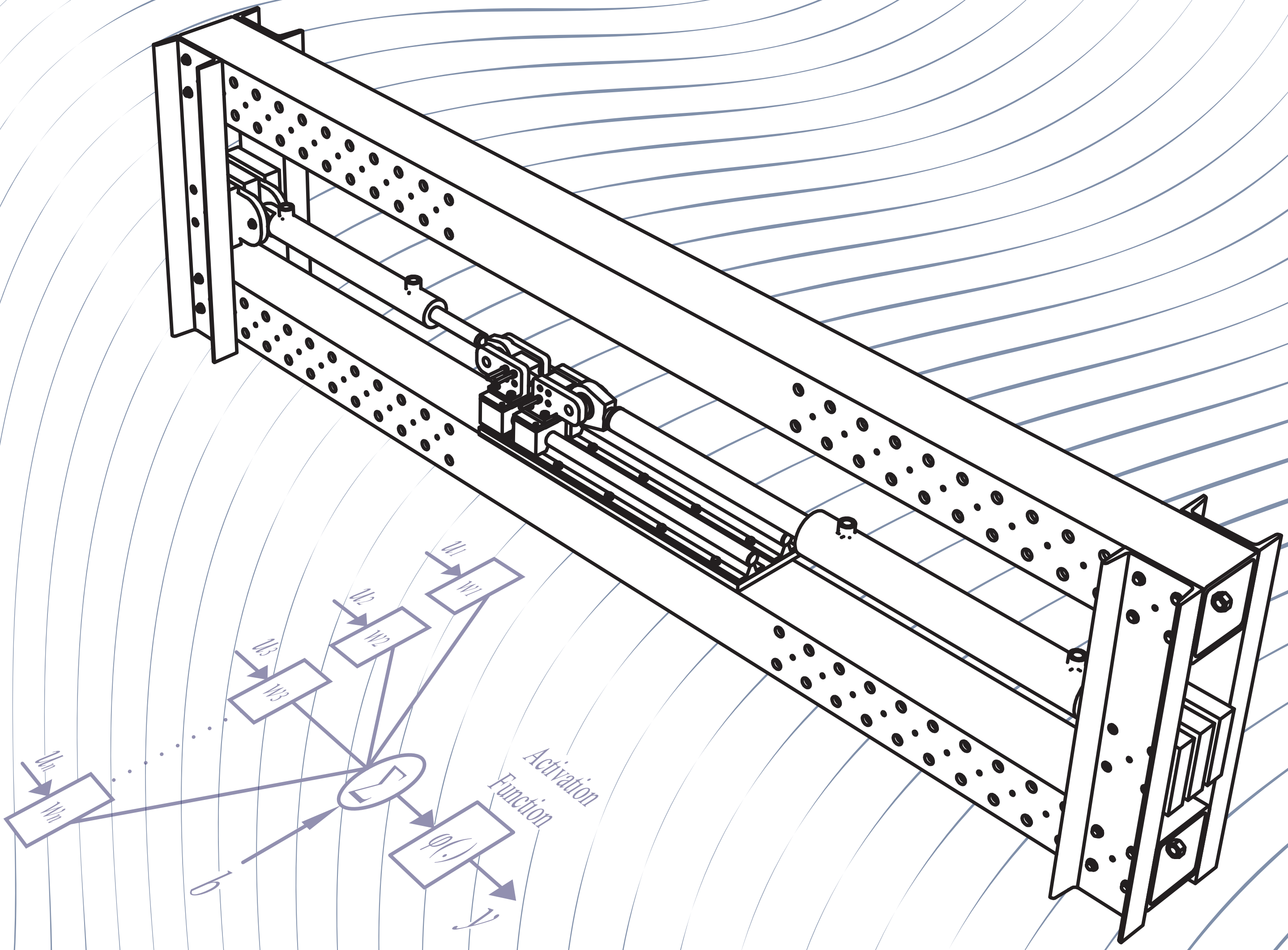
Master's Thesis

A Novel Approach for Fault Diagnosis of Hydraulic Pitch Systems inWind Turbines

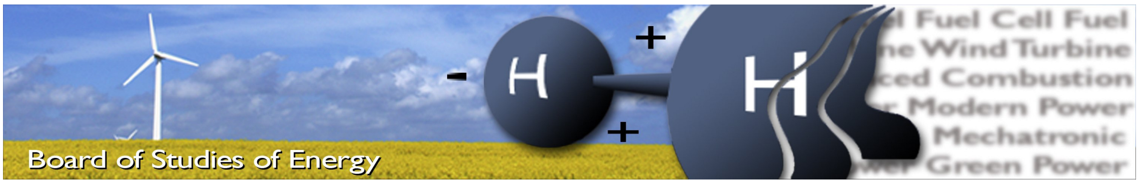


MCE4-1023
Niels Christian Bender
Kris Kristensen Riisager

AALBORG UNIVERSITY
STUDENT REPORT

Aalborg University
Mechatronic Control Engineering

1^{st} June 2016



Title:	A Novel Approach for Fault Diagnosis of Hydraulic Pitch Systems in Wind Turbines
Semester:	10^{th}
Semester theme:	Master's Thesis in Mechatronic Control Engineering
Project period:	01.02.16 to 01.06.16
ECTS:	30 ECTS
Supervisors:	Henrik C. Pedersen & Per Johansen
Project group:	MCE4-1023

SYNOPSIS:

This Master's Thesis investigates fault detection and diagnosis (FDD) in a hydraulic pitch system by applying artificial neural networks (ANN). A mathematical model of the pitch system is developed and the parametrisation of the model is conducted based on experiments. The model is applied for development and testing of several FDD schemes, which are used to evaluate which experimental tests are necessary to perform. A test rig including the main components of the pitch system is constructed in a manner that allows emulation of leakage faults. The test rig allows for general evaluation of the developed FDD scheme. The main conclusion states that a model-based scheme with diagnosis by a focused time-delay ANN shows theoretical superior performance. This is tested on the test rig to validate the scheme. The main results show a decrease of estimation accuracy of leakages when applied on the actual system, however, a stuck servo valve and a pressure transducer failure are successfully diagnosed.

Niels Christian Bender

Kris Kristensen Riisager

Copies:	3+2
Pages, total:	139 (-16 blanks)
Appendices:	A-I

By signing this document, each member of the group confirms that all group members have participated in the project work, and thereby all members are collectively liable for the contents of the report. Furthermore, all group members confirm that the report does not include plagiarism.

Resumé

Formålet med dette projekt er, at udvikle en algoritme til at fejldetektere og diagnosticere (FDD) pitch systemer i vindmøller. Dette ønskes undersøgt, da et sådant FDD system kan øge sikkerheden og mindske udgifterne forbundet med pitch systemer. Det initierende problem er som følger:

Hvordan kan et automatisk fejldetektering og diagnosticerings system designes således, at det kan anvendes på pitch systemer i vindmøller?

Det opsatte problem analyseres ved, at indsamle informationer om pitch systemet og om generel diagnostik teori. Derudover udføres et litteraturstudie af videnskabelige artikler omhandlende fejldetektering- og diagnosticerings- metoder anvendt på hydrauliske systemer. Dette benyttes til, at identificere eksisterende forskning på området.
På baggrund af analysen af det initierende problem, defineres specialets problem formulering til:

Ved at anvende standard kunstige neurale netværk, hvordan kan fejl i et hydraulisk servo system detekteres og diagnosticeres? De betragtede fejl udgør: intern/ekstern lækage strømninger, fastlåst servo ventil samt defekt tryk sensor.

Den anvendte metode til at udvikle et FDD system er først, at opsætte en matematisk model af et hydraulisk testsystem. Denne bruges til, at opnå viden om hvorvidt og hvorledes fejltyperne påvirker systemet. Modellen kan derfor anvendes til, at designe forskellige Kunstige Neurale Netværk (KNN) FDD tilgange ved, at lære et KNN at genkende bestemte mønstre. Derefter udføres indledende teoretiske tests af de to tilgange.

De foreslåede FDD tilgange er baseret på kendte teknikker, men sammensætningen samt valg af de mønstre der ønskes genkendt er ikke foreslået på andre lignende hydrauliske systemer. Gennem de indledende simuleringer påvises det, at bestemmelse af lækage strømninger kan bestemmes, og den højeste nøjagtighed opnås med den model-baserede tilgang.
Da det er bevist teoretisk muligt, at den model-baserede opsætning kan diagnosticere systemet, udvides opsætningen til at inkludere de resterende fejl samt yderligere mønstre. Det konkluderes, at det anvendte NARX KNN som estimator ikke var i stand til, at genkende de ønskede mønstre.
Derfor anvendes en "perfekt" estimator, for at påvise om der teoretisk kan stilles en diagnose af de ønskede fejl under flere test situationer, samt med estimeret målestøj inkluderet. Dette

bekræftes for to ud af tre situationer. Hvor en situation uden bevægelse i systemet førte til problemer.
Den udledte tilstands estimator kunne ikke verificeres i alle situationer på testsystemet, men denne var forsøgt anvendt til sammenligning af en NARX KNN estimator. Det eksperimentelle arbejde begrænses til én test situation der gentages for de fejl der ønskes testet. Sammenligning af den praktiske- og teoretiske nøjagtighed anvendes til at påvise, at en fastlåst servo ventil samt defekt tryk sensor kan detekteres praktisk, hvorimod estimering af lækage strømninger ikke viser samme nøjagtighed i praktiske tests. Dette antyder at der er mulige forbedringer i enten data opsamlingen eller den udviklede FDD algoritme.

De teoretiske hovedresultater af FDD algoritmens nøjagtighed, ved et tilfældigt bevægelses mønster kan ses i fig. 1.

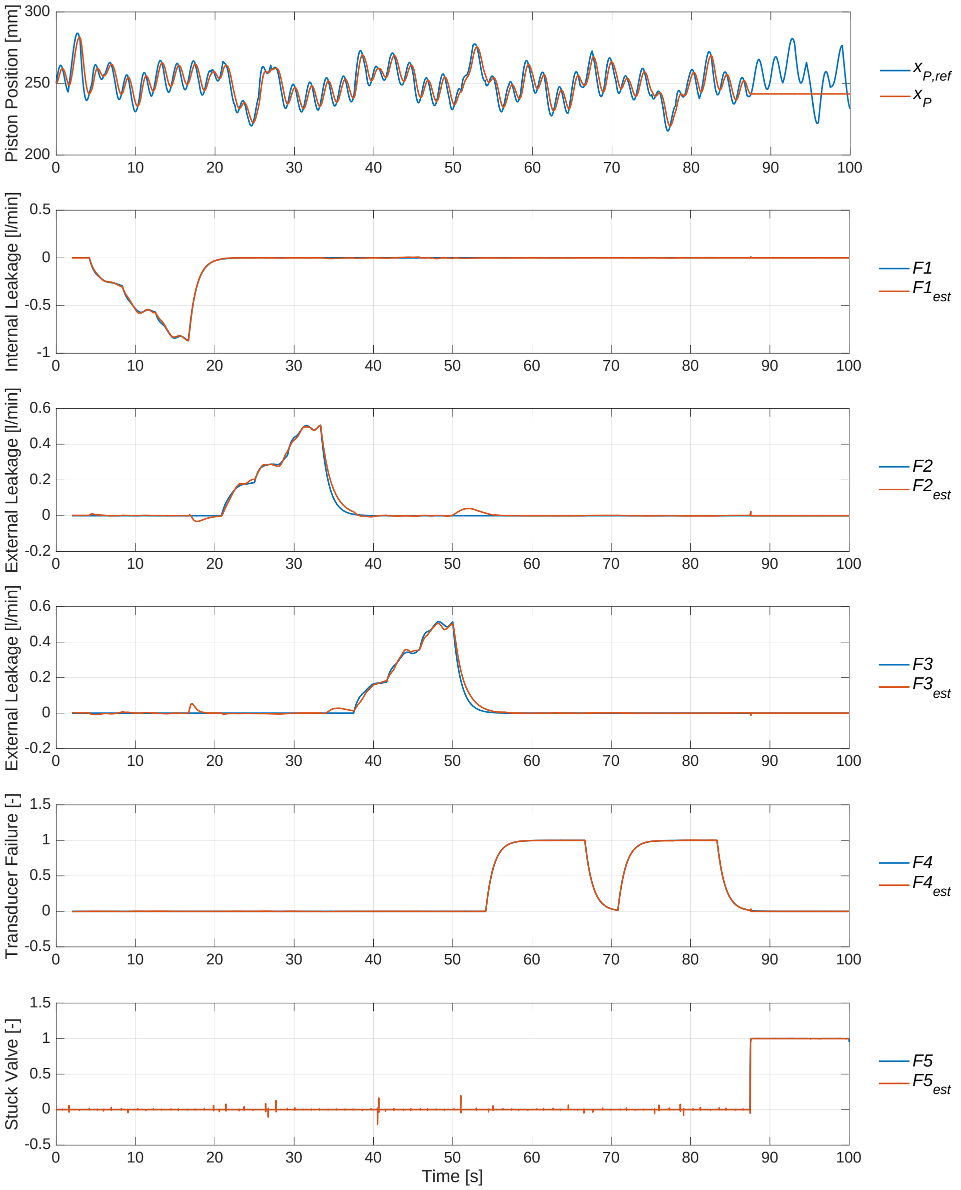


Figure 1: Simuleret fejl status sammenlignet med estimat fra den udviklede FDD algoritme.

Preface & Nomenclature

The content of the Master thesis: *A Novel Approach for Fault Diagnosis of Hydraulic Pitch Systems in Wind Turbines*, is the documentation of the work done by the group MCE4-1023 at Aalborg University, Board of Studies of Energy at the School of Engineering and Science during the spring semester of 2016 with the semester theme: *Master's Thesis in Mechatronic Control Engineering*.
The thesis concerns development and testing of fault detection and diagnosis approaches by utilising artificial neural networks, where the diagnosed system is similar to the hydraulic rotor blade pitching system used in wind turbines.

Acknowledgement

The authors would like to express their gratitude to Henrik C. Pedersen & Per Johansen for guidance and supervision throughout the project. Thanks to the laboratory staff at Aalborg University for preparation of the test rig.

