions offline before physical hardware implementations. Unavailable or challenging to implement measurements can be replaced by generated data from the simulation model and finally helps to deploy a robust algorithm.

This thesis provides a brief introduction to fault detection and predictive maintenance methodologies and a basic terminology. The 2 chapter describes the main goal and problems in these areas and focuses on similarities and differences between these two approaches.

Developing the simulation model of the double-acting pneumatic actuator and comparing it with the real-life equipment using different approaches is described in chapter 3, 4, 5 and 6.

The following chapter 7 illustrates signal-based predictive maintenance methods using different sensors available in a demonstration device. Appling preprocessing, feature extraction, and classification model, sensors were evaluated in terms of functionality, accuracy, and price.

The model-based predictive maintenance techniques and simulation model exploitation are demonstrated in chapter 8. The simulation model is used to determine the residual signals between the measured data and the simulation model's output. Also, using a simulation model, degradation data are generated and used in the remaining useful life estimation.

2 Theoretical Survey

This chapter contains a short introduction to the main goals and problems presented in fault detection and analysis and predictive maintenance techniques. A brief review of methodologies used in these fields and general approaches. Section 2.4 digital twin describes scenarios where a simulation model is used in predictive maintenance and helps develop robust, efficient algorithms.

2.1 Problem Definition

In practice many types of machinery require some calibration and monitoring for adequate working. An anomaly or fault detection in time can prevent machinery from damage that causes loss of money due to non-working or destroyed equipment. Predicting where the fault appears reduces the cost of diagnosis and replacement operations. The possibility of estimating the remaining useful life allows to optimize a maintenance process and reduce maintenance costs [2].

Smart manufacturing, the combination of sensors, the possibility of preprocessing and extracting useful information from measurements and decision algorithms based on this information, allows increasing production efficiency and significantly reducing maintenance operations.

Types of Maintenance There are three main types of maintenances (fig. 2.1). Each following type of maintenance requires increasing complexity of monitoring and decision algorithms [7]:

- Reactive maintenance, where maintenance coming after the life of the system is excess.
- Preventive maintenance is driven item by schedules that may keep the system safe but not optimal from an efficiency/cost perspective.
- Predictive maintenance is an effort to optimize a maintenance strategy.

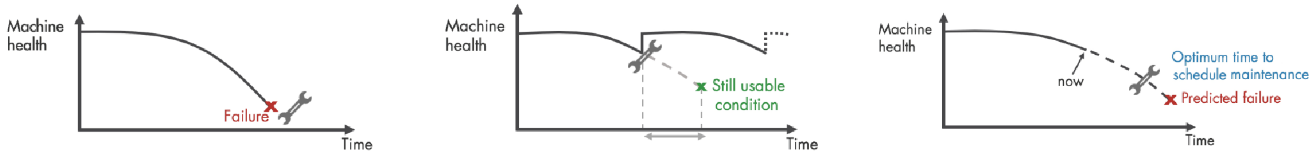


Figure 2.1: Reactive, preventive and predictive types of maintenance [7]

Fault Types A fault is not an acceptable deviation of at least one characteristic or parameter of the system from the standard condition. There are different faults by their sources.

- Plant faults appear in system behavior and cause manufacturing performance.
- Component fault
- Sensor faults occurred in the sensor during measurements.
- Combination of faults

In many cases, faults lead to a system failure and the system is no longer able to perform required functions. There may also be a malfunction after which the system returns to normal operation.

Faults can be classified by the location where they appear, by a fault form, or based on the form in which the fault is added to the system [2].

2.2 Fault Detection and Analysis (FDA)

Fault Detection and Analysis, FDA (Fault Detection and Isolation, FDI) is a subfield of control engineering focused on detecting the fault and identifying where this fault is located [5]. The main goals of FDI are

- Fault detection, detect anomalies in real-time
- Fault isolation, find the root cause
- Fault identification, estimation of the magnitude, type, or nature of the fault

Several methods are partly overlapped but divided into two main categories.

Signal-Based methods Signal-Based methods (SB), explore measured data and extract useful information in the form of features 2.2. The following methods belong to the SB approach:

- Limit and trend checking
- Spectral analysis
- Data analysis (PCA)
- Pattern recognition

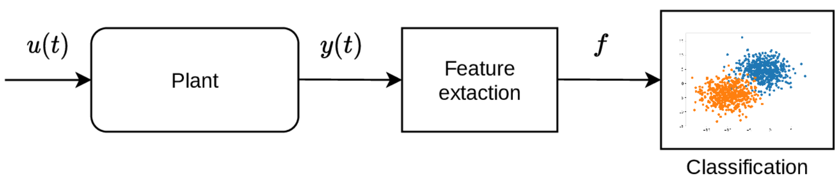


Figure 2.2: Signal-Based Method

Model-Based methods Model-Based methods exploit models identified from real-life systems 2.3. The model-based approach is suitable when it is difficult to gain useful information using only measured signals. If the system structure is known, it is possible to extract features such as state variables or some system parameters. Another option is to compare real system behavior with nominal healthy model and use residuals as inputs to decision algorithms [12]. Typical model-based techniques include

- Residual estimation (compare measurements with "healthy" model)
- Polynomial coefficients
- State variables estimated using state observers
- Parameter estimation

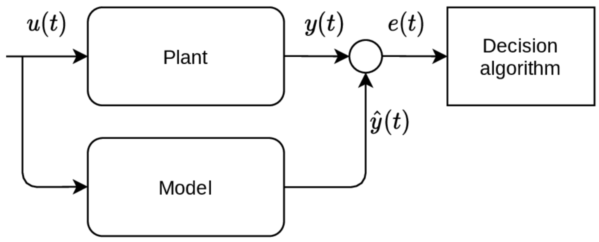


Figure 2.3: Model-Based Method

Automated fault detection depends on input from sensors and postprocessing algorithms. In many manufacturing applications, sensor failures are the most common equipment failure.

The result of FDI is the detection and identification of faults that occur during the operation of the device. Subsequently, predicted faults are processed using fault tolerance and predictive maintenance algorithms.

Fault Tolerance: Provide the system with the hardware architecture and software mechanisms that will allow, if possible, to achieve a given objective in normal operation and given fault situations [5].

2.3 Predictive Maintenance (PdM)

Predictive maintenance (PdM) is cost-effective maintenance strategy that predicts time to failure and warns of an anticipated location where this could occur.

2.3.1 Goals

There are two main goals of predictive maintenance, remaining useful life (RUL) estimation and identification where the future failure can appear or what is the reason for decreasing RUL. As a result of PdM is RUL representing the number of cycles, days or time before the fault occurred. And the probability of when or where this fault can appear [12].

2.3.2 Overview of the PdM development workflow

Figure 2.4 represents the recommended PdM development workflow. The development of predictive maintenance algorithms starts with raw measured signals from sensors. For

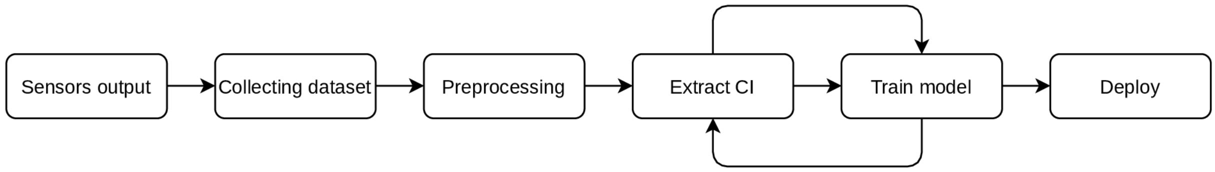


Figure 2.4: Predictive maintenance development sequence

further working with data, it is a good manner to combine measurements to a dataset with a logical structure of elements. In this thesis, a common data ensemble structure was used. Each measurement has its own data file with all measured signals at a particular time.

If collected data require some preprocessing techniques as data cleaning, smoothing or filter the signal, detrend, normalizing, etc., it can be done at this step.

The next step is to extract condition indicators using predictive maintenance methods described in 2.3.3. As long as the optimal solution is not found, try to figure out the best combination of condition indicators described in 2.3.4 and train different classification models iteratively. After the efficient solution is found, deploy the algorithm to work recursively with the study-case system.

2.3.3 Condition Indicators

In the prediction maintenance field, features extracted from measured signals are called **Condition Indicators, CI**.

Condition Indicators represent some system behavior and hide information about system operation conditions. Generally, CI is represented by three main domains. There is a time domain, frequency domain, time-frequency domain. But in fact, CI can be any system parameter or value corresponding with the system's current condition [12].

The methods of extracting condition indicators from the signal are defined in the same way as in FDI 2.2.

The **signal-based approach** is suitable when we have measurements from the system in different operating conditions. But there is a problem that signal-based approach enables classifying and learning just the patterns observed in the training dataset. On the other hand, the **model-based approach** uses physical failure models and does not require a large dataset of failure data. And they may work in situations never observed in data before. Moreover, the model-based method is helpful in case the measured signal has a more complex relationship with the input signal.

Between common signal-based CI belongs:

- Time-domain: mean, standard deviation, RMS, skewness, etc.
- Frequency-domain: mean frequency, peak values/frequencies, power bandwidth, etc.
- Time-frequency domain: Spectral entropy/kurtosis, moments, etc.

Model-based approach use model properties such as:

- poles and zeros location
- damping coefficient
- state variables values

- modal analysis
- residual values

2.3.4 Condition Indicators Ranking

Multiple condition indicators can be extracted from each sensor signal. A good practice to reduce the number of CI and keep only those which provide essential information.

One of the possibilities is applying Principle Component Analysis (PCA) to transform features from one coordinate system to a new orthogonal basis. Data reduced by using the first n principal components that optimally describe the variance of the dataset. Applying the PCA algorithm still requires the extraction of all condition indicators from the signal.

Another option is to rank the futures using the Analysis of Variance (ANOVA) algorithm. This algorithm describes relations among CI in the form of their mean values. The result gives information about how much particular CI represents data. Using the first n CI, we reduce the number of CI and reduce the number of extracted features from measured signals. This fact means that using ANOVA reduced the time and complexity of calculations [12].

2.3.5 Fault Classification

Classification models are used to recognize faults from a set of CI. The set of CI must contain labels that determine the current condition of the device in the form of fault code, string, etc. The correlation between different CI can be explored using a 2D or 3D scatter plot. The model performance is usually represented by total accuracy and confusion matrix, where on one axis there are true labels and on the other there are predicted from the model. The common types of classification models are:

- Decision Trees
- Supported Vector Machines (SVM)
- Neigherest Neighbors (KNN)
- Ensemble Classifiers
- Neural Networks (ANN)

A good practice is to divide an original dataset of CI into train, validation and test sub-datasets to prevent model overfitting. Choosing the best classification model depends on training data and requires experiments with different models.

2.3.6 Remaining useful life

The remaining useful life (RUL) is the expected time remaining before the machine requires repairment or replacement, and it is a central goal of PdM.

The problem of estimating the remaining useful life is connected with evaluating condition indicators associated with the system's degradation process. These condition indicators must satisfy the requirements for monotonicity, trendability, and prognosability [12].

The models used to estimate the remaining useful life depend on the historical data which are available. There are three types of possible models.

Survival model The survival model is considered when we have only failure data available, but the whole degradation history is not recorded. The probability density function can be obtained from failure data and used to estimate RUL.

Degradation model The degradation model gives an option to estimate RUL based on data without failure moment captured but only recorded degradation process. In this situation, it is necessary to determine a safety threshold that CI must not cross.

Similarity model In case we have the whole history of the degradation process of similar systems, including failure, the similarity model can be used. The upcoming CI is compared with historical degradation paths obtained from the training dataset and the best similarity trend is evaluated as RUL value.

2.4 Digital Twin

A digital twin is a digital representation of the real-life system. It can be represented as a component, a system of components, or as a system of systems.

A digital twin can be updated with incoming data from sensors. Fitting the model to new data, the digital twin represents the current condition state of the real-world object. There are many advantages of using models in PdM. A digital twin can hold historical data about system behavior. Apart from this, it can be used for simulation system operation in different conditions, designing control and simulating future behavior (RUL, "What-if"). The dataset extended by data from the simulation model represents synthetic dataset. This dataset type can contain different measured fault and healthy data of the system and hard to realizable in real-world fault situations [3].

A mathematical model of the real-world system can be created using different approaches.

- First-principles modeling requires an understanding of the fundamental process of the system.
- Physical modeling (Simscape).
- Data-driven modeling where the system is represented as a Blackbox.
- Combination of multiply approaches.

2.5 Comparison PdM and FDA approaches

Figure 2.5 presents a relative arrangement of Predictive Maintenance (PdM) and Fault Detection and Identification (FDI or FDA) algorithms. From the figure, it is clear that Predictive Maintenance is an extension of the FDI approach, with recommended workflow techniques suitable for optimizing system maintenance.

Both methods are closely overlapped and use quite similar techniques. However, predictive maintenance over the FDA is extended by RUL estimation. And it leads not only to fault detection and monitoring at a given moment but also to the possible prediction of a fault in the near future.

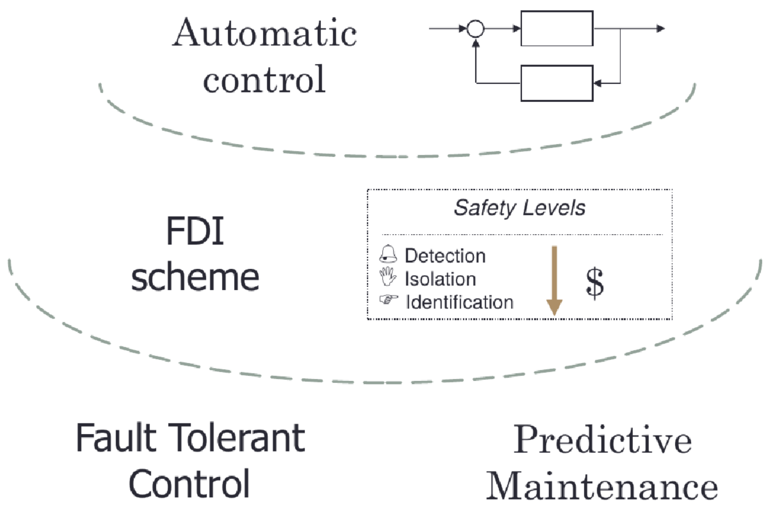


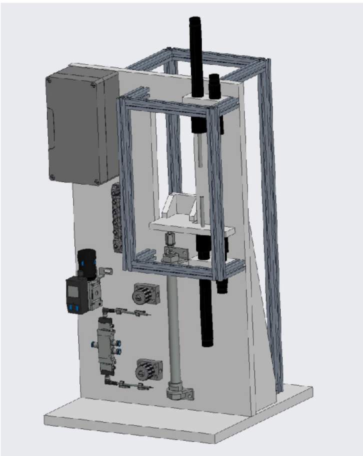
Figure 2.5: Relative arrangement of PdM and FDI algorithm [5]

2.6 Applications

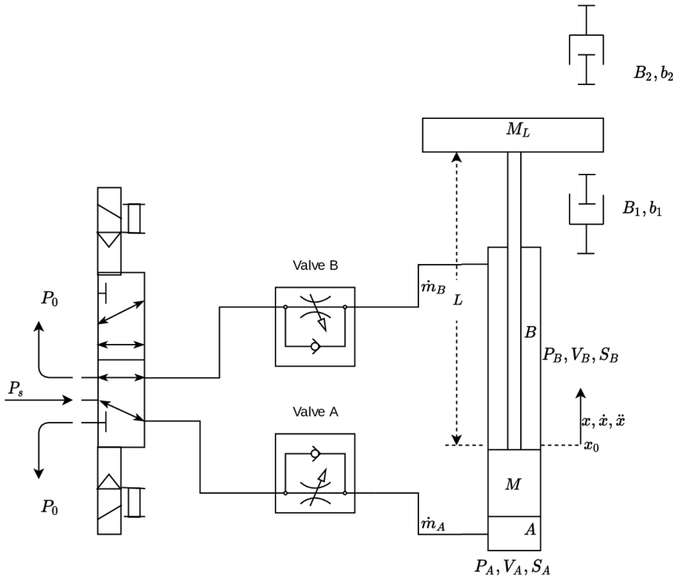
The most significant interest in PdM is the manufacturing sector that requires efficiency maintenance strategies to increase productivity and reduce money-lost [6]. The PdM is used in the field that is highly dependent on safety types of machinery such as aircraft or rail industry. Using the PdM condition monitoring, it is possible to prevent unexpected fails. The oil and gas industry supports the PdM field; due to the amount of data collected in these industries, the PdM techniques are beneficial.

3 Demonstration Device Overview

3.1 Double-Acting Pneumatic Actuator



(a) 3D render of the demonstration device



(b) Schematically representation of the demonstration device

Figure 3.1: Demonstration device

The case study of this thesis is the double-acting pneumatic piston, with a pneumatic circuit and mechanical assembly driven by a piston. Figure 3.1b is a schematical representation of the system. Figure 3.1a is a 3D render of the system.

Pneumatic systems use air to transmit power between components in the circuit. The air is a compressible gas, and we have to consider this when designing a model. Pneumatic actuators are highly efficient and fast drives. Using compressed air pneumatic actuator can move with high velocities and supply nominal force in the kN range. One of the advantages of a pneumatic system with a piston is that only one supply line is necessary, giving many opportunities to design and maintain the system. The basic pneumatic system includes an air reservoir with supplied air, pressure lines connection, pneumatic actuator and control valve to connect the supply pressure and actuator. Resistance to movement places a mass that acts on the piston.

In this thesis, a double-acting pneumatic actuator, as shown in figure 3.1b was used. Throttling valves A and B regulate the air mass flow to the piston's chambers. Proportion valve connects supply and ambient pressure lines to achieve piston control. There are two pairs of dampers installed to prevent possible destruction impact and simulate different material penetration resistance.

The demonstration device can be used in stamping, drilling, moving applications. All system parameters concerning datasheets and measurements are described in attachment *models/params.m*.

3.2 Sensors

There are eight types of sensors located on the system. Table 3.1 describes a sensor purpose, signal name in the datastore, and the signal unit.

Sensor	**Unit**	**Description**	**Name**
Encoder	m	displacement	LeverPosition
Encoder	m/s	velocity	LeverVelocity
Accelerometer	g	accelerometer on moving part	AccelerometerMovin_axisZ/Y
Accelerometer	g	accelerometer on static part	AccelerometerStatic_axisZ/Y
Flow Sensor	l/min	air flow extrusion to A chamber	FlowExtrusion
Flow Sensor	l/min	air flow contraction from A chamber	FlowContraction
Pressure	bar	pressure measurement in reservoir	AirPressure
Microphone	V	microphone on upper bumper	MIC_uBumper
Microphone	V	microphone on bottom bumper	MIC_bBumper
Microphone	V	ambient microphone	MIC_Ambient
Temperature	^{o}C	cylinder temperature measurement	TempCylinder
Temperature	^{o}C	ambient temperature measurement	TempAmbient
Strain Gauge	Pa	strain measurements	StrainGauge
Proximity	-	upper bound detection	ProximitySensor_upper
Proximity	-	bottom bound detection	ProximitySensor_bottom

Table 3.1: Sensors overview

The dataset measured on the system contains almost five thousand measurements in different operating conditions. Each measurement includes a 10-second recording of moving the pistol up and down. This data was given in the format of massive files with the ".mat" extension, which was divided into files contains only one measurement. The divided dataset is easier to maintain, and Matlab recommends this type of datastores called Data Ensemble [1].

The measured examples are shown in figures 3.2, 3.4, 3.4, and 3.5 .

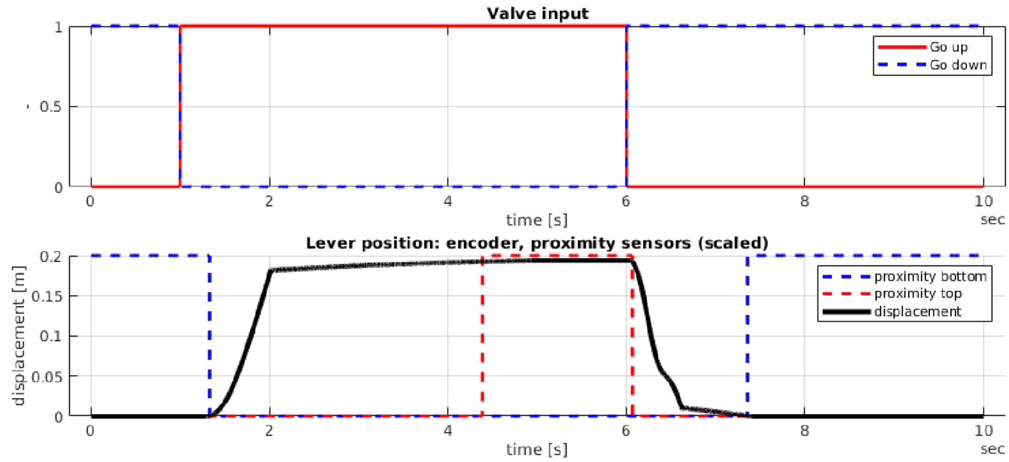


Figure 3.2: Example of measured signal

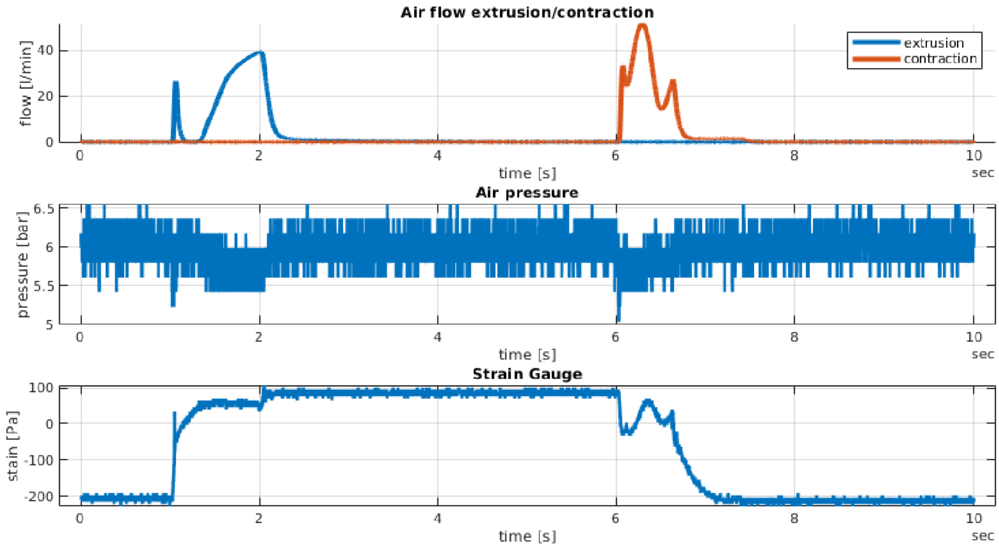


Figure 3.3: Example of measured signal

3.3 Fault Conditions

The demonstration device contains various settings that were used to change the system's behavior; these settings are presented in Table 3.2.

Different loads and material resistance is acting on the pneumatic piston during various work operations. Setting parameters can be set for each working operation to run in the so-called health condition. In which the parameters for effective functionality and extension of component life are optimally set. However, occasionally there is an undesirable change of the parameter, which can then cause a fault or inefficient functionality. These situations need to be corrected and the possible cause pointed out.

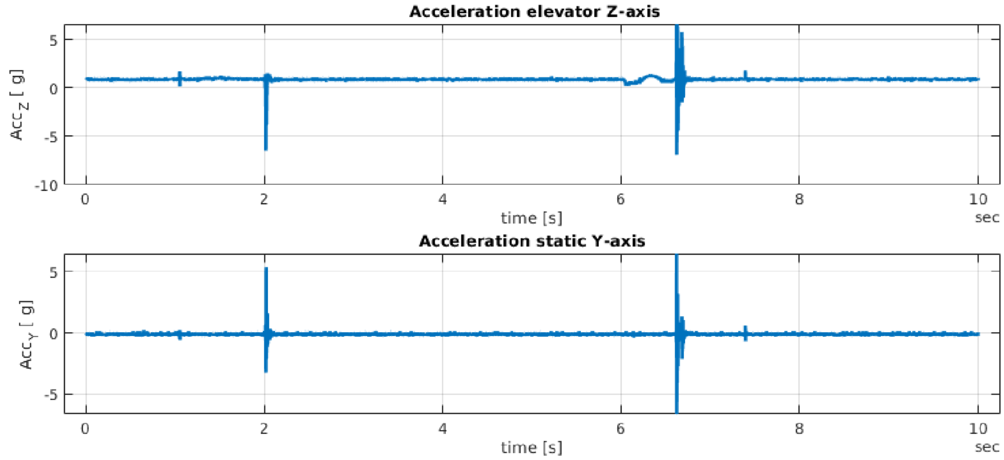


Figure 3.4: Example of measured signal

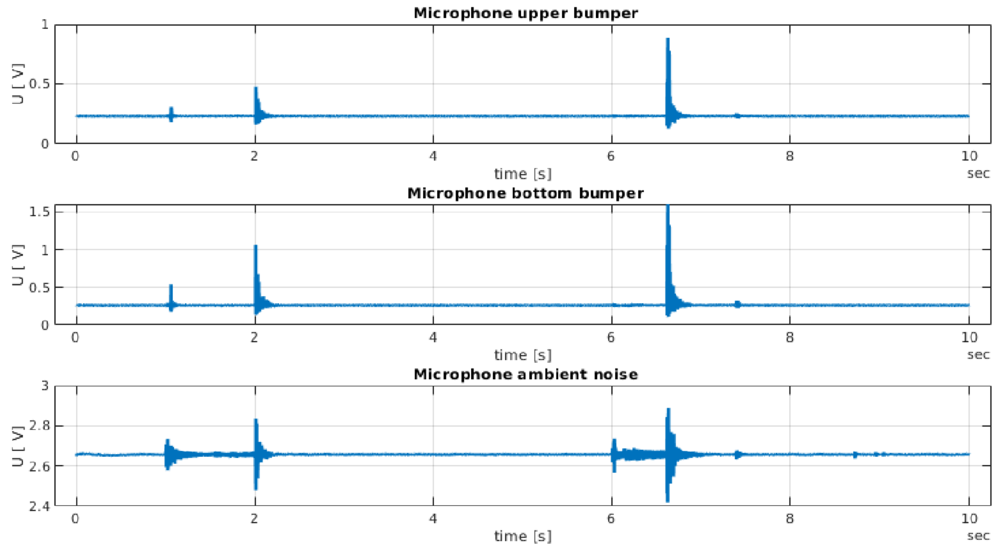


Figure 3.5: Example of measured signal

The measured dataset observes almost 250 different condition situations where setting parameters were changed to simulate the fault behavior of the system. Each case has a unique fault code for orientation in the dataset. Nevertheless, for further development, these fault codes were combined according to where the fault occurs. Thus combined fault codes were added to the dataset as labels.

These 20 labels were further used for PdM algorithms:

- Healthy
- Throttle valve 1
- Throttle valve 2
- Small damper bottom
- Small damper upper

equipment	**values**
Throttle valve 1	adjustment in range 1 to 10 [-]
Throttle valve 2	adjustment in range 1 to 10 [-]
Small damper upper	adjustment in range 1 to 10 [-]
Small damper bottom	adjustment in range 1 to 10 [-]
Large dampers	without adjustment, static value
Load mass	0, 1.25, 5, 6.25 [kg]
Supply pressure	5, 6 [bar]

Table 3.2: Demonstration device settings equipment

- Large dampers
- And combinations of these faults

4 First Principle Modeling

First-principle modeling is a common engineering modeling approach. Models developed using physical laws such as energy and mass balance, heat transfer, and so on. First-principle modeling requires knowledge of the system and the physical processes that take place in this system.

First principle models (FPMs) are usually designed in the form of a system of differential equations, algebraic differential equations, transfer functions, state-space systems, etc. In designing FPMs, it is necessary to determine the assumptions and simplifications that correspond to the level of technical resolution in a particular problem.

This chapter introduces the design of a double-acting pneumatic piston assembly model, including sensors using a first-principle modeling approach.

4.1 General physical principles

Assumptions

1. The effect of accelerated air mass is neglected.
2. The gas is ideal.
3. All the thermal processes are adiabatic.

Simplifications Throttle modeling and adjustment dampers require measurements that were unfortunately not available. In the case of throttle valves, the parameters of the throttle valves were combined with the parameters of the control solenoid valve.

Equation of state Equation of state for an ideal gas 4.1, describe the relationships between temperature T, mass m, pressure p and V volume of the gas, where $R = 287.1[\mathrm{Jkg^{-1}K^{-1}}]$ is an ideal gas constant [13].

$$pV = mRT \tag{4.1}$$

Adiabatic process All processes take place without heat exchange with the environment by given equation 4.2, where $\kappa = c_p/c_v$ is a heat capacity ratio [13].

$$p_1V_1^{\kappa} = p_2V_2^{\kappa} = const \tag{4.2}$$

Relation between heat capacities and an ideal gas constant is given by Mayer's equation as $c_p = c_v + R$. Where c_p, c_v are heat capacities at constant pressure, volume.

Bernouilli's principle Bernouilli's equation 4.3 describes flow dynamics as a sum of kinetic, potential and internal energies.

$$H_1 + \frac{mw_1^2}{2} + mgz_1 + Q = H_2 + \frac{mw_2^2}{2} + mgz_w + W_T \tag{4.3}$$

Transition to specific values:

$$h_1 - h_2 = -\int_1^2 vdp = c_p(T_1 - T_2) = c_pT_1\left(1 - \frac{T_2}{T_1}\right) \tag{4.4}$$

where

z	m	height
w	ms^{-1}	flow speed
H	J	enthalpy
ν	$\mathrm{m^3kg^{-1}}$	specific volume
Q	J	heat shared with environment
W_T	J	work shared with environment

Table 4.1: List of Symbols

Continuity equation Continuity equation 4.5 describes a mass flow through a control volume. Where S is cross-section and ρ air density.

$$\dot{m} = S_1w_1\rho_1 = S_2w_2\rho_2 = const \tag{4.5}$$

4.2 Air Expansion

Air expansion from the reservoir, one of the fundamental sets of equations used in pneumatic elements [8].