Reader's Guide

Differentiation with respect to time is denoted with a dot, e.g. $\dot{x}$. Matrices are denoted with a bold font, e.g. $\mathbf{x}$, and vectors are denoted with both bold and italic font, e.g. $\boldsymbol{x}$. Estimated values are denoted by a hat, e.g. $\hat{x}$ and normalised parameters by an over-line, e.g. $\overline{x}$. Errors of predictions are given by tilde, e.g. $\tilde{x}$.
Sections, figures, equations, tables etc. are referenced by chapter# .index# , e.g. sec. 5.2, fig. 5.2, (5.2) and tab. 5.2.

The material used for this Master thesis, has been acquired through research of articles, reports, web-pages, books and feedback from the supervisors.
All sources can be found in the bibliography sorted in alphabetical order by main author. References for these sources use Harvard-style notation so all references are marked with company name or last name of the author as well as year of publication.

The software used for modelling and graphical data presentation is MATLAB®R2015a and Simulink®.
Data acquisition and supervisory control are done with National Instruments LabVIEW®.
The documentation report is written in LaTeX.
The attached files include; MATLAB scripts and Simulink models, scripts designed for Data acquisition and supervisory control in LabVIEW, SolidWorks CAD model of the test rig, relevant digital literature, web pages and data-sheets.

Acronyms

ANN	Artificial Neural Network
ARMAX	Auto-Regressive Model Approximation with eXogenous input
BP	Back Propagation
EKF	Extended Kalman Filter
EMA	Exponential Moving Average
FDD	Fault Detection and Diagnosis
FFANN	FeedForward Artificial Neural Network
FIFO	First In First Out
FTDANN	Focused Time Delay Artificial Neural Network
GA	Genetic Algorithm
GM	Gain Margin
GNA	Gauss-Newton's Algorithm
HSS	Hydraulic Servo System
LMA	Levenberg-Marquardt Algorithm
LS	Least Squares
LVDT	Linear Variable Differential Transformer
MLP	Multilayer Perceptron
MFTDANN	Multilayer Focused Time-Delay Artificial Neural Network
NARX	Nonlinear Auto-Regressive eXogenous model
PM	Phase Margin
RMS	Root Mean Square
RT	Real Time
SCADA	Supervisory Control Algorithm and Data Acquisition
SDA	Steepest Decent Algorithm
SSE	Sum of Squared Errors
TDL	Tapped Time Delay
UI	User Interface
WT	Wind Turbine

Subscripts

A	Lumped volume A (piston side)
atm	Atmospheric pressure
avg	Average
B	Lumped volume B (rod side)
bw	Bandwidth
C	Controller
c	Coulomb
CL	Closed Loop
cr	Critical
d	Number of delays
est	Estimate
ext	External
f	Faulty
$Gaus$	A variable with Gaussian noise
i	Integral
$ideal$	Ideally simulated residuals
int	Internal
L	Leakage
l	Load
n	Number of neurons
o	Number of outputs
P	Piston
p	Proportional
ref	Reference
res	Residual
$sine$	Values belonging to test where a sine wave is applied
t	Test
V	Valve
w	Weights

Terminology

Disturbance	"An unknown (and uncontrolled) input acting on a system perturbation: An input acting on a system which results in a temporary departure from a steady state."
Fault	"A fault is an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual, standard condition."
Fault diagnosis	"The task of fault diagnosis consists in determining the type, size and location of the most possible fault, as well as its time of detection."
Fault detection	"The task of generating symptoms from extracted features."
Features	"Data extracted by, e.g. special signal processing, state estimation, identification and parameter estimation or parity relations."
Failure	"A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions".
Malfunction	"A malfunction is an intermittent irregularity in the fulfilment of a system's desired function."
Method	"A procedure, technique, or way of doing something, especially in accordance with a definite plan."
Residual	"Fault indicator, based on deviations between measurements and model-equation-based calculations."
Scheme	"Is a specific arrangement of certain topologies and methods to be followed."
Symptom	"Change of an observable quantity from normal behavior."

Parameters

Symbol	Explanation	Unit
A	Area	[m^2]
α_{th}	Thermal expansion coefficient	[°C^{-1}]
b	Viscous damping	[Ns/m]
β	Bulk modulus	[Pa]
C_D	Discharge coefficient	[-]
c_{ad}	Adiabatic constant of air	[-]
d	Diameter	[m]
ϵ_A	Volumetric ratio of free air in oil	[-]
F	Force	[N]
m	Mass	[kg]
μ	Dynamic viscosity density	[Ns/m^2]
p	Pressure	[Pa]
Q	Flow	[m^3/s]
Re	Reynolds number	[-]
ρ	Density	[kg/m^3]
T	Temperature	[°C]
x	Displacement	[m]
V	Volume	[m^3]
v	Velocity	[m/s]
ω_n	The natural frequency	[rad/s]

Contents

CHAPTER 1

INTRODUCTION

Contents

The main objective of this thesis is to develop a fault detection and diagnosis (FDD) scheme based on artificial intelligence, capable of diagnosing: pressure transducer malfunction, stuck servo valve spool and leakages in a hydraulic system.

Hydraulic systems are commonly used in applications requiring manipulation of heavy objects or more generally systems where large forces are exerted, e.g aircrafts, flight/road simulators, construction vehicles and wind turbines [H. Khan, 2002]. A reliable fault detection and diagnosis scheme to diagnose faults in the given systems is of high priority, since this can be used to prevent downtime of the given system, and help the companies to plan maintenance, hence ensuring maximum profit and safety.

The pitch system on average accounts for more than 20% of the total downtime of a wind turbine (WT) [Wilkinson, 2011], while operation and maintenance costs of a WT are estimated to be 20-25% of the total income [X. Wu, 2012]. Furthermore, a pitch system failure may lead to catastrophic failure of the WT as described in [X. Wu, 2012]. The afore mentioned issues may be reduced by applying an automated diagnosis scheme, capable of detecting faults on an early stage. Hence, there exists an incentive to investigate how a fluid power system such as the pitch system can be monitored to detect and diagnose faults.

1.1 Introduction to the Wind Turbine Pitch System

In a WT it is desired to generate a steady speed on the shaft by constantly adjusting the pitch angle of each rotor blade, to account for changing wind conditions, within the rated wind speed for the given WT. The blades on the WT connects to a hub, which is mounted to a rotating shaft connected to a generator through a gearbox. The pitch angle of each blade is adjusted through either hydraulic or electrical actuation, which is mounted within the rotating hub of the WT. In fig. 1.1 a WT interior design is depicted, and the hydraulic components in the hub of the WT are enlarged.

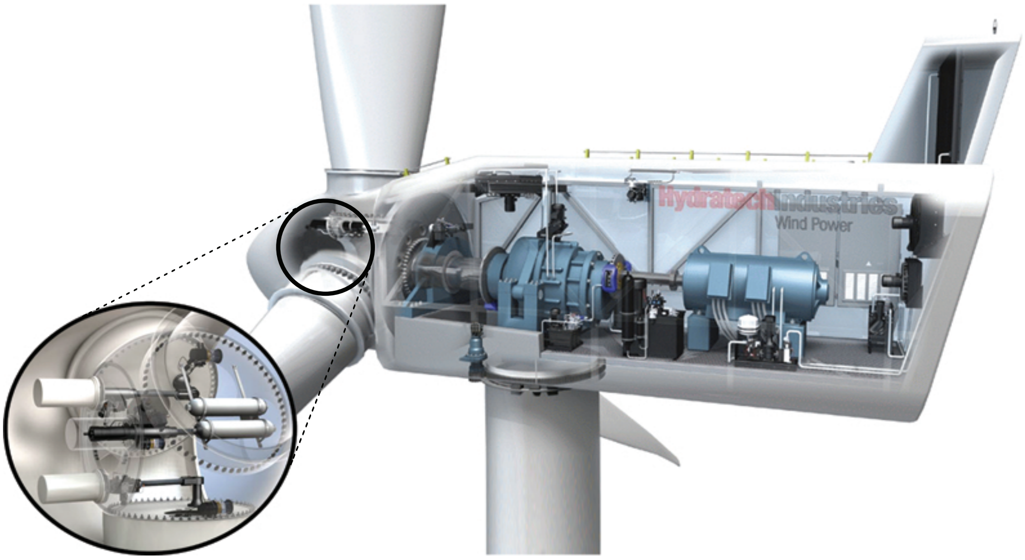


Figure 1.1: WT interior design, with enlarge of pitch system [Industries, 2016].

From fig. 1.1 it is observed that the pitch system is only one among several subsystems representing the WT. However, the pitch system is part of the safety system in a WT, causing strict requirements to the functionality and safety.

In order to design any condition-monitoring and diagnosis scheme it is necessary to analyse the different faults which can be expected in such a system. Therefore the physical structure of the pitch system will be analysed in sec. 2.1.

1.2 Introduction to Maintenance and Diagnosis

Maintenance is in general divided into three categories: Breakdown Maintenance, Preventive Maintenance and Condition-based Maintenance [Watton, 2007].

Breakdown Maintenance

Breakdown maintenance is also known as the "fix it when it breaks" type of maintenance. Breakdown maintenance is also the most costly and least efficient method, because many of the repairs could potentially be prevented with properly Preventive maintenance. The approach to let the system run until catastrophic failure could potentially result in very expensive removal and replacement.

Preventive Maintenance

This approach is conducted on a time frame recommended by the manufacturer. Preventive maintenance includes all kinds of maintenance within a WT e.g; change of the oil and filters, de-icing of the turbine blades, checking for electrical problems and replacement of hydraulic components. However, the downside of this method is that functioning parts might be replaced before the end of their end of life. This is a potentially hidden expense for the paying company.

Condition-based Maintenance

Condition-based maintenance is based on acquisition and analysis of data. Condition-based maintenance is a proactive approach against known system issues, where an automated surveillance system may be used to estimate the different faults before they reach a critical state. This will give the company time to plan the maintenance, hence avoiding the Breakdown maintenance, and the risk of replacing a functioning component.

Condition-based maintenance requires knowledge about the system's health, which may be determined by an appropriate diagnostics method. This diagnostics method relies on sensors to measure specific system states.

The measurements can in general either be direct or indirect. With the direct measurements it is possible to detect the fault directly assuming the sensor is operating as intended. However, this method is often expensive or inconvenient which is why an indirect method is a more common approach. However, this method can not be used to detect a given fault directly, hence interpretation of the measurement data is required. Furthermore, multiple faults can be included in the same measurement further complicating the detection and diagnosis.

1.3 Initial Problem Statement

Based on the operation and maintenance costs, and safety issues regarding the pitch system presented in the introduction, the following initial problem statement is formulated:

"How can an automated fault detection and diagnosis scheme be designed for the pitch system of a wind turbine?"

This statement serves as the baseline to a problem analysis and a literature review, which are presented in the two following chapters.

Part I

Problem Analysis & Problem Statement

CHAPTER 2

PROBLEM ANALYSIS

Contents

The objective of the problem analysis is to analyse the initial problem statement of this thesis and to gain basic knowledge about the system and diagnostics theory. In the following section the pitch system is analysed with a focus on the failure rate of the specific components.

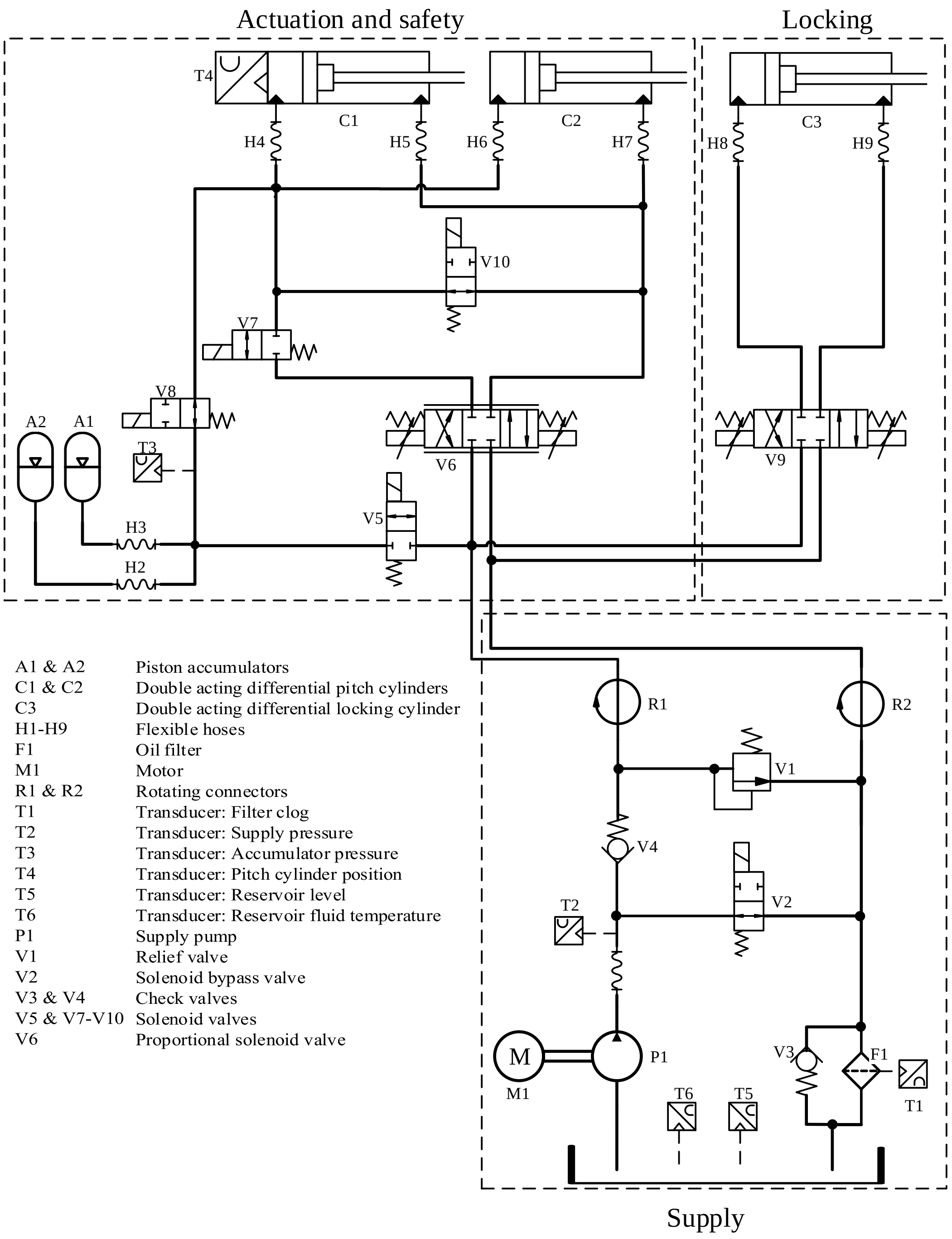
2.1 Analysis of the Pitch System

The analysis of the pitch system presented in this section is based on [J. Liniger, 2016]. In general the pitch system can be divided into three parts: Actuation/Safety, Locking and Supply. The topology of these is illustrated with a hydraulic diagram in fig. 2.1.