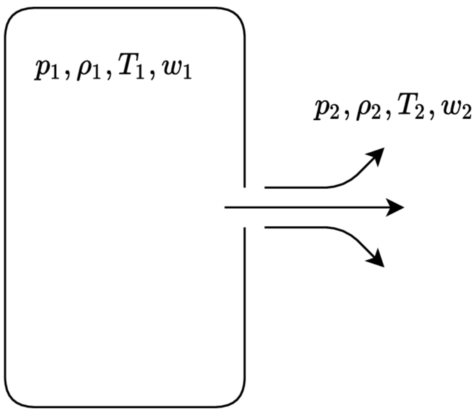


Figure 4.1: Air expansion from tank

Assuming that $W_T = 0, Q = 0$ there is no work and heat shared with the the environment, there is no difference in height $z_1 = z_2$ and the velocity difference is vast $w_2 << w_1$, applying equation 4.3, get 4.7.

$$w_2 = \sqrt{2(h_1 - h_2)} \tag{4.6}$$

$$w_2 = \sqrt{2c_pT_1\left(1 - \frac{T_2}{T_1}\right)} \tag{4.7}$$

where

$$T_1 = \frac{p_1}{R\rho_1} \qquad c_p = R\left(\frac{\kappa}{\kappa - 1}\right) \qquad \frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}} \tag{4.8}$$

Combine equations 4.7, 4.8 to get air expansion velocity 4.9.

$$w_2 = \sqrt{2\frac{\kappa}{\kappa - 1}\frac{p_1}{\rho_1}\left[(1 - \left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}}\right]} \tag{4.9}$$

From equations 4.8 express air density 4.10.

$$\rho_2 = \frac{p_1}{RT_1}\left(\frac{p_2}{p_1}\right)^{\frac{1}{\kappa}} \tag{4.10}$$

Using continuity equation 4.5 and 4.9 describe mass flow as 4.11:

$$\dot{m} = Sp_1\sqrt{\frac{2}{RT_1}}\sqrt{\frac{\kappa}{\kappa - 1}\frac{p_1}{\rho_1}\left[(1 - \left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}}\right]} \tag{4.11}$$

where 4.12 is the outflow function.

$$\psi\left(\frac{p_2}{p_1}\right) = \sqrt{\frac{\kappa}{\kappa - 1}\frac{p_1}{\rho_1}\left[(1 - \left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}}\right]} \tag{4.12}$$

Finally mass flow expansion from the reservoir is given by equation 4.13:

$$\dot{m} = Ap_1\sqrt{\frac{2}{RT_1}} \cdot \psi\left(\frac{p_2}{p_1}\right) \tag{4.13}$$

Critical flow velocity The outflow function depends on the pressure ratio p_2/p_1. This function has a maximum value when the critical pressure is reached; the mass flow becomes chocked. Critical pressure is presented by 4.14. For the overcritical pressure ratio, the mass flow depends only on p_1 and T_1 [8].

$$\left(\frac{p_2}{p_1}\right)_{crit} = \left(\frac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} = \beta_k \tag{4.14}$$

Critical pressure for air is $\beta_k = 0.528$ and critical velocity is give by outflow function 4.15. Combine equations for overcritical and undercritical pressure ratio using equations 4.14, 4.15 we get the final equation for outflow function 4.16.

$$\psi_{max}(\beta_k) = \left(\frac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \sqrt{\frac{\kappa}{\kappa+1}} = 0.484 \tag{4.15}$$

$$\psi\left(\frac{p_2}{p_1}\right) = \begin{cases} \sqrt{\frac{\kappa}{\kappa-1}\frac{p_1}{\rho_1}\left[(1-\left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}}\right]} & 0.528 < \frac{p_2}{p_1} \leq 1 \\ \left(\frac{2}{\kappa+1}\right)^{\frac{1}{\kappa+1}}\sqrt{\frac{\kappa}{\kappa+1}} & 0 \leq \frac{p2}{p1} \leq 0.528 \end{cases} \tag{4.16}$$

A detailed derivation of the equation 4.16 can be found in [8],[13].

4.3 Pneumatic Piston Pressure Model

A construction principle of the double-acting pneumatic piston is shown in the figure 4.2. There are two chambers connected to the control valve. If the control valve is connected to chamber A, the supply pressure drives mass flow into chamber A. At the same time, the port at chamber B is connected to the ambient. Due to the pressure difference between chambers, pneumatic piston stroke start moving in a positive direction. After the bound is reached and the pressure in the chamber equalizes to supply pressure, there is no longer any mass flow coming inside.

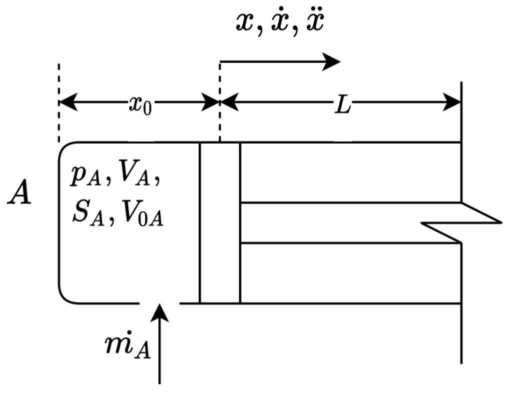


Figure 4.2: Piston chamber

Assuming an isothermal process, derivation of the equation of state $m = \rho V$ get the equation 4.17.

$$\dot{m} = \dot{\rho}V + \rho\dot{V} \tag{4.17}$$

where

$$\rho = \frac{p}{RT} \qquad \dot{\rho} = \frac{\dot{p}}{RT} \tag{4.18}$$

Equation 4.19 describe pressure difference in chamber due mass flow.

$$\dot{p} = -\frac{p}{V}\dot{V} + \frac{RT}{V}\dot{m} \tag{4.19}$$

For the adiabatic model of the pressure difference in the chamber, moreover, heat capacity ratio added 4.20.

$$\dot{p} = -\frac{\kappa p}{V}\dot{V} + \frac{\kappa RT}{V}\dot{m} \tag{4.20}$$

Volumes of the chambers can be represented concerning figure 4.2 as volumes equations 4.24.

$$V_A = S_A x + V_{0A} \tag{4.21}$$

$$V_B = S_B(L - x) + V_{0B} \tag{4.22}$$

$$\dot{V}_A = S_A \dot{x} \tag{4.23}$$

$$\dot{V}_B = -S_B \dot{x} \tag{4.24}$$

The pneumatic piston with chambers A, B is described by the system of differential equations 4.25, 4.26. These equations describe a pneumatic cylinder entirely. Furthermore, all the parameters can be directly measured or found in the datasheet [14].

$$\dot{p_A} = \frac{\kappa}{S_A x + V_{0A}}\left(-p_A S_A \dot{x} + RT_A \dot{m_A}\right) \tag{4.25}$$

$$\dot{p_B} = \frac{\kappa}{S_B(L - x) + V_{0B}}\left(p_B S_B \dot{x} + RT_B \dot{m_B}\right) \tag{4.26}$$

4.4 Control Valve Model

The pneumatic control valve manipulates air mass flow to connect piston chambers with supply and ambient pressure lines. There are different approaches to model pneumatic control valve describes [8], [15]. Demonstration device includes 5/2 bistable solenoid valve 3.1b. The movable part, valve spool driven by a magnetic field, can be in the two positions, where one of the chambers connects to the supply pressure line, another to ambient. A digital input signal switches between up and down positions. Equation 4.27, describe the input signal $u \in \langle -1, 1 \rangle$, which regulates the spool movement to acquire one of the states.

$$u = \begin{cases} -1 & \text{discharge the chamber} \\ 1 & \text{filling the chamber} \end{cases} \tag{4.27}$$

Spool dynamic and pressure lines transport delay can be modeled as a 1dof system with the time constant T and delay τ (eq. 4.28) [8]. For more precise control and modeling of the valve system, valve dead zones can be considered 4.29.

$$G(s) = \frac{1}{Ts+1} e^{-\tau s} \tag{4.28}$$

$$u_z = \begin{cases} g_z(u) < 0 & , \text{ if } u \leq u_n \\ 0 & , \text{ if } u_n < u < u_p \\ h_z(u) > 0 & , \text{ if } u \geq u_p \end{cases} \tag{4.29}$$

To parametrize the pneumatic valve discharge coefficient (coefficient of contraction) can be used. This parameter must be determined experimentally. The discharge coefficient 4.30 is the ratio between the equivalent area of the opened flow path and the maximum area of this path. The equivalent area limits the maximum mass flow value [15]. But commonly, this parameter estimates from measurements and also known as valve coefficient [13].

$$C_d = \frac{S_{eq}}{S_{max}} \tag{4.30}$$

With respect to outflow function 4.16 and mass flow function 4.13 derived in section 4.2, control valve equation is given 4.31.

$$\dot{m} = u S_{max} C_d p_1 \sqrt{\frac{2}{RT_1}} \cdot \psi\left(\frac{p_2}{p_1}\right) \tag{4.31}$$

But commonly, all parameters approximate to one coefficient estimated from measurements and also known as valve coefficient $C = S_{max} C_d$ [13].

Using the notation introduced on the schemes 3.1b, 4.2 we compile a complete set of equations for the description of the behavior of a pneumatic solenoid valve 4.32, 4.33.

For filling the chamber:

- $p_1 = p_s$
- $p_2 = p_A$ or p_B
- $T_1 = T_s$

For discharge the chamber:

- $p_1 = p_A$ or p_B
- $p_2 = p_0$
- $T_1 = T_A, T_B$

where p_s is supply pressure. p_0 atmospheric pressure, $T_A = T_B = T_0$ ambient temperature.

$$\dot{m}_A = \begin{cases} u(t)C_{A,in}p_s\sqrt{\frac{2}{RT_s}} \cdot \psi\left(\frac{p_A}{p_s}\right) & , u \in (0,1\rangle \\ u(t)C_{A,out}p_A\sqrt{\frac{2}{RT_A}} \cdot \psi\left(\frac{p_0}{p_A}\right) & , u \in \langle -1, 0) \end{cases} \tag{4.32}$$

$$\dot{m}_B = \begin{cases} u(t)C_{B,in}p_s\sqrt{\frac{2}{RT_s}} \cdot \psi\left(\frac{p_B}{p_s}\right) & , u \in (0,1\rangle \\ u(t)C_{B,out}p_B\sqrt{\frac{2}{RT_B}} \cdot \psi\left(\frac{p_0}{p_B}\right) & , u \in \langle -1, 0) \end{cases} \tag{4.33}$$

4.5 Mechanical assembly

4.5.1 Equation of motion

The motion of the pneumatic piston mechanism describes in terms of the general 1dof dynamical equation 4.34.

$$m\ddot{x} + b\dot{x} + kx = u \tag{4.34}$$

In the case of the pneumatic piston, equation 4.34 transforms into and equation 4.35 [8].

$$(M + M_L)\ddot{x} + F_d + F_g + F_{hs} + F_f = F_p \tag{4.35}$$

Where M represents a mass of the all moveable part of the piston, M_L is load mass, F_g gravity force acting to mechanical moving assembly, F_{hs} - models endpoints (hard stop), F_d represents dampers (shock absorbers) acted at endpoints, F_f describe Coulomb and viscous friction, F_p is a force produced by the pneumatic piston and given by equation 4.36.

$$F_p = P_A S_A - P_B S_B - P_0 S_0 \tag{4.36}$$

Friction Friction force was modeled as a Coulomb and viscous friction4.37.

$$F_f = F_C \cdot \mathrm{sign}(\dot{x}) + B_v\dot{x} \tag{4.37}$$

4.5.2 Hard Stop

The endpoint's material resistance can be represented as springs and dampers acting as one-way bound 4.5.3 The parameters K, D have a significant impact on the numerical stability of the simulation system; therefore, they were tuned concerning stable performances.

$$F_{hs} = \begin{cases} K_p(x - g_p) + D_p\dot{x} \cdot ge(\dot{x}, 0) & \text{for } x \geq g_p \\ 0 & \text{for } g_n < x < g_p \\ K_n(x - g_n) + D_n\dot{x} \cdot le(\dot{x}, 0) & \text{for } x \leq g_n \end{cases} \tag{4.38}$$

where $ge(), le()$ greater or equal and less or equal functions.

4.5.3 Endpoint dampers

There are two types of dampers installed in demonstration device. One pair is adjustable, and other stable. Endpoint dampers were modeled in the same way as a hard stop , emphasizing damping coefficient D.

4.6 Sensors Modeling

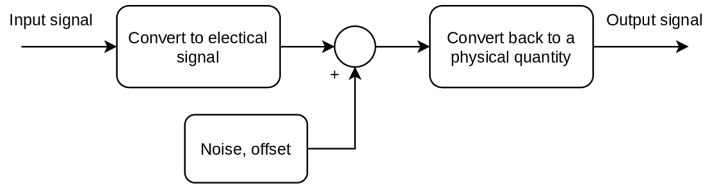


Figure 4.3: Sensors Modeling Diagram

Modeling sensors include converting the measured physical signals to an analog or digital signal, adding noise and offset parameters to have an option to model faults conditions, and after converting back to the sensor's measured units (fig. 4.3). All sensors parameters are available in attachment *models/sensors.pdf*

Flow sensors Flow sensors are a typical representative of a comfortable sensor to implement by converting the units used in the model [kg/s] into a voltage [V] concerning the datasheet. Then added measurement noise and the possibility to add offset for further experiments and finally, converting back to physical quantity with respect to the sensor measuring in [l/min].

Strain Gauge Strain Gauge was modeled similarly as a flow sensor with the possibility of experimentation with the magnitude of noise and offset.

Accelerometer The accelerometer attached to the moving part of the system was modeled using a transfer function concerning the datasheet and estimated magnitude of the measurement noise. It's also the ability to add optionally offset or off the sensor itself.

Proximity sensors In the case of digital signals such as proximity sensors, it is sufficient to control the boundaries at which the sensor is switched on.

Encoder The demonstration device includes a very precise linear magnetic encoder with a resolution $\approx 7\mu$m. This sensor provides an almost clean signal that gives an option to extract velocity signal by numerical derivation. However, to model this type of encoder with parameters of real encoder requires a minimum sample time in the range of μs. Due to this fact model of the encoder was embedded, but the output is taken directly from the model.

Not implemented Sensors that are difficult to implement or have not been included in the model have not been implemented. These sensors include microphones, a static accelerometer mounted on a construction pad, an air pressure sensor because the air reservoir was not modeled in this work, temperature sensors.

4.7 Parameter Estimation

To achieve closer behavior to the real system, it is necessary to determine all the parameters of the model. There are parameters given as physical constants, or they can be directly measured or determined in the datasheet. Parameters that do not fall under these kinds must be deducted from the measurement.

According to the simplification estimation process, throttle valves and solenoid valve parameters were combined into two valve coefficients $C_{i,in}, C_{i,out}$ in both input and output directions 4.39.

$$\dot{m}_{i,in} = u(t) \cdot \boldsymbol{C_{i,in}} p_1 \sqrt{\frac{2}{RT_1}} \cdot \psi\left(\frac{p_2}{p_1}\right) \quad \dot{m}_{i,out} = u(t) \cdot \boldsymbol{C_{i,out}} p_1 \sqrt{\frac{2}{RT_1}} \cdot \psi\left(\frac{p_2}{p_1}\right) \tag{4.39}$$

where i are ports to chambers A, B.

Solenoid valve spool dynamic was estimated with respect to equation 4.28 in different displacement measurements.

Pneumatic piston parameters were taken from the datasheet, and the remaining such as dead volumes V_{0A} and V_{0B} estimated approximately.

Hard stop endpoints were determined from the construction design of a particular pneumatic piston. The values of the damping and spring were estimated to perform their functions and at the same time maintain numerical stability.

Adjustment dampers were estimated from displacement measurement as b_{bot}, b_{up} parameters. The bounding range was directly measured from the displacement measurements.

parameter	**description**
$C_{A,in}$	valve coefficient connected to input path to A chamber
$C_{A,out}$	valve coefficient connected to output path from A chamber
$C_{B,in}$	valve coefficient connected to input path to B chamber
$C_{B,out}$	valve coefficient connected to output path from B chamber
b_{up}	upper damper value
b_{bot}	bottom damper value

Table 4.2: Parameters to reestimate for different fault conditions

4.8 Model performance

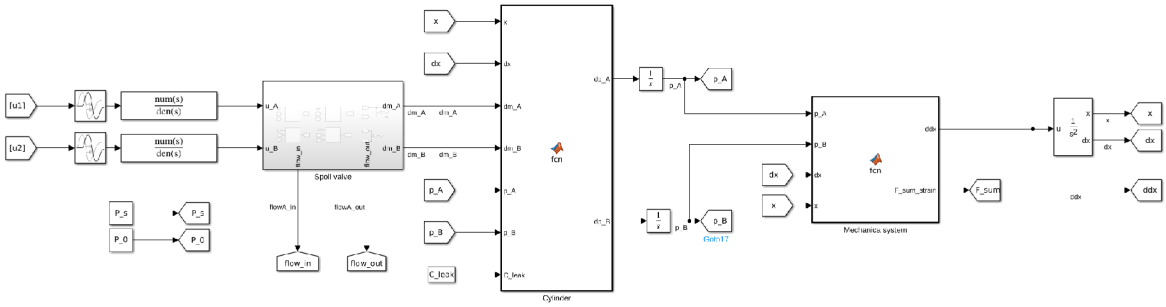


Figure 4.4: First Principle model implementation in Simulink

The model was implemented using the Matlab/Simulink software using basic Simulink operations and the Matlab-Function block. The model shows good numerical stability and allows to perform simulations with a fix step solver with a sampling time of $1 \cdot 10^{-3}$ s. Which significantly speeds up the simulations and the process of parameter estimation. Figure 4.4 shows the central part of the model in the Simulink environment.

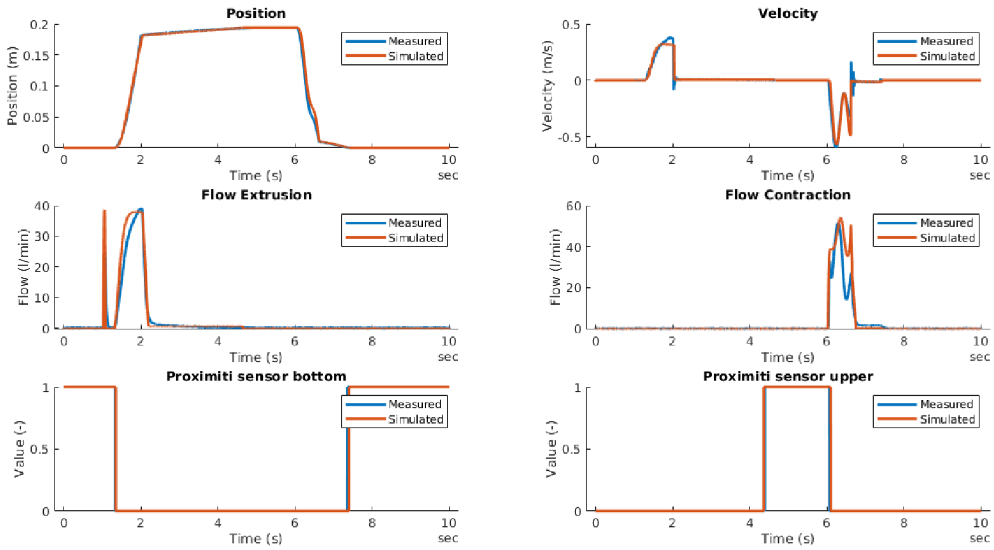


Figure 4.5: Comparison between measurement and model response

The resulting behavior of the system after parameter estimation on health conditions data is shown in Figure 4.5. However, it is possible to reestimate the basic parameters 4.2 and thus realize the behavior of the system closer to the fault state.

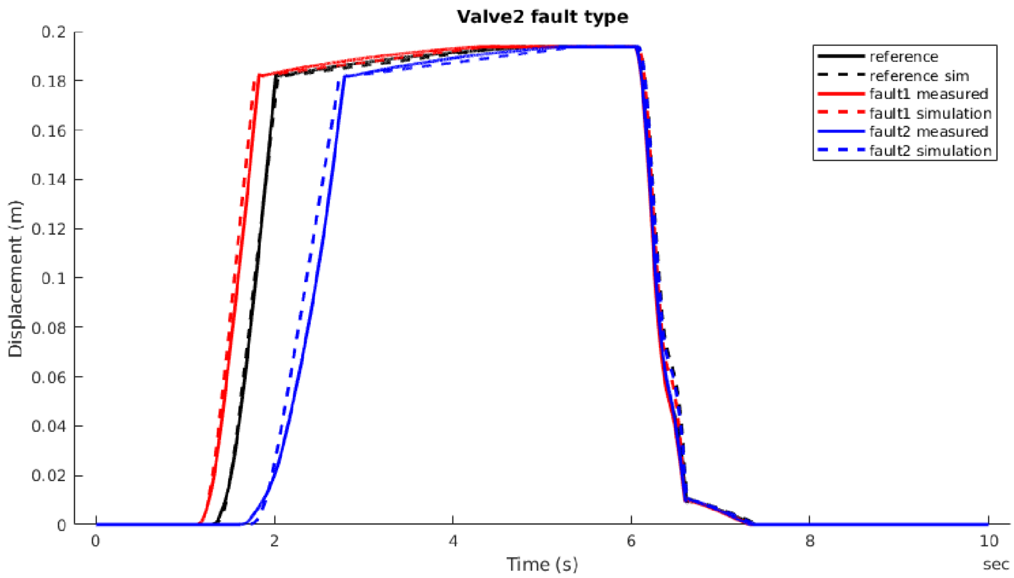


Figure 4.6: Simulation model performance in different fault conditions

Figure 4.6 shows the simulation system response with different estimated parameters for the fault states caused by the Throttle valve 2. In the case of position, the measured and simulation signals practically overlap.

5 Alternative Modeling Techniques

This chapter deals with other possibilities of modeling the technical system, particularly the double-acting pneumatic piston. Physical modeling and data-driven modeling methods were examined in terms of suitability for applying FDI and PdM strategies.

5.1 Physical Modeling

Physical modeling operates with models with a compiled layout that matches the structure of the different physical domains. In this type of software, it is possible to combine different domains to create a complex system model.

Matlab/Simulink provides a physical modeling library, Simscape [16], that meets the above specifications. Using Simscape software, the user combines a model from different blocks representing different physical functions (spring, resistance, hydraulic valve), and connection links represent some types of energy flow.

5.1.1 The double-acting pneumatic piston modeling in Simscape

In this part, the same assumption applies as in section 4.1. All the processes take place adiabatically, i.e., without heat exchange with the environment.

The resulting model was compiled using gas and mechanical domains 5.1.

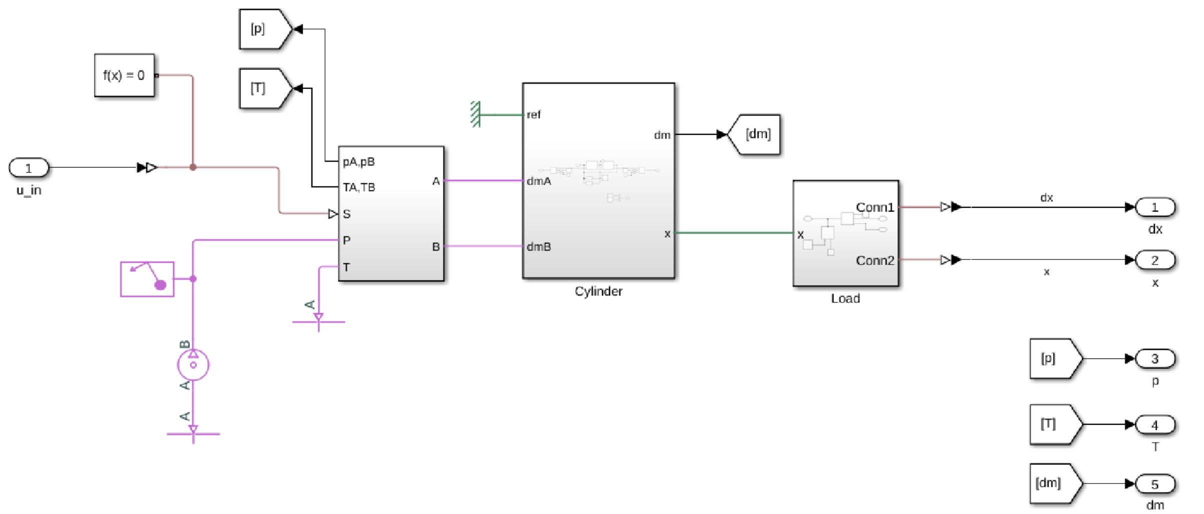


Figure 5.1: The double-acting pneumatic piston developed using Simscape software

5.1.2 Limitations

It is necessary to know well the parameters of the system.

For example, we need to have a precision-measured characteristic of flow control valve adjustment in the form of a lookup table to use a throttle valve block.

Providing simplification and reduce the model to the only control valve, there are still a few parameters that are not available such as valve and dampers coefficients mentioned before.

The main problem is the computational complexity of the model compared with the first principle model. During the parameter estimation, the first principle model is much faster than the Simscape model and gives an option to experiment with different fault states analysis.

However, both models showed quite close behavior during testing with the same parameters.

5.2 Data-Driven Models

Data-Driven modeling explores collected measured signals to identify the system structure or learn the system behavior from data [17].

Between data-driven common models belongs parametric and non-parametric models. Parametric models take part in the system identification field. A collection of different generalized mathematical models can be fitted to the input-output signals pair, such as transfer functions, polynomial models, non-linear ARX models, etc. A typical representative of non-parametric models are neural networks of various structures. In this thesis, experiments on test datasets were performed with both types of models.

5.2.1 Hammerstein-Wiener Model

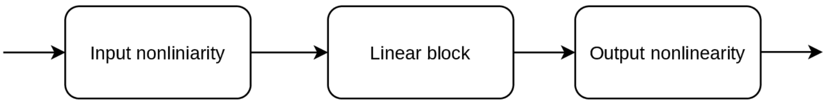


Figure 5.2: Hammerstein-Wiener model structure

The best results between parametric models using System Identification Toolbox, shown Hammerstein-Wiener Model. The model consists of three blocks 5.2, input nonlinearity, linear block and output nonlinearity. The nonlinearities are represented by different functions such as dead-zone, polynomial estimator, saturation, wavelet network function, etc.

However, using the identified model, adequate behavior to the measured data was achieved only for the position signal 5.3. The model identified for velocity signal did not show acceptable behavior 5.4. The reason is the significant nonlinearity and complexity of the system, which the simplified models cannot take into account.

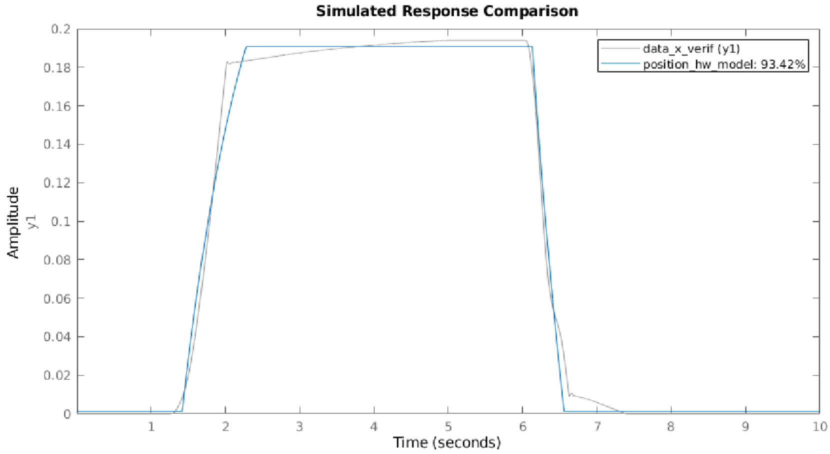


Figure 5.3: Simulated Response for Position Signal Comparison

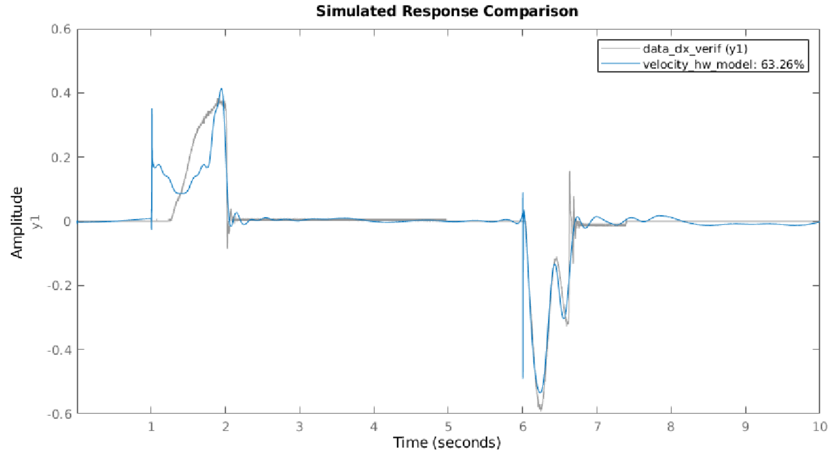


Figure 5.4: Simulated Response for Velocity Signal Comparison

5.2.2 NARX Model

Different structures can be used to train the neural network to predict system behavior. The most common way is using the nonlinear autoregressive with the external input model (NARX) [17]. This model predicts time-series data by using different numbers of time-delayed values of input and output signals 5.5.

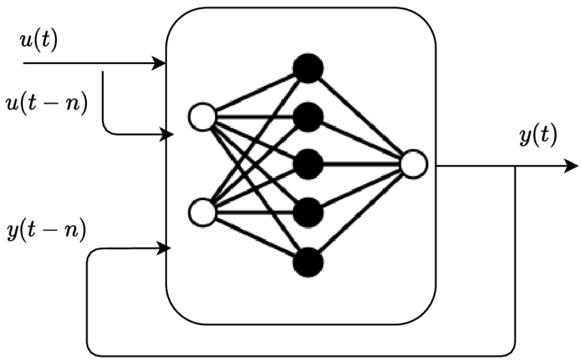


Figure 5.5: Schematical representation of NARX model

During the development of the model, it is necessary to pay attention to overfitting, which can significantly impair the performance of the model and its generalization capabilities.

Some experiments have been performed with this modeling approach. The Neural Network can predict the behavior of the system based on input.

6 Models Comparison

As mentioned earlier 2.4, the simulation model can be used in several situations. Models of the normal condition can simulate system output to a given input in normal operating conditions. This type of model can be used to provide, for example, residual estimation. Compare normal condition model with measured signals from sensors decision algorithm can evaluate possible faults.

Suppose the model can simulate the system in different conditions. In that case, it gives an option to implement "What-If" simulations and prevent fault situations that are not captured in the measured dataset.

No best solution would apply in all situations, but for a specific example of the double-acting pneumatic actuator with the measured dataset, the more efficient model can be evaluated. Table 6.1 represents the comparison simulation models in 4 categories, simulation speed, accuracy concerning the actual model, the difficulty of deploying the model, the behavior under normal conditions and the possibility of simulating abnormal "What-If" situations.