The supply circuit is stationed in the nacelle and is connected to the actuator and safety circuits through the main shaft. Both the actuator and safety circuits are rotating within the hub, and three of each system are present (one for each blade). The actuator circuit is made up of one- or two-cylinders which both are controlled by a proportional valve. The safety circuit is made up of two accumulators which is used for storing pressurised fluid, that in an emergency are able to actuate the pitching cylinders to full stroke, using the blades as an aerodynamic brake. Lastly the locking circuit is used to mechanically lock the blades.
As explained in [J. Liniger, 2016], the WT has four different operation modes; Startup, Power Regulation, Emergency Shutdown and Normal Shutdown. Each of these modes are described in the following list.

Startup

The startup mode is activated when the WT is given the command to start power production. Before the startup command is given the two pitch cylinders C1, C2 and the locking cylinder C3 are all fully extended, and all valves are deenergised. First step in the



A1 & A2	Piston accumulators
C1 & C2	Double acting differential pitch cylinders
C3	Double acting differential locking cylinder
H1-H9	Flexible hoses
F1	Oil filter
M1	Motor
R1 & R2	Rotating connectors
T1	Transducer: Filter clog
T2	Transducer: Supply pressure
T3	Transducer: Accumulator pressure
T4	Transducer: Pitch cylinder position
T5	Transducer: Reservoir level
T6	Transducer: Reservoir fluid temperature
P1	Supply pump
V1	Relief valve
V2	Solenoid bypass valve
V3 & V4	Check valves
V5 & V7-V10	Solenoid valves
V6	Proportional solenoid valve

Figure 2.1: Generalised fluid power pitch system, based on [J. Liniger, 2016]

startup mode, is a sanity check on all transducers and sensors (T1-T7). The supply pump is started, and bypass valve V2 is energised until T2 confirms supply pressure has been reached. The accumulators A1 and A2 are loaded with pressurised fluid, which is done by energising valve V5 until T3 confirms a pre-set pressure. Lastly locking cylinder C3 is retracted by using V9, and the blade is released.

Power Regulation

The power regulation mode is active when the WT is producing power. The pitch cylinders C1 and C2 are used to actuate the blade pitch rotation. Both cylinders are driven by the proportional valve V6. The pitch controller receives a pre-calculated pitch angle set-point, depending on the measured wind speed. The controller sends a signal to V6 and adjusts the pitch angle on the blade until the transition is complete. When the power regulation mode is active, valve V5 is deenergised, and valve V7, V8 and V10 are all energised. Furthermore, it should be noted that when the cylinders are extended, it means that the blade is pitching out of the wind and breaking.

Emergency Shutdown

The emergency shutdown mode is activated by deenergising all valves. This operation opens for the pressurised fluid from the accumulators to both pitching cylinders (C1 and C2). This forces both cylinders to a fully extracted position where the rod side fluid is lead to the piston side i.e. the cylinders are driven in regenerative configuration. Furthermore, from the circuit in fig. 2.1 it should be noted, that the safety circuit is separated from the actuation circuit by valve V5 and V7.

Normal Shutdown

The normal shutdown mode is activated if the wind speed is to low or high for power production, or if a maintenance procedure requires the WT to stop. To initiate the shutdown mode, the pitch angle setpoint is set to 90° which fully extends the pitching cylinders C1 and C2. When the extension is completed the locking cylinder C3 is engaged by the use of valve V9. Lastly when the locking procedure is completed, the supply circuit is deenergised.

It is described in [J. Liniger, 2016], that a special case applies for the emergency shutdown mode. Where this mode is "*safety critical and is normally designed with 2oo3 (Two Out Of Three) redundancy*". Which means it is sufficient for 2oo3 blades to turn fully 90° to perform an emergency stop. However, in the event of only 2oo3 safety systems are deployed, large loads could be generated on the WT's structure depending on the severity of the unbalanced aerodynamic forces.

2.1.1 Failure Analysis of Relevant System Components

In order to determine failures of interest, an analysis presented in [J. Liniger, 2016] has been used. The analysis covers the main components of the pitch system and the probabilities of different failures. The main components with their field failure rate are presented in tab. 2.1.

From tab. 2.1 it can be observed, that the oil is the main contributor to the combined component based field failure rate. However, since oil faults are difficult to implement in a

Component	Total field failure rate [%]
Oil	36.6
Valves	19.7
Accumulators	15.4
Pump	8.7
Pipe/Hoses	6.0
Hydraulic Block	6.0
Cylinders	2.3
Transducers	2.1
Locking system	1.8

Table 2.1: Component based field failure rate, an expanded table of the failure mechanisms can be found in app. B

controlled manner on a physical set-up, it is therefore not considered further in this thesis.
The total field failure rate of the valves accommodate for ≈ 20% where the top five contributions can be divided into the percentages presented in tab. 2.2.

Valve	Internally stuck [%]	Electrical faults [%]	External leakage [%]	Internal leakage [%]
V6	0.0	0.0	0.9	5.0
V7	6.2	1.2	1.9	1.9
V8	5.6	1.2	2.2	5.0
V9	6.2	2.5	1.2	1.2
V10	5.6	1.2	1.9	0.0

Table 2.2: Top five different valve failure mode distribution.

Though several faults are of significant occurrence, it has been decided to restrict this thesis to consider internal & external leakages in V6 and/or in the actuator. These are chosen since internal & external leakages often are associated with wear in valves and/or cylinders, and therefore this can indicate a possible fault on an early stage.
Furthermore, a stuck valve spool in V6 along with a malfunction of a transducer will be investigated. The latter two have been chosen to test whether the Fault Detection and Diagnosis (FDD) scheme is versatile, i.e. that it can function as intended with faults of different characteristics.

The volumes of which leakages can occur are considered lumped as shown in fig. 2.2.

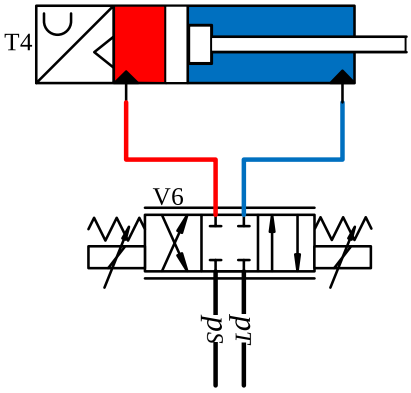


Figure 2.2: Simplified lumped hydraulic circuit in which leakages can occur. Red: V_A & blue: V_B.

The origin of the external leakage is not necessarily obvious, since it can occur in almost all valves, connections, hoses, cylinders, accumulators etc.
Therefore the identification of external leakage is simplified to: leakage from either lumped volume A or B (V_A & V_B) shown in fig. 2.2. Similar, the internal leakage can occur in either servo valve or cylinder.

Based on the above analysis, the faults **(F.1)**-**(F.5)** have been selected.

(F.1) Internal rupture/leakage between the lumped volumes.
(F.2) External rupture/leakage from V_A.
(F.3) External rupture/leakage from V_B.
(F.4) Transducer failure.
(F.5) Stuck spool in servo valve.

Diagnostics theory will be investigated in the following in order to analyse how the above faults may be detected and diagnosed.

2.2 Analysis of Diagnostics Theory

In general diagnostics is referred to, as the discipline or practice of diagnosis, where diagnosis is the recognition of the nature and cause of a certain phenomenon. In [Gertler, 1998] it is suggested that diagnostics may be classified into two main groups; model-free- and model-based FDD.

2.2.1 Model-free Fault Detection and Diagnosis

The model-free FDD methods do not need an analytical model, however it may utilise physical redundancy, limit checking, spectrum analysis, special sensors and logical reasoning. The methods are listed below from [Gertler, 1998].

(a.1) **Physical redundancy**: A method where multiple sensors are installed for the same measurement. Any major variation between the two measurements will indicate a sensor fault. However, in systems with two parallel sensors, fault isolation is not possible, since the fault could originate from either sensors. Hence, at least three parallel sensors are needed, to isolate the fault. The method of physical redundancy utilises extra hardware hence more weight and higher expenses, which could be a concern in some applications.

(a.2) **Limit and trend checking**: A widely used approach in practice, the plant measurements are compared with pre-set limits. If a measurement exceeds a threshold, a fault situation is indicated. In systems there exists two levels of these limits. First a pre-warning, then an emergency reaction. The limit checking can be expanded to surveillance of the time-trend of different variables.

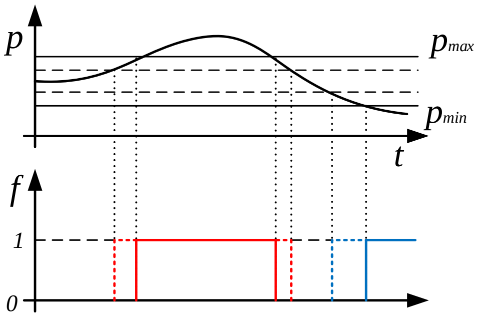


Figure 2.3: Trend checking example, where a upper and a lower threshold fault is indicated with red and blue respectively.

(a.3) **Special sensors**: The special sensors can be installed to measure specific fault-indicating quantities, E.g temperature, sound, deformation etc. The extra sensor data is applied for trend checking, as explained in **(a.2)**.

(a.4) **Spectrum analysis**: Spectrum analysis of system measurements may also be utilised for FDD. Most of the measurable system states exhibits a given frequency spectrum under normal system conditions. Any deviations from this base frequency is an indication of abnormalities in the system. The isolation part can then be utilised, since certain type of faults may have their own characteristic signature in the frequency spectrum.

(a.5) **Logic reasoning**: Is a broad range methods, which may be complimentary to the methods described above. In essence the method is aimed at evaluating the symptoms obtained by the detection system hardware and software. One of the most common forms of this method consists of the logical rule set; *IF symptom AND symptom THEN conclusion*[Gertler, 1998]. From this, each conclusion can serve as a new symptom until a final conclusion is made [L. Ning, 2011].

2.2.2 Model-based Fault Detection and Diagnosis

The model-based Fault Detection and Diagnosis method utilises an explicit mathematical model, of the monitored physical system. The general and conceptual structure of model-based FDD includes two main processes. First the Model-based Fault detection which relies on analytical redundancy, where measurements of physical system states are compared to analytically calculated system states, which are formally referred to as residuals.

Secondly, the model-based fault diagnosis is based on an evaluation of the generated residuals, where a decision making algorithm may be executed to inform if a fault has appeared.

Fault Detection Methods

The Fault detection process is illustrated in fig. 2.4.

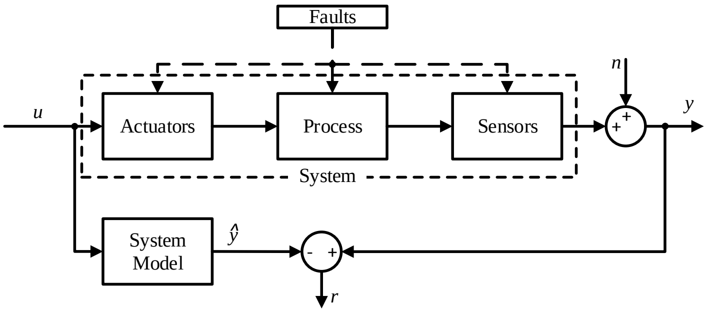


Figure 2.4: General model-based fault detection scheme, with inspiration from [Isermann, 2006].

Fig. 2.4 shows that faults can occur in the actuators, process and sensors. The input applied to the system is directed to a mathematical model which estimates the system state variables. The observed features are compared with their nominal values to declare if one or more faults are present, and hence abnormalities may be detected. These abnormalities described by the residuals serves as the basis for the fault diagnosis process.

The model based approach consists of one of the somewhat overlapping methods **(b.1)**-**(b.4)** described in [Isermann, 2006].

(b.1) **Parameter Estimation**: This is based on an identification of the system parameters in a fault-free situation, by repeatedly identifying these parameters on-line deviations can be detected.

(b.2) **Neural Networks**: The neural networks method describes the relationship between any number of inputs and outputs by utilising; gains, summations and nonlinear, linear or discontinuous activation functions. The nonlinear mapping capabilities of a neural network is defined in [Isermann, 2006] to fit any desired degree of accuracy.

(b.3) **State Observer**: An example of a state observer is the Kalman filter. The Kalman filter is an optimal observer in the sense that it minimises the mean square error of the estimated parameters if all noise is Gaussian. However, fault detection with Kalman filter requires a bank of "matched filters", one for each suspected fault. Furthermore, it must be checked which of these estimates can be matched with the actual observation.

(b.4) **Parity Equations**: These are rearranged direct input-output model equations, with fixed parameters, which are subjugated to a linear dynamic transformation. The linear parity model runs in parallel to the process, as shown in fig. 2.4.

Fault Diagnosis Methods

The previous mentioned detection features are used for the fault diagnosis method, where it is desired to determine the fault type with as many details as possible. These details could consist of the following; fault size, location and time of detection.

The diagnosis is affected by the precision of the model and the noise on the measurements. Therefore, different diagnosis algorithms need to be considered in order to determine, if a given deviation of a system feature is a fault or caused by uncertainty and noise.

A survey of the model-based diagnosis methods are presented in [Isermann, 2006], these are shown in fig. 2.5.

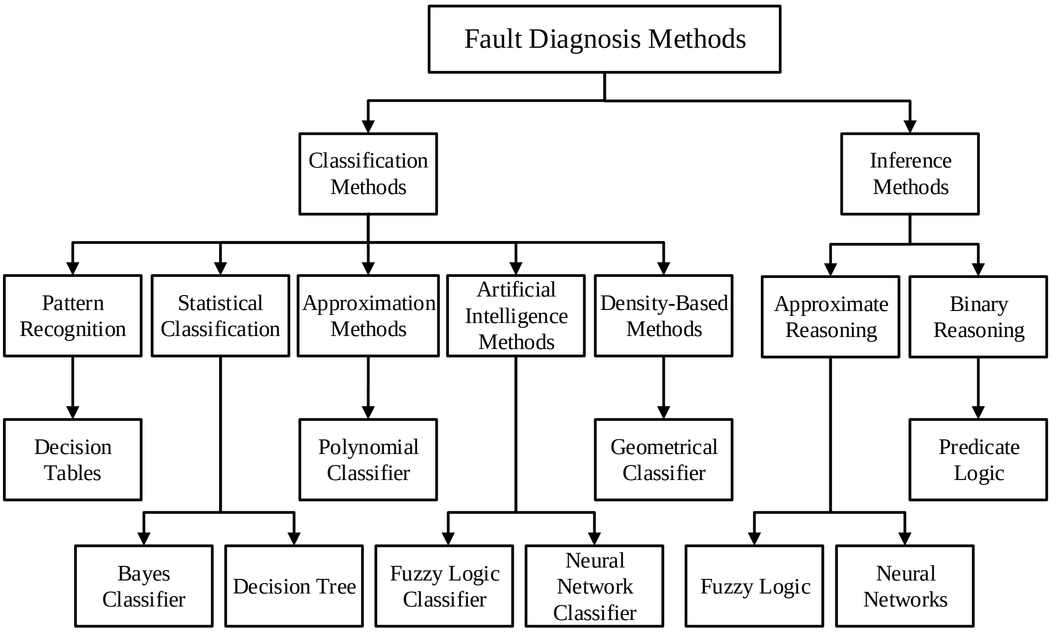


Figure 2.5: Survey of the different fault diagnosis methods, with inspiration from [Isermann, 2006].

In fig. 2.5 the diagnosis methods are divided into two main groups. The difference between classification- & inference methods, is that classification can be done without structural knowledge, of how the symptoms and fault affects the systems behaviour, whereas the inference methods depend on this structural knowledge. The general task of all the diagnosis methods is to separate a number of different faults by using a number of symptoms.

A brief analysis of diagnostics theory has been presented, where it has been illustrated that various FDD methods exists. These are desired narrowed down based on state of the art research on the subject. Therefore, a review of the literature regarding applying FDD on a system with some relevance to the pitch system is performed in the following chapter.

CHAPTER 3

Literature Review

Contents

The following chapter is a literature review of different methods for fault detection and diagnosis (FDD) of hydraulic servo systems. It is of special interest to locate the latest research on the matter of actuator leakage, stuck servo valve and malfunctions of sensors.

Sec. 3.1 is reviewing general approaches used in the industry for FDD of different systems. Throughout the review, two methods yielded promising results, based on artificial neural networks (ANN) and Kalman filters, and sec. 3.2 consists of a review of the latest advances with these two specific methods applied on a fluid power system.

Finally, the findings will be used for a conclusion, which will serve as the foundation for the problem statement of this thesis.

3.1 General Fault Detection and Diagnosis Literature

Recent application of model-based FDD has been reviewed, where the addressed articles are initially sorted with respect to the type of system. The purpose of the review is to outline the applied methods and the corresponding potential when applied in different systems. Mainly FDD applied on fluid power systems will be of interest, but also fundamental work within FDD will be investigated.

3.1.1 Nonspecific Stochastic Systems

The articles highlighted in this section give an idea of the possibilities for FDD when applied in a given stochastic system. One of the earliest articles in this field is [Beard, 1973], and the main part of the literature found regarding stochastic system FDD can be traced back to this work.

The work done by [R. J. Patton, 1996] concerns an optimal observer for residual generation and robust FDD for stochastic systems with unknown disturbances. This is claimed by the authors to be the first time such consideration has been addressed and solved for a fault diagnosis design. The output estimation error with disturbance has been used as a residual signal and a statistical testing procedure was applied to test the residuals and from this used to identify sensor and actuator faults.. The method was only tested in simulation. However, the results showed that the method could produce a good estimate of the states and that the faults could be diagnosed by threshold checking.

An optimal stochastic fault detection filter was proposed by [R. H. Chen, 1999]. The objective of the filter is to monitor certain faults, in the article mentioned as: "*Target faults and block other faults*". An advantage for this method, is that the detection filter design can be obtained for both linear time-invariant and time-varying systems.

A combination of a particle filtering algorithm with a log-likelihood ratio test was proposed by [P. Li, 2001] based on the generalised likelihood ratio method proposed for the first time by [A. S. Willsky, 1974], where the Kalman filter is replaced with a particle filter. The method is an FDD method for general nonlinear stochastic systems with non-Gaussian noise and the results from this method compared with results by using the Extended Kalman Filter (EKF) showed superior performance, and promising tendencies. However, only theoretical simulations were carried out.

In [L. Guo, 2004] a B-spline expansion, and a nonlinear weight model has been utilised to detect and diagnose problems in general stochastic systems. All tests have been done with the measured information being the probability distribution of system output rather than the value of the system output.

This review reveals that the research on general FDD of stochastic systems with Gaussian or non-Gaussian noise are model-based. The preferred state observer is some variation of the Kalman filter.

3.1.2 General Fluid Power Systems

Articles with relevant FDD strategies applied on fluid power systems are investigated in the following. This serves the purpose of analysing the different methods used within general fluid power. The arguments presented by the authors will serve as a foundation for the choice of method.

A novel adaptive nonlinear observer technique is presented in [H. Wang, 1996]. The approach is to have a fixed detection observer and an adaptive diagnosis observer separately. A modification to the diagnosis observer is proposed to enhance its robustness. Thresholds for the residual signals have been determined and theoretical verification of the technique is presented.

In [G. J. Preston, 1992] a robust method for fault detection is proposed. This utilises a combination of two observers based on nonlinear models of the hydraulic test rig. The authors conclude that the residual is capable of tracking faults in the range of 1% deviation and that it is robust towards unknown disturbances.

These articles constitute some of the early work in robust fault diagnosis on hydraulic systems, and reveals that a residual based approach have potential, due to robustness against noise and disturbance.

The work done in [R. Song, 2002] investigates a linear system identification method, named Auto-Regressive Model Approximation with eXogenous input (ARMAX) with the addition of the least squares (LS). This is used as FDD to detect and isolate faults caused by wear of the supply pump (±500 psi). The method was found not feasible due to the hard nonlinearities in the hydraulic system. However, it was shown that by utilising direct threshold checking on the estimated model the method could be used to detect and isolate faults originating from incorrect supply pressure.
Moreover, the authors state that ANN is deselected due to training time on the ANN. This argument could prove outdated due to the development in computational equipment.

A nonlinear observer-based scheme has been suggested by [H. Khan, 2005], combined with Wald's sequential test for detecting faults in a fluid power system. The fault detection is done by measuring the control signal and the actuators velocity, which are the measurable input and output variables, respectively. System and sensor faults have been simulated for the verification. The validity of the nonlinear observer scheme was confirmed by simulations and experiments, for faults due to incorrect supply pressure and sensor fault.
The authors proved that the observer was robust when driven by input signal and measured velocity.

In [K. Cao, 2007] FDD on an electro-hydraulic position servo is performed by utilising a support vector regression as model for residual generation. The fault considered is regarding the displacement sensor where two examples are given, and the fault is theoretically detected and isolated. Furthermore, the authors state that support vector regression is chosen instead of ANN due to traditional ANN approaches have difficulties with generalisation, producing models with overfitting. This issue will be revisited in sec. 8.2.

The above articles utilises parity equations and state observer methods to detect certain fluid power system faults.

3.1.3 Aviation Fluid Power Systems

Literature about FDD applied for fluid power systems used in the aviation industry has been reviewed, because of its dependency regarding system reliability, hence a high quantity of literature is available from this industry.
The purpose is to investigate which methods have been applied practically.

The paper [V. A. Skormin, 1992] presents a mathematical model of a self-contained flight actuator which implies, that all control and command computations are preformed within the actuator module. On-line parameter estimation is done to detect the following failures: leakage in the hydraulic system, loss of the control surface, excessive friction in hydraulic cylinder, mechanical damage of the actuator, loss of magnetism of the motor, and air in the hydraulic system. It is then defined which parameters of the actuator model is perturbed by the failure, and these parameters are then estimated. The conclusion is that it is possible to detect parameter changes through simulation and it is proposed to perform a prediction of failures when the elements exhibit statistically significant degradation at a "future time". Similar approach was followed for a variable displacement pump of a flight actuation system in [V. A. Skormin, 1995]. The main failure modes considered are fluid contamination by either air or water and excessive leakage. The method was only proven by simulation.

The method described by [C. S. Byington, 2004] utilises a multi-layer ANN combined with fuzzy logic classification where a Kalman filter state predictor is used to predict future progression. The system considered is a stabilator actuator with electro-hydraulic servo valves in an aircraft, used for operating flight control surfaces, landing gear, cargo doors and weapon systems. The article emphasises that ANN is effective as time-series predictors and have capabilities in learning correlations of several inputs/outputs. The ANN was applied as observer and fuzzy logic classifiers successfully diagnosed the system. Furthermore, the authors compare performance of a spectrum analysis of both pressure and servo valve control signal with the ANN predictor with the latter being superior.

In [T. Kobayashi, 2006] a hybrid Kalman filter based on a nonlinear on-board engine model and piecewise linear models is applied to a aircraft gas turbine engine. The diagnosis is performed by threshold checking and the results for this scheme are promising, since the fault detection system was able to detect biases in sensor and actuator, however it failed to detect component faults. This shortcoming, accordion to the authors, is mainly due to limitations within the available sensors, from which faults are observed. The authors propose this method for in-flight diagnostics of aircraft gas turbine engines.

The paper [T. Kobayashi, 2008] utilises dual-channel sensor measurements with a linear on-board engine model to ensure sensor redundancy. Faults of sensors and components in the aircraft engine is simulated. This is diagnosed via a threshold check, designed with respect to the statistical standard deviation of the sensor measurements. The linear on-board engine model is used to generate residuals, which are used for detection of an anomaly which is diagnosed when exceeding the afore mentioned pre-established thresholds. It is stated that due to degradation of the engine over its lifetime, the baseline system must be updated periodically constituting a major challenge.

In [W. Vianna, 2014] the objective was to estimate hydraulic leakage within a hydraulic system on-board an aircraft using only the aircraft's standard sensors such as fluid temperature, fluid pressure and reservoir level. Results showed the capability to detect leakages, although some estimations were less precise. The scheme utilises parity equations, used on several aircraft data series to validate the performance of the method.
The results showed some of the estimations were less precise in estimating leakages. However, the method is still concluded promising by the authors.

In general there have been applied various FDD methods for hydraulic systems in the aviation industry. Most of which are based on an observer and some pre-set thresholds depending on the fault. Based on the results from [C. S. Byington, 2004] the methodology of ANN as predictor combined with a fuzzy-logic diagnosis method is interesting. This is manly due to the ANN predictors superior performance.

3.1.4 Pitch System

A review of the relevant articles regarding the pitch system is used to present the current state of the art approaches for the system of interest in this thesis.

In [X. Wu, 2012] a dynamic model of the pitch system and an adaptive parameter estimation algorithm were developed to identify internal and external leakage coefficients under a time-varying load on the pitching system. Furthermore, a change in bulk modulus e.g. due to air contamination or change in fluid temperature could be estimated. A downscaled version of the set-up was developed. With use of the experimental data, the leakage and leakage coefficients were predicted via the proposed method (external leakage of 7.8% mean steady state error and 11% peak steady state error. Internal leakage of 7.3% mean steady state error and 13.3% peak steady state error).

In [B. Chen, 2013] a FDD scheme is proposed by using a knowledge-based adaptive neuro-fuzzy inference system. The main objective of the research is to achieve an automated detection of significant pitch faults within the wind turbine. The advantages using a-priori knowledge incorporation are stated by the authors as: "*the proposed system has improved ability to interpret the previously unseen conditions and thus fault diagnoses are improved*". However, in order to apply the proposed scheme, the paper describes that data of 6 known wind turbine pitch faults are needed to train the ANN with the knowledge incorporated. The origin of the faults is not addressed.

The paper [Y. Vidal, 2015] presents a FDD and fault tolerant control for a pitch system. The method is tested theoretically via aeroelastic wind turbine simulator software. The authors investigates the fault modes: high air content in the oil (from 7% to 15%), pump wear (75% of the nominal pressure) and hydraulic leakage (corresponding to 50% of the nominal pressure). These faults were successfully diagnosed by evaluating the natural frequency and damping of the system and comparing it with the nominal values. However, the severities of the investigated faults are relatively high.

In [E. Sales-Setien, 2015] a fault detection strategy applicable to the pitch system in offshore wind turbines is presented. The method consists of a PI observer and by attenuating the $\mathscr{H}_2$-norm between both measurement noise and fault signal, and the fault estimation error, thresholds were defined to diagnose the faults. The observer is designed with a trade-off between the number of allowed false alarms, minimum detectable faults and the detection time.
The authors of the article suggests, that an extension of the scheme towards other fluid power systems could be a potential future work.

In the wind power industry, FDD with application of ANN for the pitch system is not widely used. However, application of various observer-based schemes have been found and it is emphasised that the observer must be robust against changing wind loads and that the diagnosis should be able to reject measurement noise.

The study of the literature has revealed that the state of the art FDD is done with model-based methods, where the schemes are designed differently depending on application and considered faults. There does not exist any consensus about which residual generation method or diagnosis method is optimal. However, in the literature about leakage diagnosis in a hydraulic servo system, the extended kalman filter and various methods based on ANN have proved to be efficient. These have not been applied directly on the pitch system, but the hydraulic systems considered are very similar to the one of interest in this thesis. These methods and the literature concerning them are presented in the following section.

3.2 Literature Reviewed with a Focus on Applied Fault Detection and Diagnosis Method

The relevant literature about Kalman filter- and ANN-based FDD is reviewed in the following. The main reason for choosing these is due to theirs advance in the most recent and prominent research, furthermore, ANN and machine learning have gained increasing interest due to the availability of big data and increasing computational power.