The speed of the simulation or calculation complexity performs a more prominent role in the model's design, especially during the estimation of the parameters, where the simulations are performed hundreds of times in a row.

model	**speed**	**accuracy**	**normal cond.**	**abnormal**
FPM	fast	normal	yes	yes
Simscape	low	normal	yes	yes
HW model	fast	very low	-	-
NARX	fast	high	yes	-

Table 6.1: Models developed by different approach comparison

Due to the above facts, further work was continued with the help of the first principles model, and the development of the other models was suspended. The first principle simulation model will be used in the next chapter 8, PdM using Simulation Model. All models can be found in the attachment *models*; using scripts *first_principle_model_perfomance.mlx, data_driven_model_perfomance.mlx*, models can be explored interactively.

7 Signal-Based PdM

This chapter introduces the signal-based method applied to a measured dataset. The whole solution procedure will be presented on the example of the development solution on the flow sensor 7.3. Most of the methods used in this chapter are closely related between FDI and PdM approaches. These methods work directly with the measured signal by extracting condition indicators and training the classification model. It is possible to do fault detection and classification using this model.

7.1 FDI methods

There are simple solutions that offer themselves. For example, the proximity sensor can be used to monitoring whether the actuator has reached the position in the expected interval or not. Based on these data, it can be concluded whether the device performs its function or a fault has occurred. Similarly, we can monitor the flow course, and if this course exceeds any given threshold, then a fault has occurred [9]. Using more complex methods, we can not only show the occurrence of faults but also classify the cause. The implementation of these algorithms will be further discussed in this chapter.

7.2 Data Management and Preprocessing

Before the final solution was developed in the whole dataset, the smaller dataset was used for experiments and planning algorithms.

7.2.1 Data Storage

Manage Data First, a folder structure was created to collect all measured and calculated data. The measured signals were given in 6 large files with a ".mat" extension and divided into smaller files with only one measurement each. Data files have been reshaped to Data Ensembles [1] format used for Condition monitoring purposes. This format allows processing data without copying the whole dataset to memory at once but processes them one by one. In large datasets it gives an option to manipulate with data without problems with allocated memory. The full dataset contains 4840 measurements. Each measurement includes a 10-second recording of all signals collected from moving the piston up and down.

Labels The whole dataset was divided into 20 Labels by place of fault accumulate:

- Healthy
- Throttle valve 1
- Throttle valve 2
- Small damper bottom

- Small damper upper
- Large dampers
- And combinations of these faults

7.2.2 Data Exploration

Data from each of the eight sensors 3.1 were explored in an attempt to find measurement errors or anomalies in data. Figure 7.1 shown an example of the flow signal in different operation conditions.

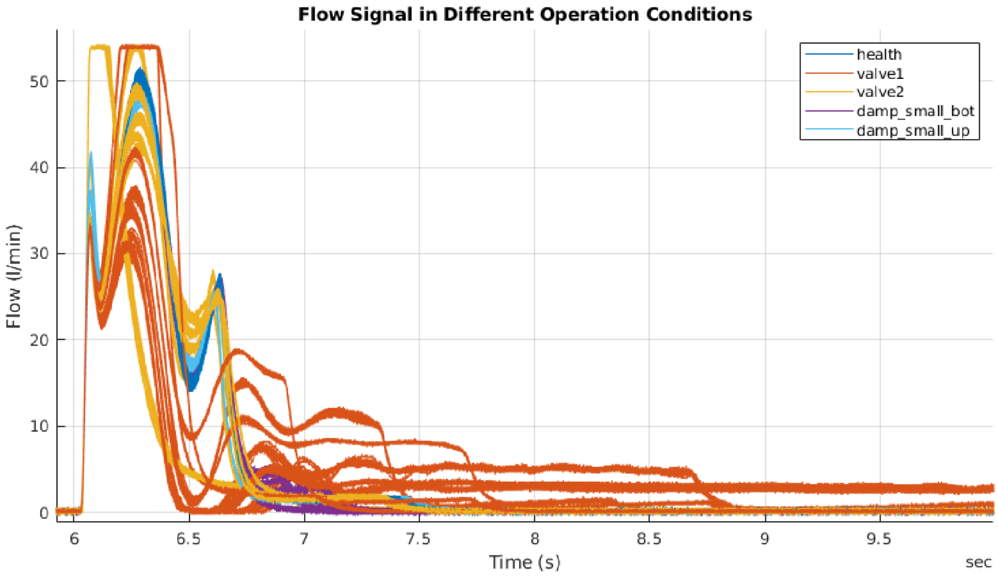


Figure 7.1: Flow Signal in Different Operation Conditions

7.2.3 Preprocessing

After the data has been processed and organized in one datastore, the possibility arises to perform signal preprocessing. Preprocessing includes smoothing, filtering, detrend the signal, and missing value removal [12].

The datastore contains some signals, such as an encoder, that is very accurate. There is no preprocessing needed to apply. Signals noisier (pressure signal or strain) have to be preprocessed and applied algorithms to noise reduction such as smoothing and filtering concerning the preservation of the information base. However, during experiments turned out that non preprocessed signals have better performance. For example, the preprocessed pressure classification model gives 78 % accuracy; model trained on CI from the raw pressure signal offers approximately 82 %.

7.3 SB Methods and Flow Sensor as an Example

In this section, signal-based methods were applying to the flow sensor as a case study example. The rest of the sensors was processed in the same way; however, each required an individual approach.

7.3.1 Flow Sensor Data

There are two flow signals in the datastore. Both are connected to port A in scheme 3.1. Signals were sampled in 1kHz frequency; thus, in 10 seconds, there are 10000 points measured.

- Flow Extrusion
- Flow Contraction

7.3.2 Condition Indicators Extraction

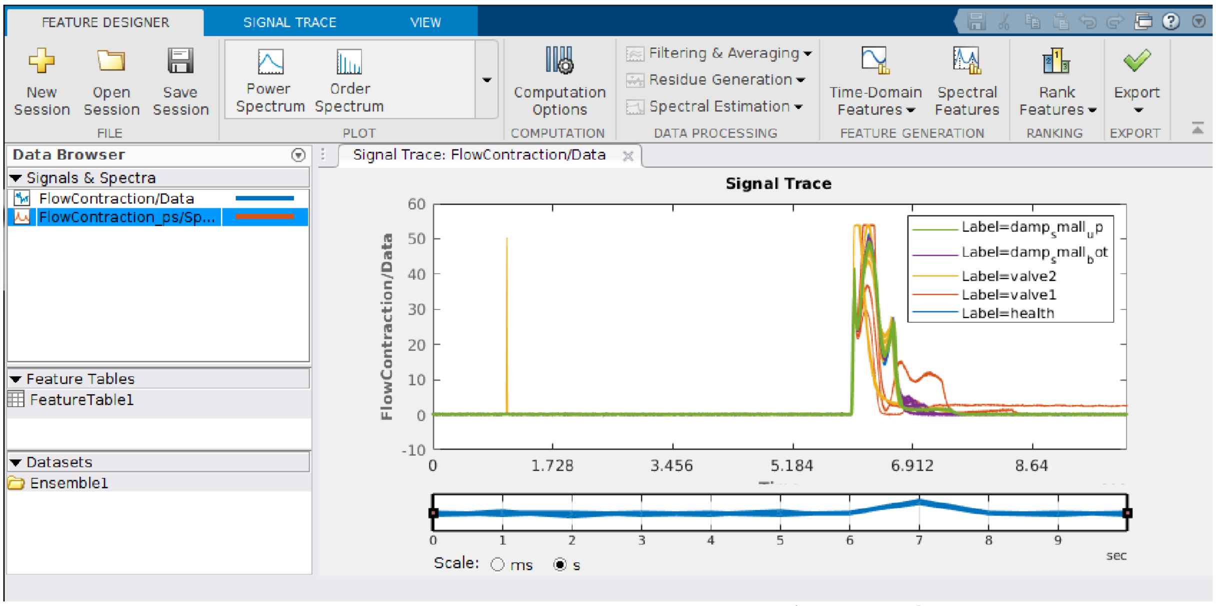


Figure 7.2: Diagnostic Features Designer App Interface

One of the reasons to use Matlab Data Ensemble format to manage the data instead of others is to use the Diagnostic Feature Designer App (fig. 7.2) [4]. This app provides an intuitive environment for extracting both statistical condition indicators and power spectral density calculations with the following extraction of frequency condition indicators. It is also possible to generate Matlab functions to deploy the algorithms on a bigger scale.

Statistical Condition Indicators For every flow signal in the dataset, statistical condition indicators were calculated [7]:

- Mean
- Standard deviation
- RMS
- Peak value
- Kurtosis

- Clearance factor
- Crest factor
- Impulse factor
- etc.

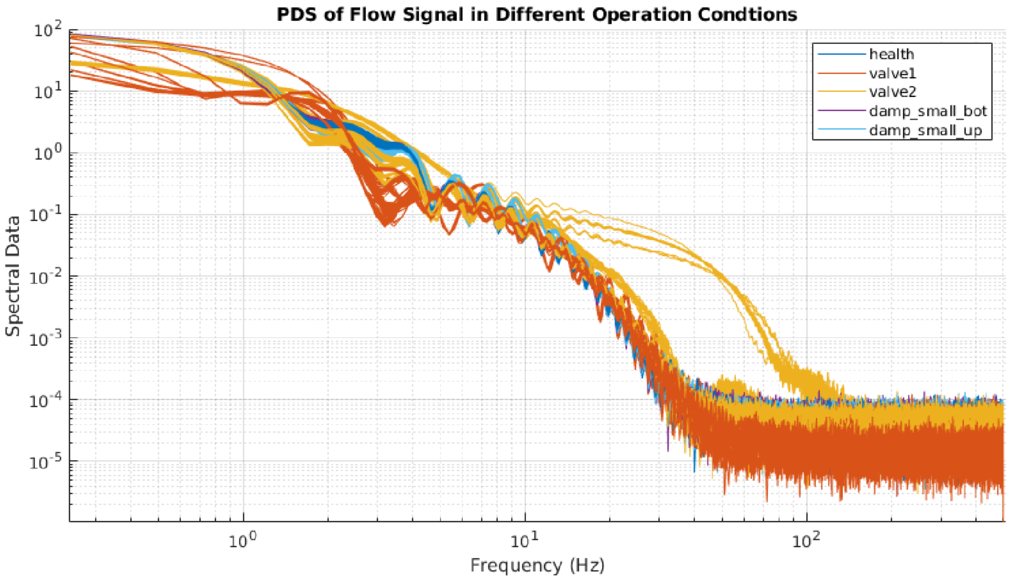


Figure 7.3: Welch's Power Spectral Density of the Flow Signal

Frequency Domain Condition Indicators Using Welch's power spectral density estimation 7.3, frequency CI were calculated [12]:

- First five peaks amplitude
- Peaks frequencies
- Spectrum band power

Extracted condition indicators were written to files with signals and easily acceptable. After each data file contains complete information about one measurement:

- Measured signals
- Setting parameters (valves, dampers, load)
- Power spectrum calculated from measured signals
- Statistical and Frequency features extracted from signals

Moreover, a table was created, which contains all condition indicators extracted, to prepare the train and test dataset for the classification model.

7.3.3 Condition Indicators Ranking

The table of calculated condition indicators contains 25 statistical and frequency CI. To train a classification model, it is good practice to reduce the number of features or transform them with PCA algorithm and use only first n principal components, to remove linearly dependent condition indicators. According to section 2 Analysis of Variance (ANOVA), specifically in our case Kruskal – Wallis one-way ANOVA algorithm was used.

The result is a sorted table 7.1 of condition indicators depending on how much variance a particular condition indicator can describe in the dataset.

	Features	Kruskal-Wallis
1	FlowContraction_ps_spec/PeakAmp1	1.4815e+03
2	FlowContraction_stats/CrestFactor	967.6028
3	FlowContraction_ps_spec/PeakAmp3	865.7571
4	FlowContraction_stats/Mean	567.6620
5	FlowContraction_ps_spec/PeakAmp4	460.0924

Table 7.1: First Five Ranked Condition Indicators using ANOVA

Figure 7.4 shows the scatter plot of the first three condition indicators for normal behavior and fault condition caused by the change of throttle valve 1. The first five condition indicators ranked by the ANOVA algorithm were used for training the final model on all 20 labels.

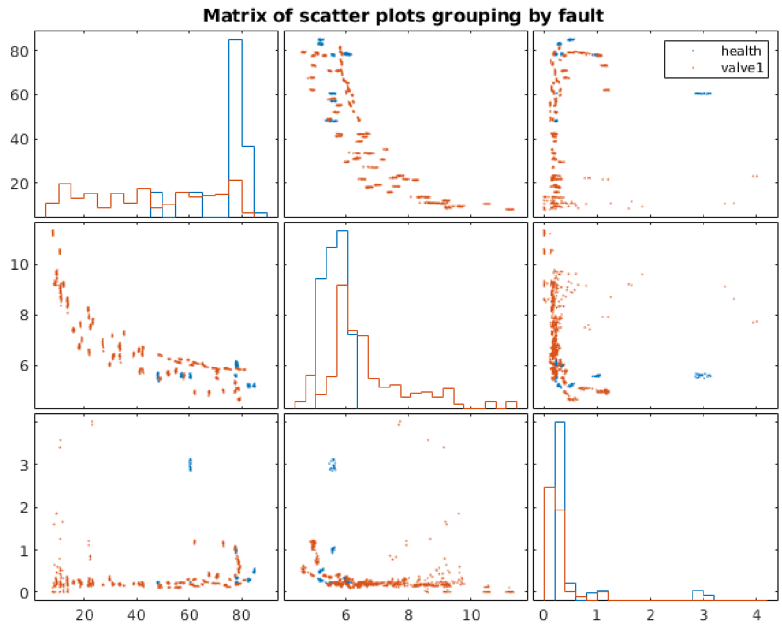


Figure 7.4: Example of Scatter Plot with different CI

7.3.4 Train Classification Model

The main goal of the classification task is to train a model that can predict the fault code or label signalized about pneumatic actuator behavior by calculated condition indicators.

There are many classification models, but it is best to try different variants and be satisfied with the best result from a practical point of view. The Classification Learner App from the Machine Learning Toolbox [18] tool can be used for experiments and iterative tuning of different condition indicators and classification models. It is possible to try several models, apply the PCA algorithm, interactively draw Scatter plot and Confusion Matrix, and generate functions for practical applications.

Train, Test Datasets By splitting data to train and test datasets, we can ensure that the training model outcomes are valid. The cross-validation resampling procedure to prevent model overfitting was used during the model fitting.

Classification Model Performance Trained classification models show excellent results on the test dataset for all three situations: using all CI, after applying the PCA algorithm and using the first five CIs recommended by the ANOVA algorithm. The accuracy evaluations of the models are shown in Table 7.2.

approach	**model**	**accuracy** [%]
all features	Bagged Trees	99.45
PCA	Bagged Trees	95.18
ANOVA	Fine KNN	97.52

Table 7.2: All Features vs PCA vs ANOVA perfomance

Figure 7.5 shows the confusion matrix from the Fine KNN classification model by training on data using the ANOVA algorithm. From the confusion matrix, it is clear that combined faults in the dataset were not observed much. However, the model can successfully resolve these fault conditions too.

From a practical point of view, in this particular case, the use of the ANOVA algorithm allows not only to reduce the number of CIs for prediction on the model but also to calculate from the signal, not 25 CIs but only 5.

Considering this fact, deploy this algorithm on a bigger scale on many devices, where the calculation complexity plays a role, using the ANOVA algorithm is justified.

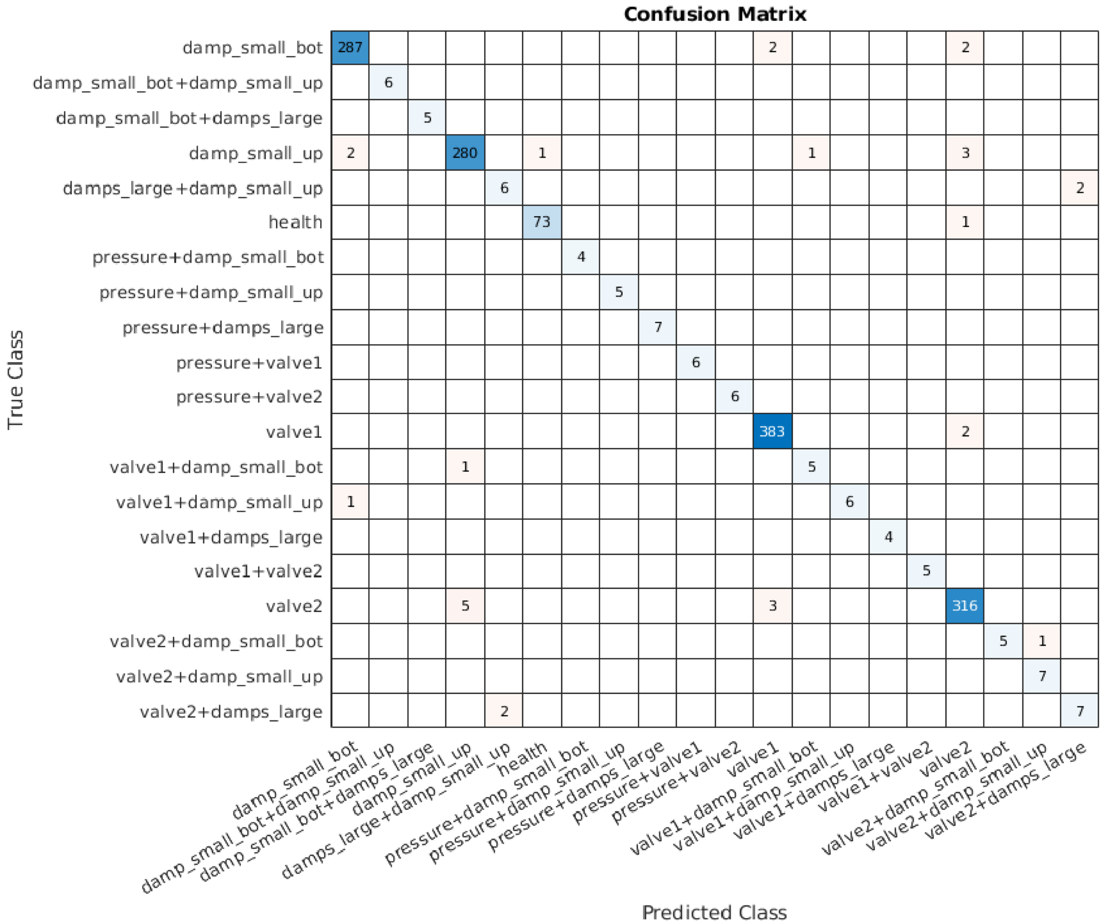


Figure 7.5: Fine KNN trained on ANOVA Dataset Confusion Matrix

7.4 Summary All Sensors Comparison

Surprisingly, all sensors showed satisfactory results on the measured dataset. Processing the entire dataset is a very demanding operation in terms of calculation. Therefore, only the final solutions were added to the attachments. The results for all sensors can be verifying by running matlab-live-script *sb/signal_based_live.mlx*.

Table 9.1 compares all the sensors used in terms of the accuracy achieved on the test datasets and the approximate prices of the sensor itself taken from open sources. Graph 9.1 visualizes these data. Here are some notes on each of the sensors.

Sensor	Acc	Encoder	Flow	Mics	Pressure	Proximity	Strain
Accuracy [%]	91.6	96.1	97.2	95.8	76.6	80.5	95.0
Cost [czk]	2x 3500	25000	6000	3x 500	1000	2x 1000	15000

Table 7.3: Comparison of sensors from accuracy/cost perspective

7.4.1 Temperature sensor

Temperature sensors do only one measurement during the experiment. These values can be represented as condition indicators without any manipulations. Plotting data from the dataset 7.7 shows that they correlated to an ambient temperature that is different in

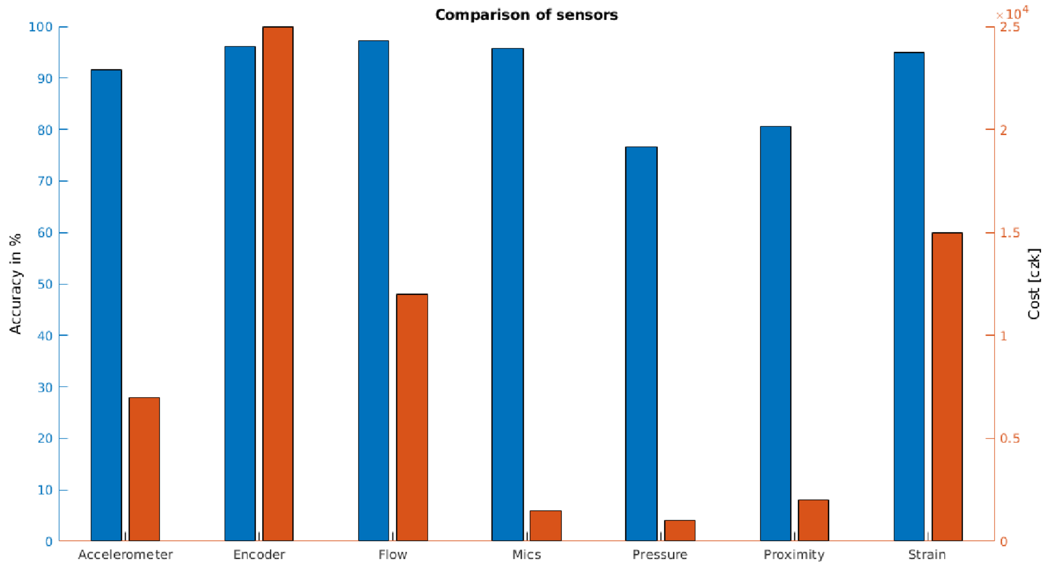


Figure 7.6: Comparison of sensors from accuracy/cost perspective

various measurement days. These data are sensitive to ambient conditions and measured data, not representative. The classification model trained on this data not robust in real life.

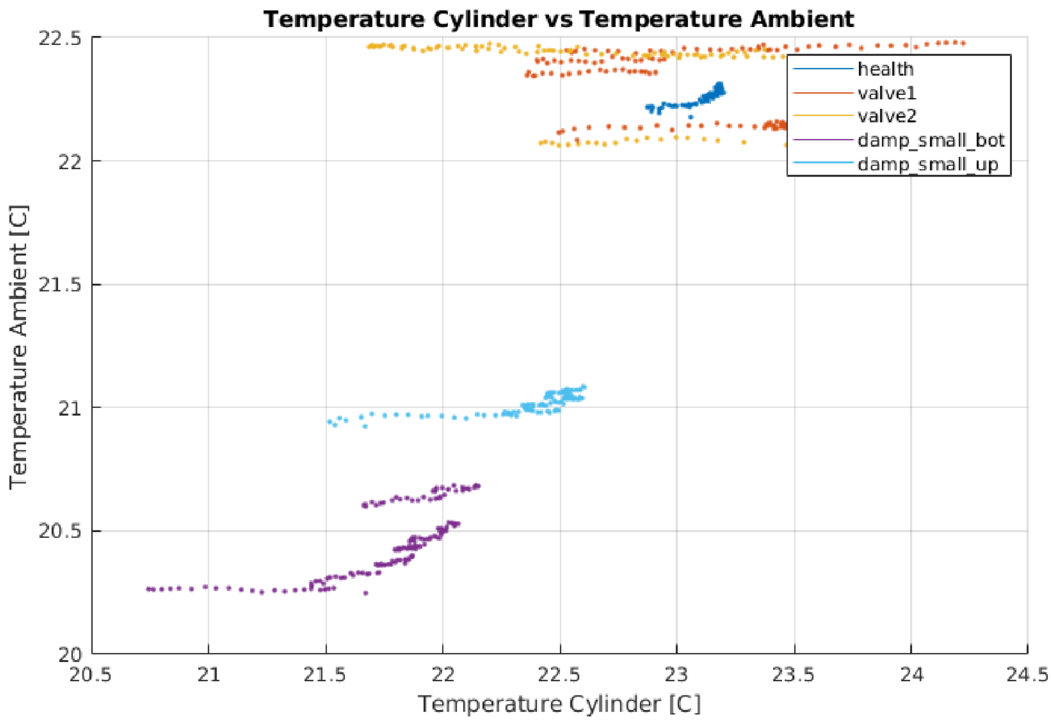


Figure 7.7: Scatter plot of temperature measured data

7.4.2 Encoder

A linear magnetic encoder is a perfect development sensor-tool for understanding system behavior and algorithm design. Up to three signals, displacement, speed, acceleration, can be available from one sensor. The trained classification model shows perfect results. From a practical point of view, the financial cost of purchasing, installing, and maintaining the sensor is unsuitable compared to cheaper sensors with similar prediction accuracy.

7.4.3 Microphones

Cheap, good results, but maybe problems with real life integration (noise from another machines). Another problem cannot be modeled in simulation system. For predictive purposes require data from real model.

7.4.4 Accelerometers

There are two accelerometer sensors. Each sensor contains two signals on the x, y-axis. One sensor is placed on the movable part of the stand device; the second is on the statical part without movement and measure only vibrations. Sensors show good accuracy; choosing one of the two accelerometers, the static one, is preferable.

7.4.5 Proximity Sensors

As mentioned before, proximity sensors can be used for simple inspection purposes 7.1. Proximity sensors are digital and provide only statistical condition indicators; from statistical CI offers valuable information, only a few CI's due to signal shape.

7.4.6 Flow Sensors

Flow sensors achieve the best results. It is possible to achieve ≈ 97 % accuracy using only one sensor. If the practical application requires maximum accuracy, the flow sensor is the best candidate. Nothing less in terms of price is an expensive sensor.

7.4.7 Air Pressure

The pressure sensor measures the pressure in the reservoir. Data from this sensor is not fully informative for the possibilities of predicting and identifying a fault condition. This sensor showed low accuracy compared to the others. From an economic view, combining a pressure sensor with another sensor does not make sense due to existing sensors such as microphones that are better from an accuracy/cost perspective.

7.4.8 Strain Gauge

Strain Gauge showed excellent results, but in general, it is similar to an encoder because it is an expensive sensor that requires maintenance. From a practical point of view, there are better candidates for industrial applications.

8 PdM using a Simulation Model

This chapter deals with model-based methods and the possibilities of using a simulation model to design and develop a PdM algorithm. A demonstration of the possibility of generating sensor fault conditions is presented in section 8.1. Using identified Hammerstein-Weiner model to extract condition indicators in the form of a dynamic system parameter shown in section 8.2.1. In section 8.3, the simulation model is used as a nominal, and residual estimation is performed with the following training of the classification model. The left sections 8.4 deal with the use of a simulation model to generate degradation data. And the use of a newly generated dataset to estimate the remaining useful life.

8.1 Using Simulation Model to Generate Fault Data

In this section, the simulation model will play the role of a digital twin for experimenting. Digital Twin can be used to model situations that did not capture in the original dataset or if it is hard to model some cases with real-world hardware.

As an example, we can model sensors fault such as sensor drift or complete signal loss. Suppose the simulation model signal is in good agreement with the real-world system. In that case, the generated data can complement the primary dataset, introducing a more significant number of observed fault situations.

8.1.1 Sensor Fault Modeling

Three basic situations measuring the nominal behavior of the system were simulated. By adding measurement noise to the system, a ”noise” fault situation was created. Another modeled case was made using the offset. In Figure 8.1, flow sensor fault condition signals are generated. This straightforward situation illustrates the possibility and simplicity of performing experiments with a simulation model to develop robust PdM algorithms.

The matrix of Scatter plots grouping by faults 8.2a shows how condition indicators are distributed. The data are well separable, which means that these condition indicators are suitable for use in classification. The confusion matrix 8.2b provides 100 % accuracy on test data. Which in this simplified situation is possible. In more complex cases, achieving 100 % is practically impossible, but it is possible to get close.

8.2 Model-Based Condition Indicators

The model-Based approach is suitable when it’s challenging to identify condition indicators using only signals. In some cases, it’s useful to fit some models from data and extract condition indicators as some system parameter [12], [5].

Static models If the system behavior can be identified from the data as a static model, we can extract condition variables from this model as model parameters. For example, if the model is fitted to a polynomial model, polynomial coefficients can be used as condition

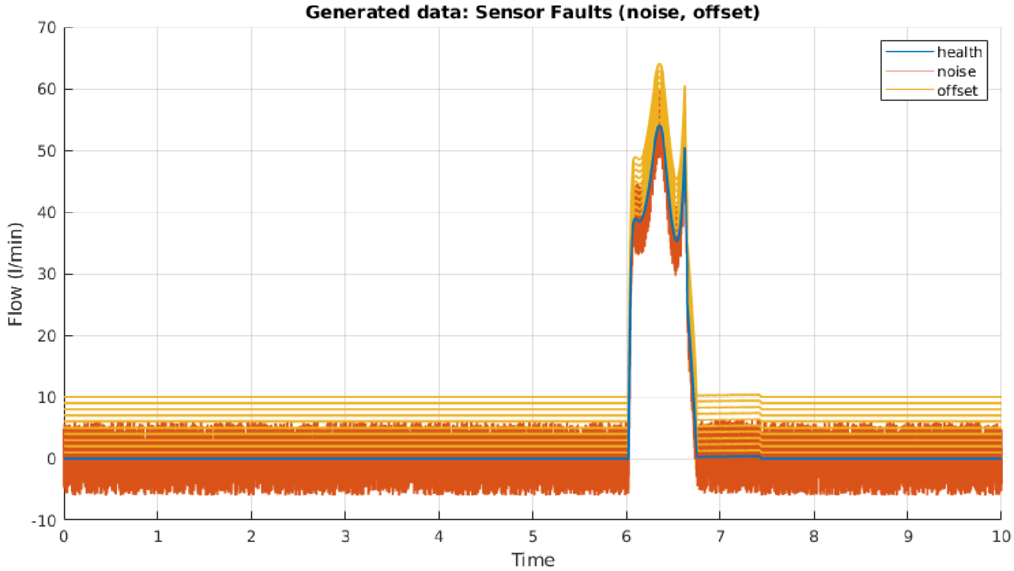
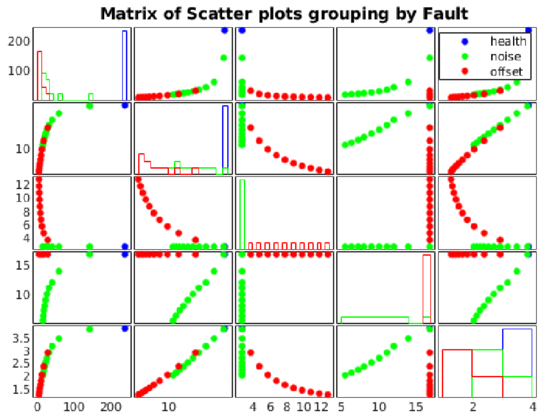
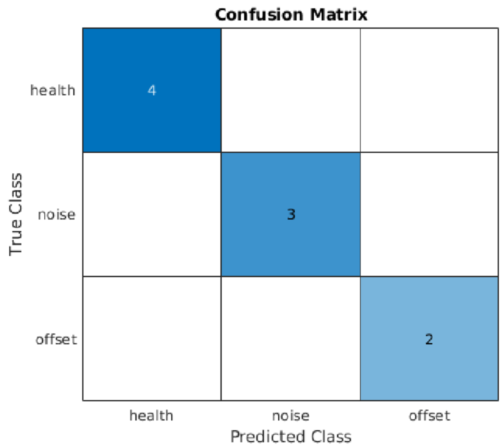


Figure 8.1: Sensor response in different fault conditions



(a) Sensor fault condition indicators distribution



(b) Confusion matrix test dataset

Figure 8.2: Classification performance

indicators.