3.2.1 Kalman Filter Based FDD

Different methods for tackling this issue have been investigated throughout the years, one of which utilises the Extended Kalman Filter (EKF). The articles; [L. An, 2003], [L. An, 2004] , [L. An, 2014a] and [L. An, 2014b] investigates the method of the EKF, for diagnosis of circuit failure and actuator leakages. The scheme designed by [L. An, 2003] proved to detect abnormalities of 10% in the supply pressure. However, the authors stated that because of the uncertainty of the system and noise, the level of the residual under normal condition is much higher than those in simulation, which reduces the resolution of the developed FDD scheme.

The study in [L. An, 2006] confirms on previous published papers focusing on hydraulic leakage fault detection and isolation. The article verified the effectiveness of the suggested FDD scheme based on the EKF for detecting a change of supply pressure presented in [L. An, 2003]. The FDD scheme is tested to be effective in experiments, and it is demonstrated that with a proper residual generation strategy, various leakage faults at a hydraulic actuation system can be detected with a precision of 0.6 l/min to 1.3 l/min. Furthermore, the study found that the increase of residual error is proportional to the increase of leakage, which led to the conclusion that the quantity of the leakage could be estimated from the residuals.

In [L. An, 2004] the external and internal leakages of the hydraulic servo system were investigated, and it was experimentally documented that increasing leakage resulted in more significant residuals. The work was continued in [L. An, 2014a] documenting diagnosis of experimental hydraulic actuator leakage faults using EKF. Here actuator displacement and line pressures were available measurements and the internal leakage detected and estimated correctly was in the range of ≈ 0.825 l/min.
In [L. An, 2014b] internal and external leakages of 0.25 l/min were detected correctly, which was an improvement from previous results proven by the same authors. According to the authors the work presented in this article contributes to the field and lays a foundation for developing on-line health monitoring of hydraulic actuators.

Further use of the standard Kalman filter, has been applied by [T. Kobayashi, 2003]. The method uses multiple Kalman filters, each of which is designed for detecting a specific sensor or actuator fault. The method showed, that it is possible to detect and isolate sensor and actuator bias errors. However, a drawback for the method is that a very large amount of filters have to be designed for a given system.

In the above, the detectable leakage flows with state of the art FDD schemes have been quantified and the information from the articles about residual behaviour can be used for development of FDD later in this report.
In general the results are from a situation by applying a sine position reference or pseudo-random step inputs. Furthermore, the leakages are diagnosed by applying the moving averaged values of residuals. This is done to filter the signals and this method will be applied when interpreting the results in this thesis.

3.2.2 Artificial Neural Network Based FDD

Fault diagnosing of internal and external leakages in hydraulic fluid power systems consists of few attempts with utilisation of ANN for diagnostics. Furthermore, the existing research is performed with poor computational equipment compared with the processing power of today's equipment, [T. T. Le, 1997], [S. He, 2000], [L. Hongmei, 2006a] and [Z. Yinshuo, 2013].

In the research conducted by [T. T. Le, 1997] multilayer perceptron type ANNs are used to identify and isolate leakages, where both single-leakage and multiple-leakages were investigated for a hydraulic servo system. It was proven that an external line leakage of 0.2 l/min was detectable. However, detection of internal leakage across the cylinder seal did only yield promising results for flows above 1.0 l/min. These results are based on line flows and pressures directed trough a multilayer perceptron with 50 hidden neurons and with 31 different outputs. The network training period was ≈ 4 h computed on a Pentium 90 MHz PC operating a UNIX workstation. Lastly it was found, that the ANNs performance was degrading if the working oil temperature deviated from the temperature used for training the network. Furthermore, a network for both piston extending and retracting were trained with situations of constant valve signals. The authors claim that by adding the absolute value of the pressure drop in the actuator and the flow ratio, the convergence of the ANN is accelerated.

In [W. J. Crowther, 1998] an ANN is applied for FDD on a hydraulic actuator circuit and it is directly stated that: "*A novel aspect of the work is that a physical understanding of the system to be diagnosed is used to determine the inputs required for the neural network. This is in contrast to approaches where all available data are presented to the network in the expectation that the relevant information can somehow be extracted.*" Furthermore, the authors claims that: "*The case for using neural networks to solve difficult, so-called 'intelligent' problems has perhaps been overstated in the last few years. However, for the present problem, neural networks offer a number of advantages: in particular, their ability to deal with highly nonlinear dynamic systems and high speed of operation when implemented in hardware.*"
The faults considered in the research concern incorrect supply pressure, increased drive actuator cross-line leakage and increased load dynamic friction. These faults were introduced one at a time, since it is assumed statistically unrealistic for several faults to occur at the same time. The tests were conducted with a maximum peak-to-peak input signal corresponding to 10% of the actuator stroke with steady state for less than 10% of the test period (giving a

switching time of 20 ms). The multilayer perceptron network was trained via back-propagation with 50 samples of; control currents, pressure drops and piston positions forming each input vector giving 150 inputs, and the ANN was designed with 10 hidden neurons. The network output consisted of a parameter estimation of the three parameters relevant for the mentioned faults. During the training of the ANN it was found that increased friction was more difficult to detect than the other faults.
The authors concluded that a network had to be trained for each specific fault, meaning a three-fault network was necessary. Moreover, it was emphasised that: "*A major challenge is to obtain training data that cover the whole system output vector space for the faults of interest. This problem is directly related to that encountered in traditional system identification or parameter estimation schemes.*"

Although no fault diagnosis is performed in [S. He, 2000], it has been reviewed due to its investigation of a nonlinear system identification method for estimating the model of a hydraulic force-acting system utilising ANN. The method used for approximating the system model is the ANN modelling method called Nonlinear AutoRegressive-Moving-Average (NARMA). The article compares the results from the NARMA model, with a linear counter method called AutoRegressive-Moving-Average (ARMA).
The ANN was trained with an amplitude modulated pseudo-random binary sequence and tested with a sinusoidal signal not present in the training data.
The comparison between the ANN method and the linear method, showed that the ANN could produce RMS errors six times smaller than the linear model.
This proves the strong identification capabilities of ANN applied on nonlinear system.

In [Ehsan Sobhani-Tehrani, 2009] it is stated that when utilising ANN for FDD, problems might arise when measurement noise or other disturbances are injected in the system. The authors emphasise that this issue can be addressed by using an ANN-based residual generation decision-making scheme, initially introduced by [R. J. Patton, 1994]. Furthermore, it is stated that a static NARX or static recurrent network is superior at predicting nonlinear system behaviours even though environmental conditions are changing and/or noise is present in the system. No literature about application of this on fluid power has been found, but the topology with combination of ANNs used in observer-based FDD in fluid power has been documented in the two following articles.

In [L. Hongmei, 2006a] an investigation of the use of a two stage improved Elman ANN is done. The first stage was used as a fault observer, and the second stage was used to locate the fault through the residuals created by the first stage ANN. The generated residuals showed promising performance when a 1 Hz sinusoidal reference with 20 mm amplitude was applied. This detected both amplifier fault and leakage fault. However, the Elman ANN is an old method, and is not recommended for industrial use any more, only for historical purposes [MathWorks, 2015a].

The authors from [L. Hongmei, 2006a] utilised the aforementioned approach in [L. Hongmei, 2006b], except the observer and identification network is a radial basis function ANN and is used to detect an amplifier fault and a leakage with similar increase as in [L. Hongmei, 2006a].

Based on this it is intriguing to test ANN-observer-based FDD by utilising networks that have not yet been tested.

In [Z. Yinshuo, 2013] an improved back-propagation training algorithm for ANN is proposed for FDD of a hydraulic system. By varying the steepest decent optimisation algorithm the convergence towards the minimum sum of squared errors is enhanced. The learning period was decreased from 20.679 s to 7.026 s with the use of an Intel Core2 Duo processor T7700 2.5 GHz and 2 GB RAM. It was further concluded, that the method had a strong learning ability.
Based on this it can be concluded that, the training period with current training algorithms and computation power is not an issue for development of FDD based on ANNs.

Comments

It has been observed that the general applied hardware versus time required for training speaks in the favour of ANN in general. Furthermore, the fact that it has been successfully applied in similar systems and the authors state that further work can be done are positive signs.

According to Mathworks the optimal ANN algorithm for predicting nonlinear time-series is the NARX algorithm and in [Diaconescu, 2008] the first conclusion was that the NARX recurrent ANN have the potential to capture and estimate the dynamics of nonlinear dynamic systems with different time delays. In this way previous data can be used along with the current and therefore a vector of each training variable can be utilised.

To determine the size of the NARX network a Generic Algorithm (GA) was applied by [Hosovsky, 2011], where the GA optimises both the number of neurons in the hidden layers of the ANN and the size of the regressor vector. The results achieved showed, that a much simpler model, which can be derived by the use of the GA algorithm is capable of achieving good accuracy compared to a trial and error method used for generating the network structure and dynamic order definition.

3.3 Part Conclusion

[R. Song, 2002] and [K. Cao, 2007] presented arguments that ANN was not interesting due to the time required for convergence and the difficulties with generalisation, hence, producing models with overfitting. According to [MathWorks, 2015c] and [MathWorks, 2015d] whom develops algorithms for ANN, the convergence period has been drastically improved and solutions to overfitting issues have been developed.

Utilisation of ANN may be done on both calculation of residuals and diagnosis of these, and as stated in [Isermann, 2006] the method may be utilised to approximate nonlinear relationships to any desired degree of accuracy. Furthermore, the ANN may be connected directly to specific system states and detect abnormalities e.g. in terms of parameter estimation.

The Kalman-filter schemes have shown good performance and application of this is thoroughly researched. On the contrary, limited research about ANN based FDD schemes is available.
Due to the general increasing interest of ANN and the presented arguments the authors are intrigued by utilising machine learning for FDD.
Since the system of interest contains several non-linearities, it is suggested to predict the time-series describing the system states, with the Nonlinear AutoRegressive network with eXogenous inputs (NARX) ANN with feedback connections enclosing several layers of the network by [MathWorks, 2015b].

Diagnosis is proposed to be done with the use of either residuals or the directly measured states, and it is desired investigated through simulation which topology has the most interesting potential.

The information gathered by the review of state of the art literature will be used to determine the final problem statement of this thesis.

CHAPTER 4

Problem Statement

The initial problem statement has been elaborated by analysing the research done within the area of Fault Detection and Diagnosis (FDD) for hydraulic servo systems. The conclusion of the preliminary analysis was that application of machine learning methods have interesting potential, hence the final problem statement is:

> *"Utilising standard artificial neural networks, how may faults be detected and diagnosed within a hydraulic servo system?*
> *The faults considered are: internal/external leakages, stuck servo valve and pressure transducer failure."*

To solve the presented problem, the project is divided into three main tasks shown below (each bullet point is an elaboration of the purpose and reasoning of each main task).

Problem Solution Strategy

1. Derivation of a mathematical model describing the dynamics of the hydraulic servo system.
   - A comparison of simulated results of a healthy and faulty system, will be used to investigate whether it is realistic to detect the chosen faults with the measurable system states: valve control signal, pressures, actuator- position & velocity. The measurable system states have been selected to test a novel configuration of symptoms, however, based on the system states used in [T. T. Le, 1997], [W. J. Crowther, 1998], [C. S. Byington, 2004], [H. Khan, 2005] & [W. Vianna, 2014].
   - Through simulations of the model, the different fault situations of interest may be investigated. The data collected from this investigation will be used to design the FDD schemes.
   - The mathematical model of the system will be used to evaluate the performance of the designed FDD schemes. This is considered a more feasible solution procedure, than spending time and money on testing it on lab equipment. Furthermore, the process of testing the different FDD schemes may be automated, so human interference time may be reduced.

2. Design of a model-based, and model-free FDD scheme and simulations of the selected fault modes.
    - Preliminary testing of the two FDD schemes will be done to identify the most promising scheme, which is proposed as candidate for further investigation. The preliminary tests will be done with default design parameters for the ANNs (proposed by MathWorks in [M. H. Beale, 2016]). This is done to obtain an understanding of the general performance of the ANNs.
    - Optimisation of the design parameters of the proposed FDD method, to determine if a performance increase is achievable. The optimisation will be done by utilising the MathWorks' GA optimisation.
    - The theoretical performance of the proposed FDD scheme will be evaluated at different system conditions, to investigate the capabilities of the scheme. The specific situations are:
        - Test with several patterns to evaluate if the scheme can recognise the same fault at different situations.
        - Similar test without velocity measurement to evaluate how this affects the FDD performance.
        - Test with measurement noise to evaluate the schemes' generalisation capabilities.
        - Test with a pseudo-random load, to investigate the robustness of the FDD scheme.
        - Test of gradual internal leakage over a period of 1 week, to investigate if a single fault can be diagnosed in an arbitrary time-frame.
3. Design of supervisory control and data acquisition (SCADA) software in LabVIEW for experimental tests.
    - The architecture of the SCADA software will be designed to include control algorithms, data logging, and FDD schemes. It will also be considered that the SCADA software may be used by future students working on the same set-up. Furthermore, it will contain different safety features, to protect the hardware and personal during tests. The above features are required in order to perform the necessary experimental work.
    - Experiments on the set-up will be used to verify the mathematical model, this may eliminate the need of actual data for every faulty situation and thereby the FDD scheme may be designed via simulation. The theoretical designed scheme will be tested on the actual set-up, to validate the schemes' performance.
    - If the FDD scheme designed via simulation cannot be validated, actual data will be collected from the system when a fault is introduced and test the proposed FDD scheme on the set-up.

Part II

Modelling & Fault Detection and Diagnosis Designs

CHAPTER 5

FORMULATION OF A MATHEMATICAL MODEL OF THE HYDRAULIC SYSTEM

Contents

In the following chapter, the governing equations used in modelling of the hydraulic servo system are presented. This model is needed in order to perform a theoretical investigation of fault detection and diagnosis and it must include the possibility of including the faults desired investigated, in a manner that mimics the actual behaviour of the faults. It is suggested to apply the model for theoretical testing and development of a Fault Detection and Diagnosis (FDD) scheme. Therefore, the derived model must be validated, however, it is not a requirement that it is verified to fit the actual test rig.
The mathematical model is based on the topology depicted in fig. 5.1.