Dynamic models Signals showing dynamic behavior can be identified as dynamic models such as State-Space or AR, ARX, NLARX (Nonlinear auto recursive model), and so on. Then condition indicators can be extracted as poles, zeros damping coefficients from the identified model.

State obesrvers Another possibility is to use the Kalman filter and other state observers to estimate all state variables from the measured signal. It is suitable if the system's condition is directly dependent on some state variable that is difficult to measure directly [12], [9].

8.2.1 Using Hammerstein-Wiener Model

In this demonstration, the Hammerstein-Wiener Model was used to identify the system using position measurement 8.3. A smaller data set was used for the experiment, which contains 660 measurements, six primary fault states. An HW model was identified for each signal position. Condition indicators were extracted in the form of a system coefficient, both a linear block and an input/output layer.

The training of the classification model was unsuccessful, and the resulting accuracy did not exceed 40 %. Therefore, I consider this approach inappropriate in this particular case.

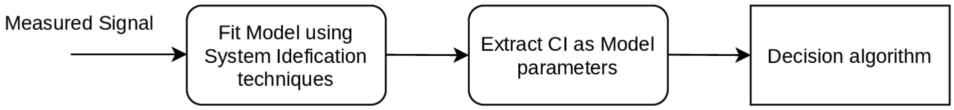


Figure 8.3: Using identification model for PdM workflow

8.3 Using Simulation Model for Residuals Estimation

The residual Estimation approach is another option to use a simulation model to achieve fault detection. The residual is a subtraction of two signals in the form $e(t) = y(t) - \hat{y}(t)$ as shown in 8.4.

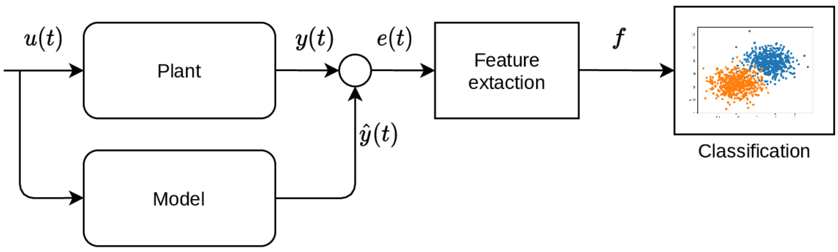


Figure 8.4: Residual estimation diagram

Residual estimation can be helpful when the system response is highly dependent on the input signal, and the measured dataset does not observe all possible faults. Residuals are very sensitive detectors of problems. In some cases where the system changes operation conditions but still operates in a healthy state and this change does not reflect the nominal simulation model, the decision algorithm may signalize a fault. This type of fault, also known as false positive, indicates problems that do not exist [11]. Generally, this approach requires a smaller amount of data for training the classification model. It is very suitable for system monitoring, where if the residual of two signals outreaches any given threshold, a fault state has occurred [11], [12], [9].

To demonstrate residual estimation and save calculation time, a smaller dataset was used. Since the signals represent the same 10-second intervals, the simulation was performed only once and then used as the nominal reference behavior for all calculations of all residuals. However, for deploying this algorithm, the simulation model runs in real-time and continuously generates residuals.

A linear encoder was used as an example. Figure 8.5 shows the residual for the measured and reference signal. These residual signals were then combined to the dataset from which condition indicators were extracted as statistical parameters. Using the same

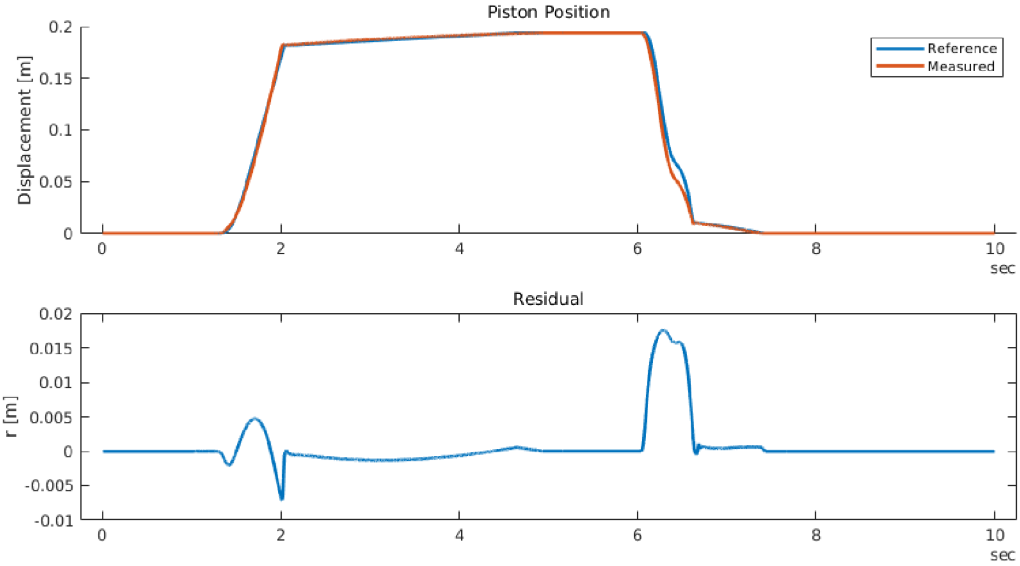


Figure 8.5: Residual signal of measured and simulated position

steps described in the signal-based example 7.3, condition indicators were ranked 8.1, and the classification model was trained. The trained classification model shows excellent accuracy of 99.49 %. Predictions are shown in confusion matrix 8.6.

	Features	Kruskal-Wallis
1	LeverPosition_res_stats/RMS	543.82
2	LeverPosition_res_stats/PeakValue	271.94
3	LeverPosition_res_stats/Std	222.89
4	LeverPosition_res_stats/THD	215.34
5	LeverPosition_res_stats/Kurtosis	129.66

Table 8.1: First Five Ranked Condition Indicators using ANOVA

For comparison using the signal-based method applying to the same dataset, classification results are similar 99.49 %. By given the results, the residual estimation method may seem unnecessary. In this particular case, from a practical point of view, there is no improvement of the result, but the calculation time increases significantly.

8.4 Using Simulation Model to Generate Prognostic Data

Another option is to use a simulation model to simulate a system degradation process. We can evaluate CI from sensor signal by changing a system's mechanical properties as friction or mass flow leakage. Another advantage is that we can design experiments on the model to evaluate what type of data we require from a real-world system to develop a robust algorithm [12].

8.4.1 Air Leak Modeling

One of the common failures in pneumatic actuators operation is air leakage from the chamber where the piston is located. Dust and other contaminants can damage the

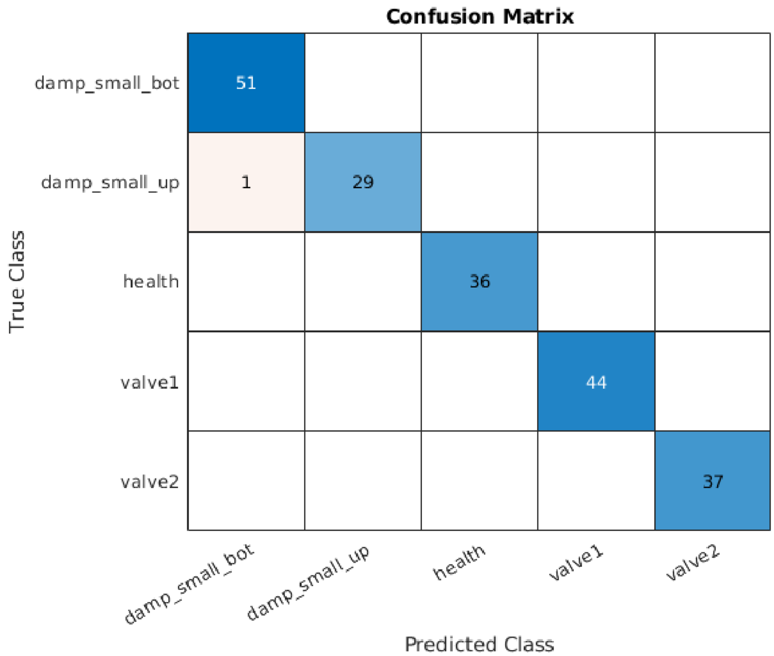


Figure 8.6: Classification model perfomance

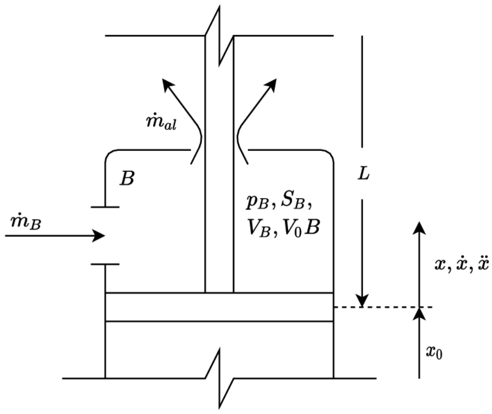


Figure 8.7: Schematic representation of the air leak process

connection between the cylinder and the piston, causing air leakage. This problem is schematically illustrated in Figure 8.7.

In this example, air leakage from the chamber was modeled the same as air expansion from the reservoir, described in section 4.2. Due to the notation in Figure 8.7, the air leakage process describes equation 8.1.

$$\dot{m}_{al} = C_{al} p_B \sqrt{\frac{2}{RT_B}} \cdot \psi \left(\frac{P_0}{P_B} \right) \tag{8.1}$$

Air leakage is reflected in the pressure in chamber B according to the equation 8.2.

$$\dot{p_B} = \frac{\kappa}{S_B(L-x) + V_{0B}} \left(p_B S_B \dot{x} + RT_B [\dot{m_B} - \dot{m}_{al}] \right) \tag{8.2}$$

Figure 8.8 shows the development of pressure in the chamber without air leakage and with a very significant leakage value.

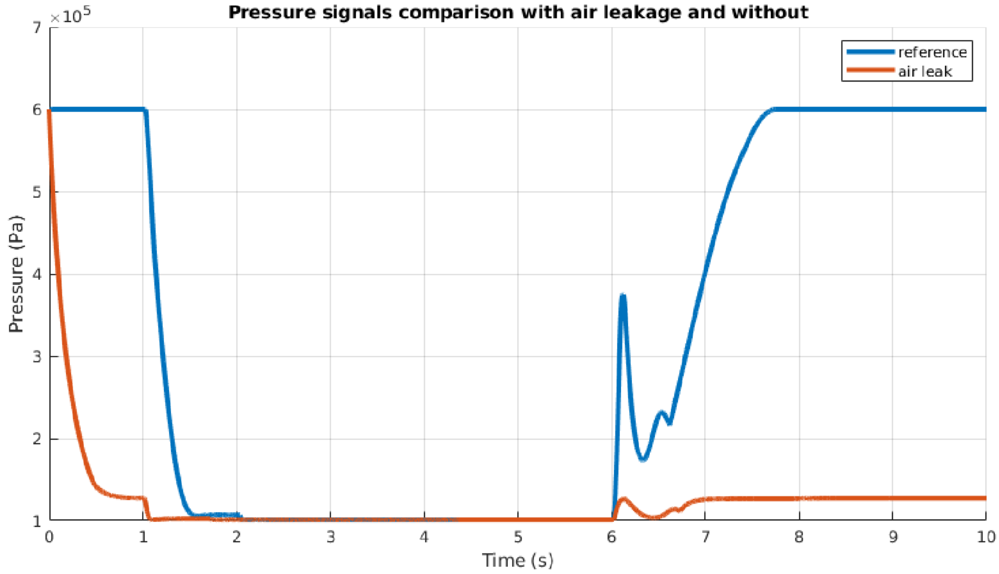


Figure 8.8: Development of pressure in the chamber with air leakage

This fault was modeled on a simulation model with different dynamics of coefficient C development in the range $C_{al} \in (10^{-10}, 10^{-6})$. The following sections describe how this data can be used to design RUL estimation.

8.4.2 RUL

The dataset contains 25 simulations, each with a various number of cycles and different dynamics of air leakage development. After CI extraction in the form of shape factor figure, 8.9 represents the development of each simulation.

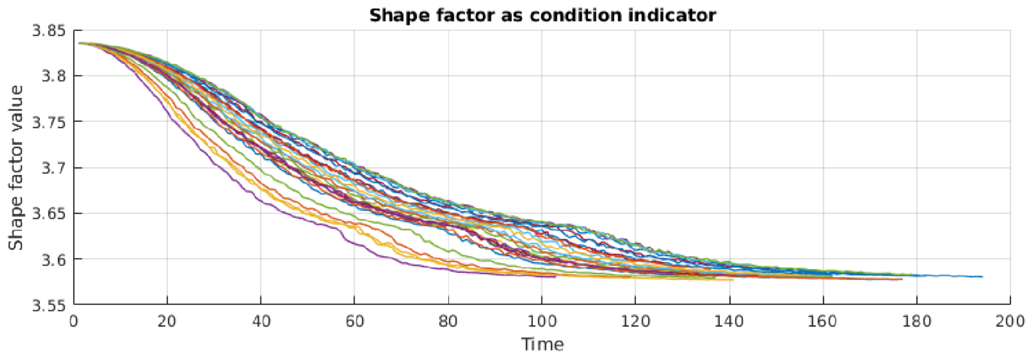
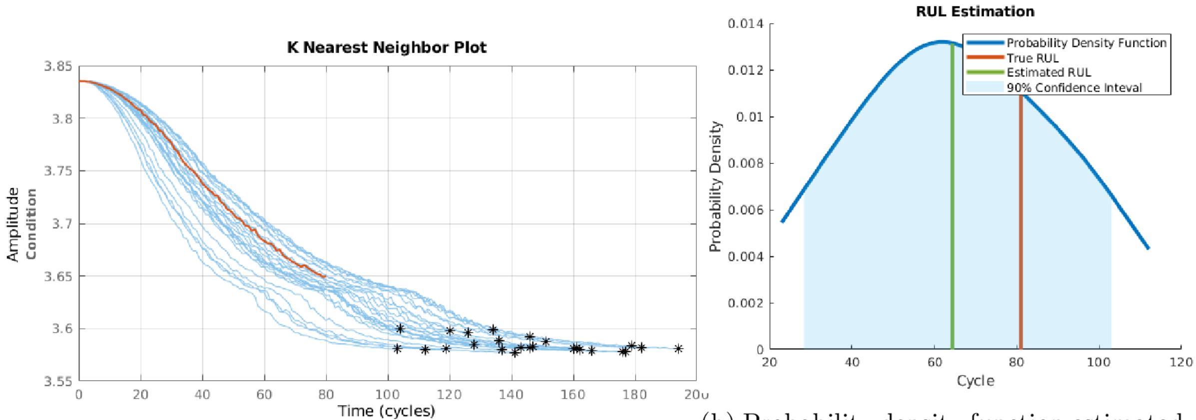


Figure 8.9: Development of condition indicator

Prognostic CI For RUL algorithm development, prognostic CI is used. The prognostic CI can be any parameter that represents the degradation behavior of the system over time. The monotonicity test can be used for ranking prognostic CI. The shape factor was selected during the design, but more CIs showed promising results in this particular case.