5.1 Servo Valve Expressions

The following section consists of a derivation of the dynamic equations describing the servo valve. The used servo valve is a 40 l/min D633 by MOOG at nominal pressure of 35 bar [MOOG, 2009].

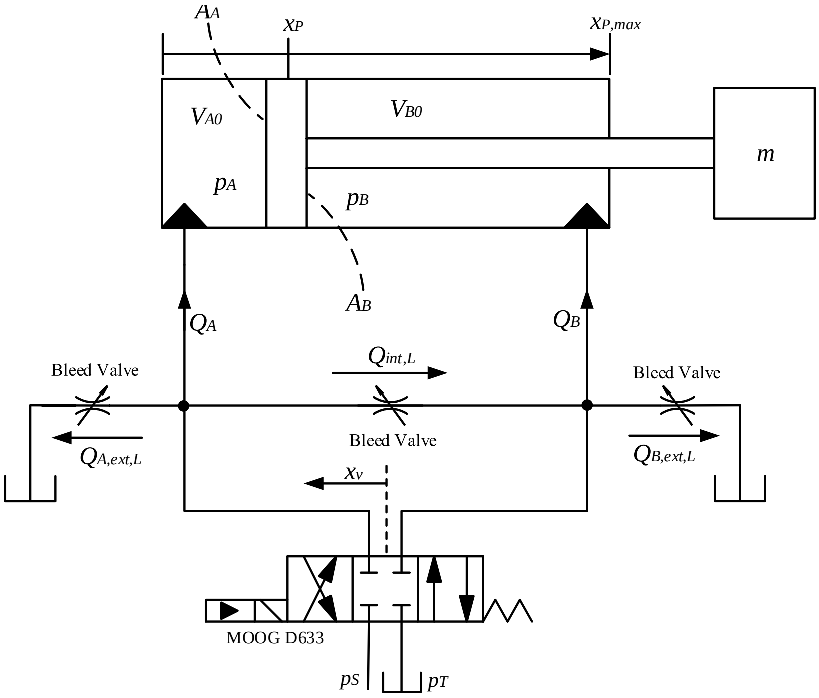


Figure 5.1: ISO schematic of the modelled hydraulic servo system with the applied notation.

The flows in and out of the lumped volumes A & B are controlled by a proportional valve, where the flows passing through the valve are given by (5.1)-(5.6).

$$Q_A = Q_{S\to A} + Q_{A\to T} \tag{5.1}$$

$$Q_B = Q_{S\to B} + Q_{B\to T} \tag{5.2}$$

$$Q_{S\to A} = sgn(p_S - p_A)Q_n x_V \sqrt{\frac{|p_S - p_A|}{\Delta p_n}} \quad , \quad x_V \geq 0 \tag{5.3}$$

$$Q_{A\to T} = sgn(p_A - p_T)Q_n x_V \sqrt{\frac{|p_A - p_T|}{\Delta p_n}} \quad , \quad x_V \leq 0 \tag{5.4}$$

$$Q_{S\to B} = sgn(p_B - p_S)Q_n x_V \sqrt{\frac{|p_S - p_B|}{\Delta p_n}} \quad , \quad x_V \leq 0 \tag{5.5}$$

$$Q_{B\to T} = sgn(p_T - p_B)Q_n x_V \sqrt{\frac{|p_B - p_T|}{\Delta p_n}} \quad , \quad x_V \geq 0 \tag{5.6}$$

where

p_S	is the pump pressure.	[Pa]
p_T	is the tank pressure.	[Pa]
Δp_n	is the nominal pressure drop across the valve.	[Pa]
Q_n	is the nominal flow through the valve.	[m^3/s]
x_V	is the normalised valve displacement.	[-]

The normalised valve displacement ranging from $-1 \leq x_V \leq 1$ can be controlled with a voltage signal ranging from $-10 \leq U_{V,DC} \leq 10$V with linear conversion to valve displacement.

Furthermore, it should be noted that the orifice equation is only valid when the flow has a full developed turbulent flow distribution. This is assumed valid with the expected flows and opening areas during normal operation.

5.1.1 Valve Dynamics

The transient dynamics of the valve spool displacement is approximated by a second order transfer function, which is based on the information available from the manufacturer, [MOOG, 2009]. The spool displacement (x_V) with regard to the voltage input ($U_{V,DC}$) is described by (5.7).

$$G_V(s) = \frac{x_V(s)}{U_{V,DC}(s)} = k_V \frac{1}{\frac{s^2}{\omega_{n,V}^2} + 2\zeta_V \frac{s}{\omega_{n,V}} + 1} \tag{5.7}$$

The gain $k_V = 0.1$ and describes the relationship between voltage input and spool reference. Furthermore, the estimated damping from the data-sheet is 0.8 and the bandwidth lies in a range of 25-60 Hz depending on reference signal. With the assumption that the valve will normally be in the range of ±25% of its maximum stroke, the bandwidth is 60 Hz.

5.2 Actuator Expressions

From the free body diagram shown in fig. 5.2 the different acting forces can be seen.

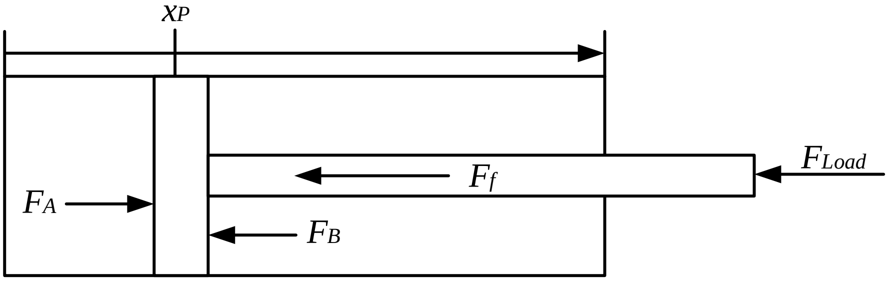


Figure 5.2: Free body diagram of the hydraulic actuator.

The acting forces, from fig. 5.2 are used in the force equilibrium, given by Newtons second law in (5.8).

$$\ddot{x}_P m = \left(\underbrace{p_A A_A}_{F_A} - \underbrace{p_B A_B}_{F_B} \right) - F_f - F_{Load} \tag{5.8}$$

The applied continuity equations are given by (5.9) and (5.10).

$$\dot{p}_A = \frac{Q_A - Q_{int,L} - Q_{A,ext,L} - \dot{V}_A}{A_A x_P + V_{A0}} \beta_{eff} \tag{5.9}$$

$$\dot{p}_B = \frac{Q_B + Q_{int,L} - Q_{B,ext,L} - \dot{V}_B}{V_{B0} - A_B x_P} \beta_{eff} \tag{5.10}$$

where

A_A	is the piston side area	[m^2]
A_B	is the rod side area	[m^2]
V_{A0}, V_{B0}	are constant volumes of the actuator chambers and hoses	[m^3]
$Q_{int,L}$	is the internal leakage flow	[m^3/s]
$Q_{A,ext,L}, Q_{B,ext,L}$	are the external leakage flows	[m^3/s]
x_P	is the displacement of the piston	[m]

The displacement of the piston in the cylinder is bounded by the dimensions of the actuator, meaning: $0 \leq x_P \leq x_{P,max}$.

The constant volumes are defined by utilising hose volumes, maximum stroke length and a constant dead volume. This is defined by (5.11) & (5.12).

$$V_{A0} = V_{dead} + V_{hose,A} \tag{5.11}$$

$$V_{B0} = A_B x_{P,max} + V_{dead} + V_{hose,B} \tag{5.12}$$

where

V_{dead}	is the dead volume inside the cylinders	[m^3]
V_{hose}	is the connecting hose volume	[m^3]

Here the dead volume is defined to be the volumes inside the cylinders that cannot be affected by the piston.

The change in volumes are described by (5.13) & (5.14).

$$\dot{V}_A = A_A \dot{x}_P \tag{5.13}$$

$$\dot{V}_B = -A_B \dot{x}_P \tag{5.14}$$

5.2.1 Friction Model

The two sources of friction considered are the one occurring when the piston is moving inside the cylinder, and secondly, the one that may be introduced by the non-dense medium in which the piston is moving.
Friction is a force resisting the relative motion of solid surfaces, and the normal types of friction are: dry friction, fluid friction, lubricated friction, skin friction and internal friction, described by [S. Andersson, 2006].

It is assumed, that the sliding surfaces are lubricated, hence, the friction will decrease with increased sliding speed until a point where mixed or full film situation is achieved [S. Andersson, 2006].
The Stribeck friction model includes dry- and fluid friction and is formulated by (5.15).

$$F_f(\dot{x}_P) = sgn(\dot{x}_P)\left(b_c|\dot{x}_P| + F_c + (F_s - F_c)e^{-\frac{|\dot{x}_P|}{c_s}}\right) \tag{5.15}$$

where

b_c	Viscous friction parameter.	[Nm/s]
c_s	Stribeck parameter	[-]
F_c	Constant Coulomb friction.	[N]
F_s	Maximum static friction.	[N]
F_f	Modelled friction.	[N]

The Stribeck model can provide decent estimation of the real friction in a hydraulic cylinder, since it contains both Coulomb-, viscous- and Stribeck friction.

When the sliding direction is changed the Stribeck model can cause numerical problems due to friction discontinuity at $\dot{x}_P = 0$. However, this issue has been countered by combining the Stribeck model with a hyperbolic tangent to accommodate the transition in the change of sliding direction. The modelled friction is then given by (5.16).

$$F_f(\dot{x}_P) = \tanh(k_{tan}\dot{x}_P)\left(F_c + (F_s - F_c)e^{-\frac{|\dot{x}_P|}{c_s}}\right) + b_c\dot{x}_P \qquad (5.16)$$

This rewriting causes inaccurate Stribeck effect at low velocities (in between the chosen k_{tan}). However, the test sequences will not require operation around this velocity for long periods, i.e the piston will be in motion for most of the tests.
The modelled friction force with parameters determined in chap. 11 can be seen from fig. 5.3.

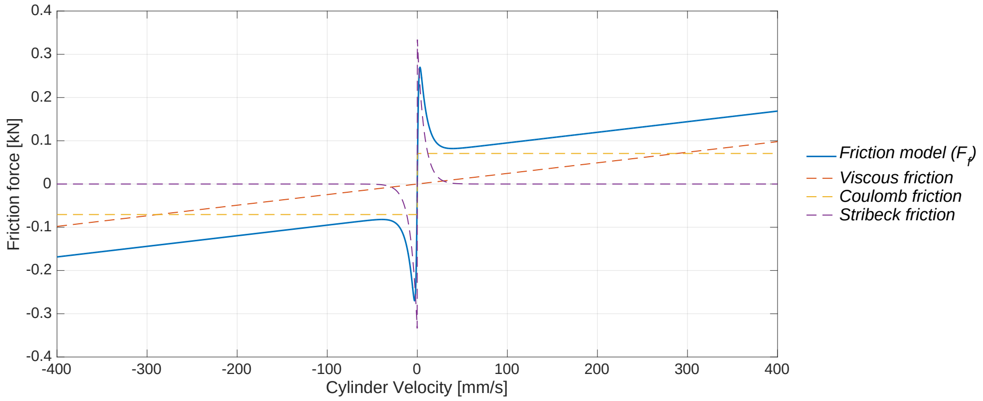


Figure 5.3: The total friction force as function of piston velocity. Furthermore the viscous-, Coulomb- & Stribeck- friction contributions are depicted with dotted lines.

It is observed from the figure that introduction of hyperbolic tangent results in a lower peak when direction of velocity changes.

5.2.2 Effective Bulk Modulus

The effective bulk modulus describes the stiffness of the fluid and is a function of pressure, temperature and contamination of the oil (air, metal debris and other impurities). The pressure used is the one of the respective volume. The derivation of the effective bulk modulus is explained in detail in [T.O.Andersen, 2003] and the expression is given by (5.17).

$$\beta_{eff}(T,p,\epsilon_A) = \frac{1}{\frac{1}{\beta_{oil}} + \epsilon_A(p)\left(\frac{1}{\beta_{air}} - \frac{1}{\beta_{oil}}\right)} \qquad (5.17)$$

The air stiffness can be described by (5.18).

$$\beta_{air}(p) = c_{ad}\, p \qquad (5.18)$$

where

c_{ad}	is the adiabatic constant of air, 1.4	[-]
ϵ_A	is the volumetric ratio of free air in the oil.	[-]
p	is the absolute pressure in the chamber.	[bar]
p_{atm}	is atmospheric pressure.	[bar]
T	is the temperature of the fluid.	[K]

The β_{oil} used for simulation is from [H. Rahn, 2015] where the stiffness of oil was experimentally estimated to be 6500 bar with reference volumetric ratio of free air at 0.0038%. This is used as an initial guess and will be adjusted when the verification tests are conducted in chap. 11.

The effective stiffness of hydraulic oil is influenced by the amount of free air in it. Air is much more compressible than oil and therefore it can have a strong influence on the effective stiffness. To estimate the stiffness it is assumed that the air molecules does not dissolve in the oil, meaning a constant amount of free air molecules exist in the oil.
The volumetric ratio of free air, ϵ_A, can be calculated by (5.19) where adiabatic conditions are assumed, [T.O.Andersen, 2003].

$$\epsilon_A(T,p) = \frac{1.0}{\left(\frac{1.0-\epsilon_{A,0}}{\epsilon_{A,0}}\right)\frac{\rho_0}{\rho(T,p)}\left(\frac{p_{atm}}{p}\right)^{\frac{-1}{c_{ad}}} + 1.0} \tag{5.19}$$

where

$\epsilon_{A,0}$	is the reference volumetric ratio of free air in the oil at atmospheric pressure.	[-]
p	is the absolute pressure in the chamber.	[bar]
p_{atm}	is the atmospheric pressure.	[bar]

The density of oil decreases when the temperature increases, but increases when pressure increases. This dependency can be described by using the empirical based Dow and Fink equation shown in (5.20), [T.O.Andersen, 2003].

$$\rho(T,p) = \rho_0(T)\left(1 + A_\beta(T)p - B_\beta(T)p^2\right) \tag{5.20}$$

$$A_\beta(T) = (-6.72\cdot 10^{-4}T^2 + 0.53T - 36.02)10^{-6} \tag{5.21}$$

$$B_\beta(T) = (2.84\cdot 10^{-4}T^2 - 0.24T + 57.17)10^{-9} \tag{5.22}$$

$$\rho_0(T) = \frac{\rho_{15}}{1+\alpha_{th}(T-(15+273.15))} \tag{5.23}$$

where

α_{th}	is the thermal expansion coefficient	[°C^{-1}]
$A_\beta(T)$	is the Dow and Fink temperature coefficient.	[bar^{-1}]
$B_\beta(T)$	is the Dow and Fink temperature coefficient.	[bar^{-2}]
$\rho_0(T)$	is the density at atmospheric pressure.	[kg/m^3]
ρ_{15}	is the density at atmospheric pressure and 15°C	[kg/m^3]

This estimation can be used to obtain the density at any given temperature and pressure while the oil density at atmospheric pressure and the thermal expansion coefficient are known.
$\alpha_t h$ is independent of oil temperature and pressure and is between 0.0065 and 0.007.