RUL Models Residual similarity, pairwise similarity and linear degradation models were used for data experiments.

Figure 8.10 presents results of the residual similarity model, results of RUL estimation



(a) Test vs train path development

(b) Probability density function estimated RUL

Figure 8.10: Residual similarity model performance

satisfying. Pairwise similarity model finding degradation path that is the most correlated to test data. The residual similarity model fits an ARMA (Autoregressive Moving Average Model) model on the train data and then computes the residuals between predicted data from the ARMA model and the test data [12]. The pairwise similarity model on the generated dataset shows similar results as the residual similarity model.

Due to figure 8.9, we can determine a safe threshold that we do not want to exceed and then use the degradation model. In this case, a linear degradation model was used. This model creates a linear degradation profile to evaluate the RUL [12]. The results of the linear degradation model are pretty good 8.11. Predicted RUL shows a deviation from the true RUL of about 10 %, which is more than sufficient in this case.

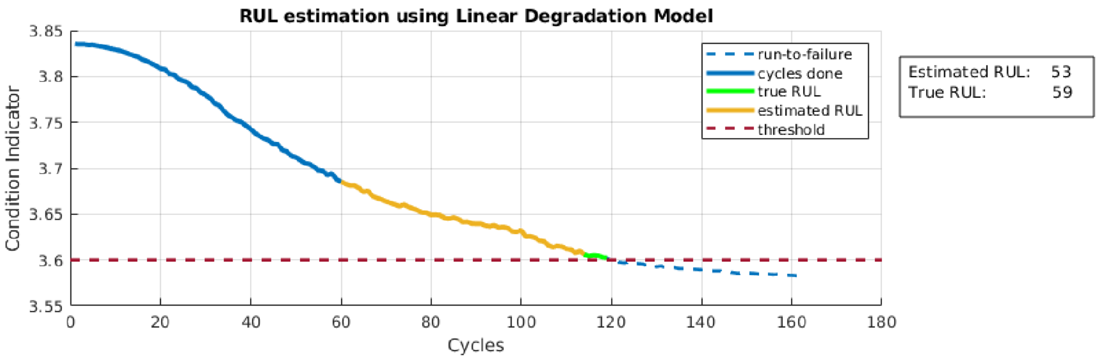


Figure 8.11: Linear degradation model performance

8.5 Summary

A simulation model is a powerful tool for the development of the PdM algorithm. The possibility of generating unavailable or difficult to collect data gives an advantage for implementing robust and efficient algorithms. Since the signal-based method has shown perfect results on the pneumatic pistol application, using model-based methods such as a residual estimation seems unnecessary. All results are available in *mb/mb_.mlx, mb/rul.mlx*

9 Conclusion

The goal of this thesis was to demonstrate and verify fault detection and predictive maintenance techniques on the double-acting pneumatic piston assembly as a case-study object.

9.1 Simulation Model

One of the outcomes from the thesis is a simulation model of the double-acting pneumatic piston system built based on differential equations from the pneumatic-mechanical domain, modeled and developed using Matlab/Simulink software. The simulation model was estimated with parameters of healthy system behavior. However, there is an option to reestimate parameters to fault state and simulate the system in a fault condition.

Due to the available measured data and significantly nonlinear dynamics of the system, the simulation model shows good agreement with the measured data. In contrast to the model built using Simulink/Simscape library, it is distinctly less computationally expensive while maintaining numerical stability. These facts are fundamental when parameter estimation is in progress.

The simulation model was used to experiment with the system's behavior in different conditions, model fault situations and generate data to design and develop robust predictive maintenance algorithms.

9.2 Signal-Based PdM

Another outcome is verifying the possibility of classification and detection of a fault condition applying predictive maintenance techniques, using signal-based and model-based methods.

The experiments were performed on a dataset measured on a demonstration device using seven types of sensors.

A signal-based method is based on the extraction of useful information directly from the signal in time-frequency domains. Each sensor required an individual approach for preprocessing, extracting features, ranking features and building the classification models. But generally, there is minimal preprocessing needed to keep the possible helpful information.

The table 9.1 contains the comparison of sensors in 2 categories, accuracy performed in the test dataset and sensor cost. The graph 9.1 visualizes these data.

Surprisingly, all sensors showed an accuracy of more than 75 %. Microphones offer excellent performance from a cost/accuracy perspective, and they are suitable for installation and maintenance.

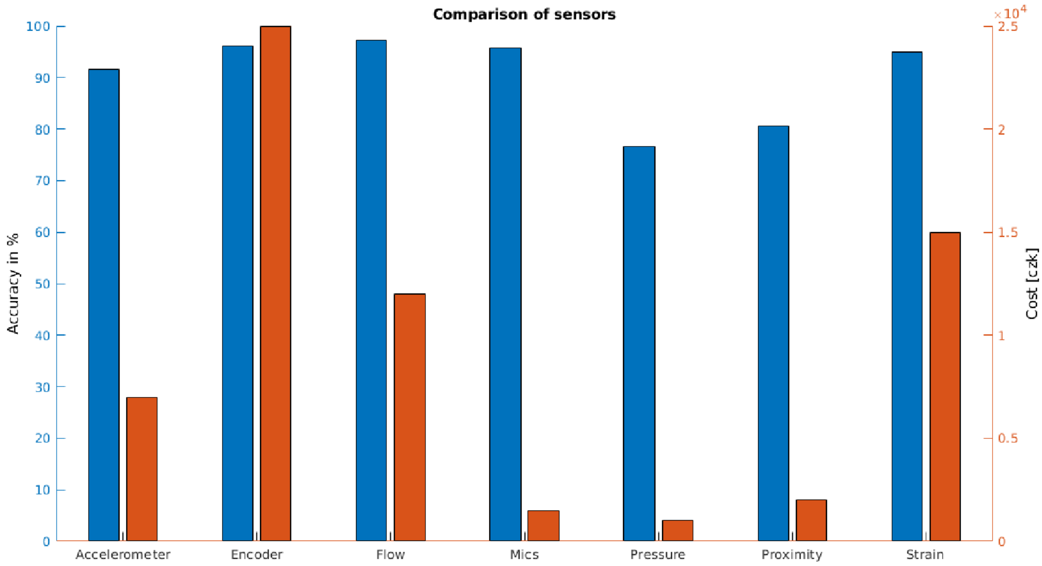


Figure 9.1: Comparison of sensors from accuracy/cost perspective

Sensor	Acc	Encoder	Flow	Mics	Pressure	Proximity	Strain
Accuracy [%]	91.6	96.1	97.2	95.8	76.6	80.5	95.0
Cost [czk]	2x 3500	25000	6000	3x 500	1000	2x 1000	15000

Table 9.1: Comparison of sensors from accuracy/cost perspective

9.3 Model-Based PdM

The next part of this thesis was to apply model-based methods and using a simulation model for predictive maintenance algorithms. These algorithms are practical when it is hard to extract useful information using a signal-based method. Or it is suitable in some cases where we understand the system dynamics and know how to exploit some system variables as condition indicators.

The use of the method of extraction features in the form of a Nonlinear system identification model coefficient, specifically with the Hammerstein-Wiener model, did not give reliable results. Extracted features have no statistical dependence, and it is impossible to predict fault type using this method on the measured data from the pneumatic piston as a case study.

On the other hand, the residual estimation using the simulation model showed excellent results. The measured position signal was compared with the signal from the simulation model in normal behavior. This residual signal was used to classify the fault condition and achieve 99 % on a smaller dataset. But given the results obtained using the signal-based method, the residual estimation method may seem unnecessary. In this particular case, from a practical point of view, the improvement of the result by a few percent does not bring fundamental changes, but the calculation time increases significantly.

The possibility of modeling and simulation sensor faults was also verified using the simulation model. Although it is challenging to collect fault data from the sensor in real-life conditions, fault data can be generated from the simulation model and even combined with the primary dataset to create a synthetical dataset.

9.3.1 RUL

One of the main goals of predictive maintenance is to estimate the remaining useful life. The original dataset does not contain a record of historical data that shows degradation behavior.

A common problem in the maintenance of pneumatic actuators is the leakage of air from the chamber where the piston is located. This situation was modeled on the simulation model and generated data were used for RUL estimation.

The generated dataset contains 25 simulations with different failure dynamics. Each simulation includes a different number of cycles depending on the failure dynamic before the system failure occurs. Each cycle contains a 10-second measurement of the system's response. In the experiment, a flow signal was chosen as an object of interest. From the flow signal, the shape factor parameter was calculated and used as a condition indicator.

The outcome is that it is possible to estimate the remaining useful life on generated degradation dataset by using the residual similarity model, pairwise similarity model and linear degradation model. The prediction results are satisfying; figure 9.2 shows the linear degradation model RUL estimation on the test data.

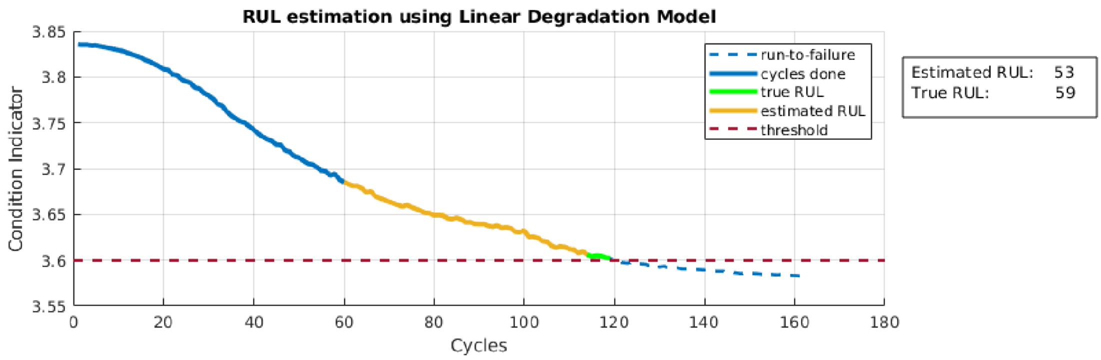


Figure 9.2: RUL estimation results using linear degradation model

9.4 Further Development

As a further development, it would be appropriate to estimate the modeled system parameters piecewise to improve the results, emphasizing the characteristics of throttle valves and dampers with adjustments.

Perform air leak fault condition measurements and collect historical degradation data from a real pneumatic piston. Subsequently, evaluate the dynamics of the failure caused by the air leak. Verify the possibility of estimating the remaining useful life using a flow sensor. It could be an interesting case study to verify a possibility of RUL estimation using microphones. If the performance of the available sensors is deficient, the pressure measurements in the chamber can be performed. The pressure in the chamber is directly dependent on the air leakage from the chamber, as presented in equation 8.2. An example of pressure changes from the simulation model is shown in figure 8.8.

List of Abbreviations

FDA Fault Detection and Analysis

FDI Fault Detection and Isolation

PdM Predictive Maintenance

RUL Remaining Useful Life

FPM First Principle Model

HW Hammerstein-Wiener Model

NARX Nonlinear Autoregressive with External Input Model

ARX Autoregressive with External Input Model

CI Condition Indicator

ANOVA Analysis of Variance

PCA Principal Component Analysis

List of Figures

List of Tables

Bibliography

[1] *Data Ensembles for Condition Monitoring and Predictive Maintenance* [online]. The MathWorks [cit. 2021-5-21]. Dostupné z: https://www.mathworks.com/help/predmaint/ug/data-ensembles-for-condition-monitoring-and-predictive-maintenance.html

[2] DUBRAVKO, Miljković. *Fault detection methods: A literature survey.* 2011.

[3] *Digital Twins for Predictive Maintenance* [online]. The MathWorks [cit. 2021-5-21]. Dostupné z: https://explore.mathworks.com/digital-twins-for-predictive-maintenance/landing-36US-760M7.html

[4] *Diagnostic Feature Designer* [online]. [cit. 2021-5-21]. Dostupné z: https://www.mathworks.com/help/predmaint/ref/diagnosticfeaturedesigner-app.html

[5] *Fault detection and isolation an overwiew* [online]. In: . s. 1-68 [cit. 2020-8-31]. Dostupné z: https://indico.cern.ch/event/459867/attachments/1187105/1721587/FDI_CERN.pdf

[6] HASHEMIAN, H. M. a Wendell C. BEAN. State-of-the-Art Predictive Maintenance Techniques*. *IEEE Transactions on Instrumentation and Measurement* [online]. 2011, **60**(10), 3480-3492 [cit. 2021-5-21]. ISSN 0018-9456. Dostupné z: doi:10.1109/TIM.2009.2036347

[7] *Introduction to Predictive Maintenance with MATLAB* [online]. [cit. 2021-5-21]. Dostupné z: https://www.mathworks.com/content/dam/mathworks/ebook/predictive-maintenance-ebook-part1.pdf

[8] ISERMANN, Rolf. *Mechatronic Systems* [online]. 2005. London: Springer London, 2003 [cit. 2021-5-21]. ISBN 978-1-85233-930-2. Dostupné z: doi:10.1007/1-84628-259-4

[9] ISERMANN, Rolf. *Fault-Diagnosis Systems* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006 [cit. 2021-5-21]. ISBN 978-3-540-24112-6. Dostupné z: doi:10.1007/3-540-30368-5

[10] LUGHOFER, Edwin a Moamar SAYED-MOUCHAWEH, ed. *Predictive Maintenance in Dynamic Systems* [online]. Cham: Springer International Publishing, 2019 [cit. 2021-5-21]. ISBN 978-3-030-05644-5. Dostupné z: doi:10.1007/978-3-030-05645-2

[11] STANLEY, Greg. *Fault Management - the Overall Process and Life Cycle of a Fault* [online]. 2020 [cit. 2021-5-21]. Dostupné z:

https://gregstanleyandassociates.com/whitepapers/FaultDiagnosis/Fault-Management/fault-management.htm

[12] *Predictive Maintenance Toolbox User's Guide* [online]. March 2021. The MathWorks, March 2021 [cit. 2021-5-21]. Dostupné z: https://www.mathworks.com/help/pdf_doc/predmaint/predmaint_ug.pdf

[13] PHILIP J, Pritchard a Mitchell JOHN W. *Fox and McDonald's introduction to fluid mechanics*. 9. Nashville, TN: John Wiley, 2015. ISBN ES8-1-118-91265-2.

[14] *Round cylinders DSNU/ESNU* [online]. [cit. 2021-5-21]. Dostupné z: https://www.festo.com/media/pim/156/D15000100122156.PDF

[15] ŠULC, B. a J. A. JAN. Non Linear Modelling and Control of Hydraulic Actuators. *Acta Polytechnica*. 2002, **42**(3). ISSN 1805-2363. Dostupné z: doi:10.14311/354

[16] *Simscape Model and simulate multidomain physical systems* [online]. [cit. 2021-5-21]. Dostupné z: https://www.mathworks.com/products/simscape.html

[17] System Identification Toolbox. *System Identification Toolbox* [online]. [cit. 2021-5-21]. Dostupné z: https://www.mathworks.com/help/ident/

[18] *Statistics and Machine Learning Toolbox* [online]. [cit. 2021-5-21]. Dostupné z: https://www.mathworks.com/products/statistics.html