The modelled effective stiffness is illustrated graphically in fig. 5.4. The results are shown with constant temperature of 40°C at different volumetric ratios.

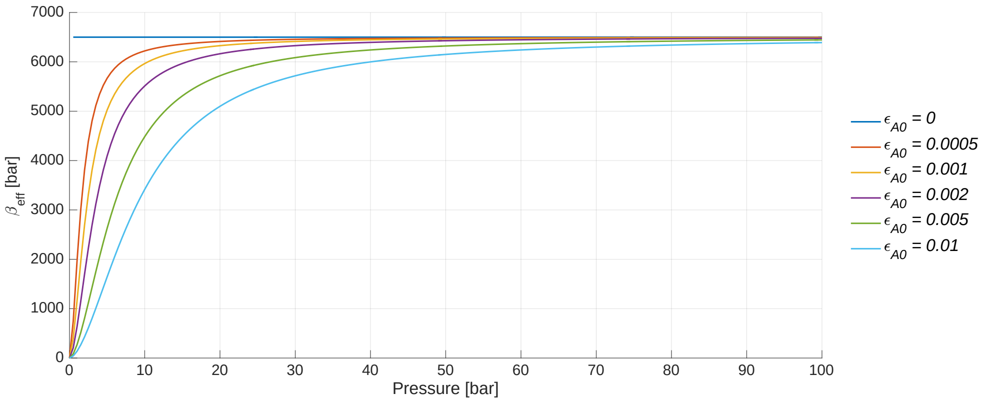


Figure 5.4: Effective stiffness (β_{eff}) as function of pressure at different volumetric ratios, $T = 40°\text{C}$.

It is observed that the effective stiffness will be affected by the air in the fluid and yields that a higher concentration of air gives a lower stiffness. However, increase in pressure causes the impact of entrapped air to become less significant.

5.3 Leakage Expressions

In the system schematic on fig. 5.1 three leakage flows are depicted. By opening the bleed-valves, both internal and external leakages can be emulated. Double-acting cylinders normally include almost no cross-port leakage, and this leakage is therefore disregarded.
The required leakages lie in a range of 0-1.5 l/min, and with the expected pressure drops this flow is obtained at low orifice openings. When applied to (5.24) these levels can cause Reynolds numbers in the range, at which the flow distribution changes between laminar and turbulent.

$$Re = \frac{\rho v_{avg} d_H}{\mu} \quad ; \quad d_H = \sqrt{\frac{4A(x)}{\pi}} \tag{5.24}$$

where

$A(x)$	is the variable opening area of the orifice.	[m^2]
d_H	is the hydraulic diameter of the orifice.	[m]
μ	is the fluid dynamic viscosity.	[Ns/m^2]
ρ	is the fluid density.	[kg/m^3]
v_{avg}	is the average fluid velocity trough the orifice.	[m/s]

Due to the expected range of Reynolds numbers the flow through the bleed valves will be described by a model which evaluates and uses the flow distribution to calculate the flow. This is described by (5.25).

$$Q_L = C_D(Re)A(x)\sqrt{\frac{2}{\rho}|\Delta p| sign(\Delta p)} \tag{5.25}$$

where

$C_D(Re)$	is the discharge coefficient of the orifice as a function of the Reynolds number.	[-]
Δp_{cr}	is the minimum pressure drop resulting in turbulent flow.	[Pa]

The change of discharge coefficient for a rounded off orifice is described by (5.26). It is assumed based on [T.O.Andersen, 2003] that a turbulent flow within a valve requires a Reynolds number above ≈ 625.

$$C_D(Re) = \begin{cases} \frac{C_D}{\sqrt{Re_{cr}}}\sqrt{Re} & \text{for } Re \leq 625 \\ C_D & \text{for } Re > 625 \end{cases} \tag{5.26}$$

where

C_D	is the discharge coefficient of the orifice at turbulent flow.	[-]
Re_{cr}	is the critical Reynolds number.	[-]

5.3.1 Internal Leakage

The internal leakage between lumped volume A & B is described by the orifice equation (5.25) and the pressure drop is defined by (5.27).

$$\Delta p_{int,L} = (p_A - p_B) \tag{5.27}$$

5.3.2 External Leakage

The pressure drops for the external leakages are described by (5.28) and (5.29), which leaves lumped volume A & B respectively.

$$\Delta p_{A,ext,L} = (p_A - p_{atm}) \tag{5.28}$$

$$\Delta p_{B,ext,L} = (p_B - p_{atm}) \tag{5.29}$$

The flows are unidirectional unless cavitation occurs.

5.4 Overall Model Structure

The considerations necessary to describe the hydraulic servo system have been presented and the interconnection of these are illustrated in the overall model structure in fig. 5.5.

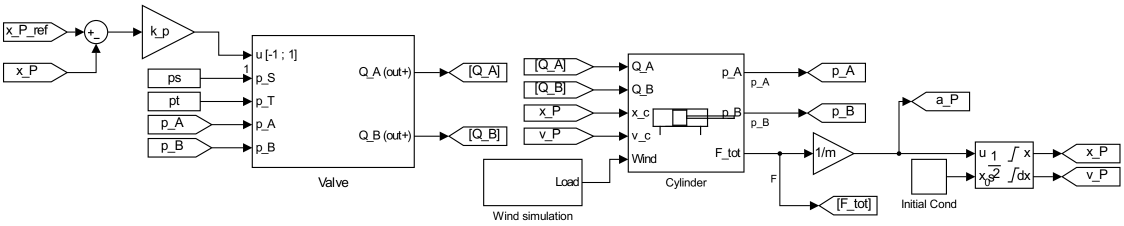


Figure 5.5: Simulink schematic of the hydraulic servo system model structure.

The model includes pre-defined initial conditions on the variables: A & B pressures, piston position and velocity. Furthermore, the additional assumptions have been applied:

- Supply and tank pressure are measurable.
- Constant temperature and air contamination of the fluid.
- Hoses are assumed stiff and lossless.

Due to an expected varying supply pressure on the test rig, the model is only feasible if the supply pressure can be measured and applied in the simulation.
It should be noted that in order to achieve constant supply pressure a properly dimensioned

accumulator (4 l) is placed as close as possible to the main connection line, which from [Mohieddine Jelali, 2003] should be less than 0.05 m. However, the applied pump is not pressure controlled and therefore constant supply is not a guarantee.
The applied system parameters are presented in app. I.

The temperature of the fluid in the pump system is monitored, and experimental tests are only conducted if the temperature is 50-60°C. Therefore an assumption of constant temperature is valid and the range of 10°C is not considered to have any noticeable effect on either density or fluid stiffness based on [T.O.Andersen, 2007].
The fluid contamination is not expected to change over the period of time in which the experiments are conducted.

In [T.O.Andersen, 2007] it is stated that the decreased efficiency caused by loss in hoses is normally disregarded. If pressures rise to very high levels and hoses are not sufficiently dimensioned this can contribute with a decreased effective stiffness due to flexibility of the hose. However, with the desired pressure magnitudes and the short length of applied hoses this impact is disregarded for the sake of model simplicity.

The benefits of the presented assumptions are that both model and fault diagnosis can be developed more rapidly. The disadvantage is an increased likelihood of model uncertainties.

5.5 Part Conclusion

The governing equations describing the dynamics of the hydraulic servo system shown in fig. 5.1 have been presented.

Modelling of the components included in the system is well documented and the mathematical model is therefore considered valid for preliminary testing, although the actual system parameters are not known.
An estimation of fluid properties and frictions have therefore been applied based on previous experience and this has been combined with information from data-sheets. Hereafter, the system's soft parameters are estimated by experimental tests as described in chap. 11.
This is done to utilise the available time most efficiently, while the test rig is being manufactured. The non-verified model will be used to extract data for a healthy and faulty system, which can be used to do preliminary testing and development of fault detection and diagnosis, hence a proof of concept can be established.

CHAPTER 6

Simulated Fault Analysis

Contents

In the following chapter it will be investigated whether the leakage faults will result in theoretical abnormalities exceeding the level of measurement uncertainty. The test sequence for the preliminary fault detection and diagnosis tests will be explained.

6.1 Description of the Preliminary Test- & Fault Sequence

The preliminary position trajectory consists of a sinusoidal reference with constant frequency (3 rad/s) and constant amplitude (0.01 m). This is chosen due to its simplicity and repeating nature, making the influence of each fault comparable.

The preliminary fault sequence will include **(F.1)**-**(F.3)**.

(F.1) Internal rupture/leakage between the lumped volumes.
(F.2) External rupture/leakage from volume A .
(F.3) External rupture/leakage from volume B.

The impact caused by these faults is found by observing changes on the measurable system states during simulation. These states are given by **(c.1)**-**(c.4)**.

(c.1) Chamber pressures (p_A, p_B) with a range of [1:180] bar
(c.2) Piston position (x_P) with a range of [0:0.5] m
(c.3) Piston velocity ($\dot{x}_P$) with a range of [-0.1769:0.1327] m/s (limited by the valve's nominal flow of 40 l/min)
(c.4) Valve spool position (x_V) with a range of [-1:1]

The approach is to have a *healthy* and *faulty* system and do a comparison of the simulated noise free states, to investigate whether the residuals are in a measurable range. In reality the residuals are sensitive to measurement noise and model inaccuracies, which is why the residuals have to lie outside thresholds ensure that an actual fault has occurred. The thresholds are defined as shown in **(d.1)**-**(d.4)** based on the applied transducers measurement resolution.

(d.1) Pressures in the test cylinder can be measured with a certainty of ±0.3% of the maximum pressure (250 bar), i.e. 0.75 bar.
(d.2) Piston position is measurable with a certainty of ±0.01% of the maximum stroke (1 m) i.e. 0.1 mm.
(d.3) Piston velocity can be measured with a certainty of ±0.5% of the actual velocity.
(d.4) Valve spool position is given with a resolution of 0.02 mA with the LVDT output range being 4-20 mA.

The concept of generating ideal residuals is depicted in fig. 6.1.

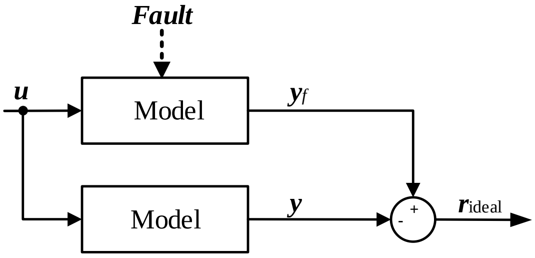


Figure 6.1: Structure of the generation of ideal residuals with comparison of the outputs from the model and from the model introduced to a fault.

In fig. 6.1 the vector $\boldsymbol{y}_f$, represents the faulty system states and $\boldsymbol{y}$ is the *normal* output states of the system.

Internal and external leakage in a hydraulic servo system have been proven detectable at around 0.25 l/min by [L. An, 2014b]. Therefore, a similar level is desired investigated.

The leakage is simulated with the orifice equation and by considering the expected pressure drops, an estimate of required opening area may be determined to obtain the correct magnitude of leakage flow.

The test sequence is designed in such a way that the system returns to a healthy state before introducing another fault, each fault is introduced one at a time in equal increments as shown in fig. 6.2.

As seen in fig. 6.2 each fault has three opening areas. The reason for having these degrees of severity, is to include the system's behaviour at several leakage magnitudes. Thereby the diagnosis algorithm may be able to determine both which fault has occurred and its severity.

Fault **(F.1)** has been simulated with opening areas ($A_{int,L}$): 0.1, 0.2 & 0.3 mm^2. This is determined so that the minimum flow is ≈0.3 l/min with the simulated load pressure levels.
Fault **(F.2)** has been simulated with opening areas ($A_{ext,A,L}$): 0.04, 0.08 & 0.12 mm^2. This is chosen in a similar manner as for **(F.1)**. The first opening corresponds to ≈0.2 l/min.
Fault **(F.3)** has been simulated with opening areas ($A_{ext,B,L}$): 0.03, 0.06 & 0.09 mm^2. With the same reasoning as for **(F.2)**.

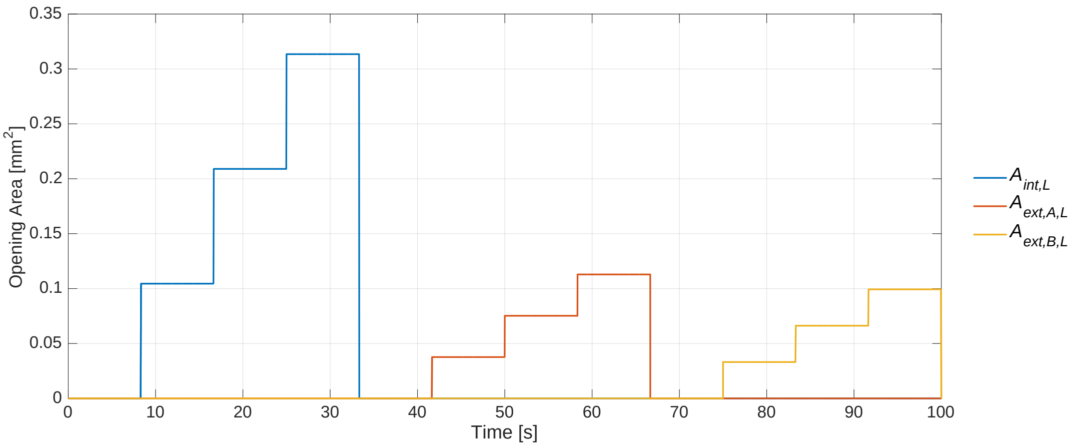


Figure 6.2: The fault sequence applied for preliminary testing.

6.2 Simulated Residuals Analysis

The residuals considered are based on the analysis presented in sec. 7.3 and simulation with the defined fault sequence gives the ideal residuals ($\boldsymbol{r}_{ideal}$) shown in fig. 6.3.

Based on the results shown in fig. 6.3 it is observed that all leakage faults are visible in the lumped volume pressures, however, **(F.1)** does not result in pressure deviations consistently outside the measurement uncertainty. **(F.2)** & **(F.3)** are significantly outside the level of uncertainty.
The position residual is outside the uncertainty span at **(F.1)**, however, the lowest severity of **(F.2)** & **(F.3)** is in the periphery of what is expected measurable. The system is operated in Closed-Loop, and if the applied control strategy is robust to system changes it is not expected that the position residual will contribute with any useful information. However, this additional compensation will propagate in the required valve position, which is why the residual of this is included.
The valve position residual indicates that the faulty system is endeavouring to compensate.
The velocity shows a significant transient residual outside the uncertainty span when the specific faults are introduced/removed. The static situation does show a deviation outside the uncertainty, although it is not consistent.

6.3 Part Conclusion

Based on the analysis of the leakages it is concluded that abnormalities may occur in the measurable variables p_A, p_B, x_P, $\dot{x}_P$ & x_V. The magnitudes of all the residuals are in a promising range compared with the thresholds from **(d.1)**-**(d.4)**, however, the lowest severity of fault is considered to be in the periphery of what is expected measurable and it is therefore not feasible to attempt diagnosing faults below this. It is decided to advance with the specified fault magnitudes to test whether the faults can be found theoretically.
The knowledge about the faults' affection of the specific states will be used for the design of Fault Detection and Diagnosis presented in the following chapter.

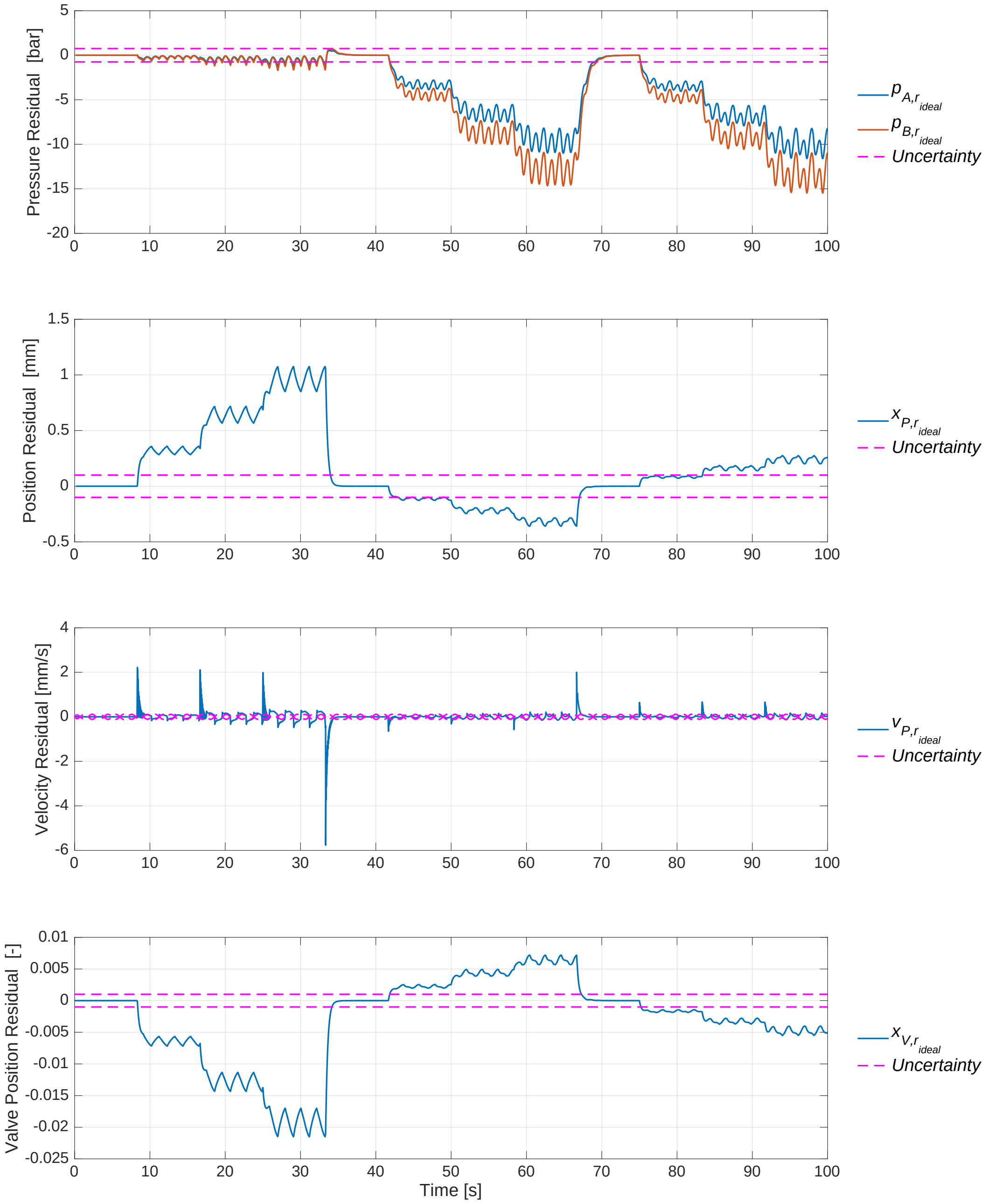


Figure 6.3: Simulated results of the residuals when a sine trajectory and the presented fault sequence is applied. The thresholds from **(d.1)**-**(d.4)** are shown with dotted lines.

CHAPTER 7

ARTIFICIAL NEURAL NETWORKS APPLIED FOR FAULT DETECTION & DIAGNOSIS

Contents

In the following chapter an analysis and discussion of Artificial Neural Networks (ANN) and how these can be used for fault detection and diagnosis (FDD) are presented. A comparison of two ANN based FDD schemes is presented and one will be selected for further development based on the simulated performance.

7.1 Basics of Artificial Neural Networks

Essentially an ANN is a composition of mathematically modelled neurons, which is based on knowledge from neuroscience, specifically about the interaction of neurons in the biological brain [W. Mcculloch, 1943].
All these artificial neurons are interconnected, which makes it possible to describe a relationship between certain input and output signals [Rosenblatt, 1958].

7.1.1 The Artificial Neuron

An artificial neuron model has gained increased interest over the years, from the world of biology and into the world of practical mathematics.
The mathematical model describing the artificial neuron is the fundamental building block for ANNs and is illustrated in fig. 7.1.

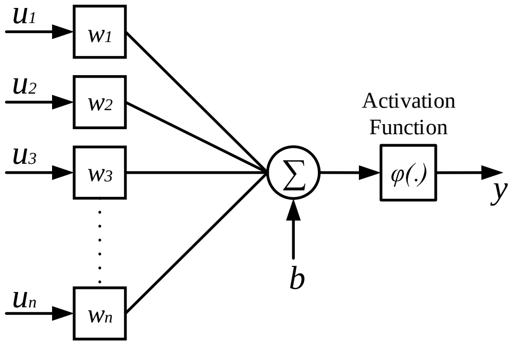


Figure 7.1: Structure of a basic neuron with weights (w), bias (b) and activation function ($\varphi(.)$), with inspiration from [Watton, 2007].

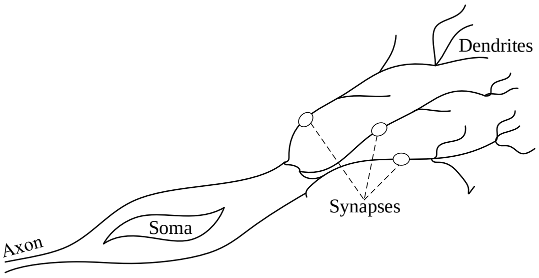


Figure 7.2: Structure of a single biological neuron, with inspiration from [S. Rajasekaran, 2012].

The composition in fig. 7.1 and 7.2 show that n number of inputs, which in biology are referred to as *dendrites*, are multiplied by individual gains (w). These gains are in biology referred to as *synapses*. A summation of the weighted input signals from the *dendrites* is performed where a bias (b) can be included, this is inspired by the biological *soma*. Lastly, the weighed sum signal is passed through an activation function inspired by the biological *axon*, which generates the total output, and passes it to the next layer of neurons.
The weighed sum signal through the activation function, may be designed to scale the output in a manner that fits the application and as the name implies, be used to activate the signal generated by the *soma*.
The most common activation functions within the artificial neuron are: linear, step and sigmoid functions, which are presented in app. C. The neurons use normalised values, and the activation functions are therefore often restricted between [-1:1]. [Hornik, 1991] showed that it is not the specific activation function, but rather the multilayer feedforward architecture itself which gives neural networks the potential of being universal approximators. This is also referred to as the universal approximation theorem.

The Universal Approximation Theorem

The universal approximation theorem claims that the standard multilayer feedforward network of a single hidden layer containing finite number of hidden neurons, and with arbitrary activation function are universal approximators. In mathematical terms the theorem is:

Theorem 1 *Let $\varphi(.)$ be a non-constant, bounded and monotonically-increasing continuous arbitrary activation function. Let I_m denote the m-dimensional unit hypercube $[0 \quad 1]^m$. The space of continuous functions on I_m is denoted by $C(I_m)$. Then, given any function $f \in C(I_m)$ and $\varepsilon > 0$, there exists an integer, N, real constants, $v_i, b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$, where*

$i \in \{1, \cdots, N\}$, such that we may define:

$$\tilde{f}(\boldsymbol{x}) = \sum_{i=1}^{N} v_i \varphi\left(\boldsymbol{w}_i^T \boldsymbol{x} + b_i\right)$$

which is an approximate realisation of the function f, where f is independent of φ.

$$|\tilde{f}(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon$$

when $\boldsymbol{x} \in I_m$ or any other compact subset $\in \mathbb{R}^m$, i.e. functions of the form $\tilde{f}(\boldsymbol{x})$ are dense in $C(I_m)$.

by [Cybenko, 1989] where appropriate proof and lemma are given. The theorem in short states that the ANN topology is an universal approximator regardless of activation function as longs as it fulfils the above stated conditions.
The choice of activation function in this thesis is therefore solely based on it being differentiable, making gradient based training algorithms possible.

For the sake of perspective it is further stated that when utilising the artificial neurons in an engineering practise, the necessary amount of neurons can vary from a few to several hundreds, depending on application. In contrast to the billions of neurons present in the human brain [S. Rajasekaran, 2012].

7.2 Introduction to ANN used for FDD

In this thesis, two approaches are being reviewed either based on system state residuals combined with fault diagnosis or alternatively, if system measurements should be used directly to do fault diagnosis. These approaches are depicted in fig. 7.3 & 7.4.

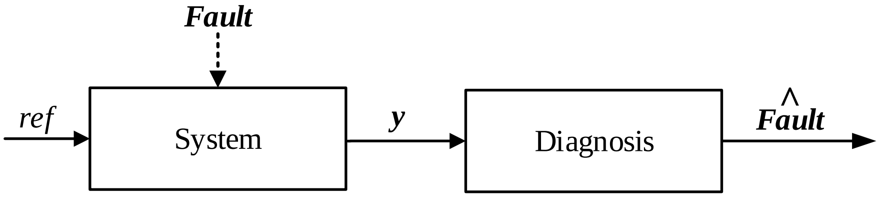


Figure 7.3: Diagram showing the overall structure of the Fault Detection and Diagnosis (FDD) scheme, for a data driven approach.

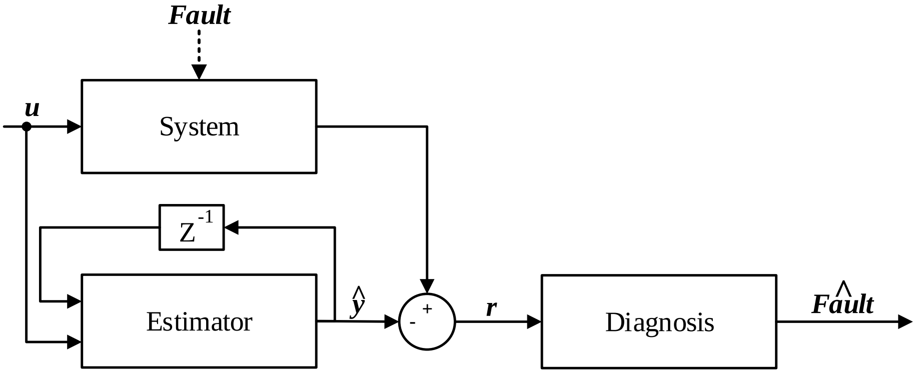


Figure 7.4: Diagram showing the overall structure of the FDD scheme, for a model-based approach.

The potential of these approaches applied in similar hydraulic servo systems was presented in the literature review in chap. 3. None of the approaches yielded superior performance, which is why both are investigated.

Model-Based FDD

In the model-based FDD the relationship between several measured system states ($\boldsymbol{y}$) and estimated (predicted) system states ($\hat{\boldsymbol{y}}$) are used to extract information about the system's health (can also be referred to as *features*), an abnormality will be detected if actual and predicted states diverge.

The detectable faults depend on the model accuracy, since a more accurate model will give a higher certainty that change in the residual truly is caused by a fault. Highly accurate time-series predicting models have been presented in the literature by using various types of ANNs, which serves as a promising argument for studying this scheme.

To investigate this further it is decided to test a Nonlinear AutoRegressive eXogenous (NARX) model, since [R. Isermann, 2010] argues that the NARX model is effective for nonlinear system identification problems.

The NARX residuals ($\boldsymbol{r}_{NARX}$) will be compared with the ideal residuals ($\boldsymbol{r}_{ideal}$), to determine the NARX models performance. The processes for generating these residuals are depicted in fig. 7.5.

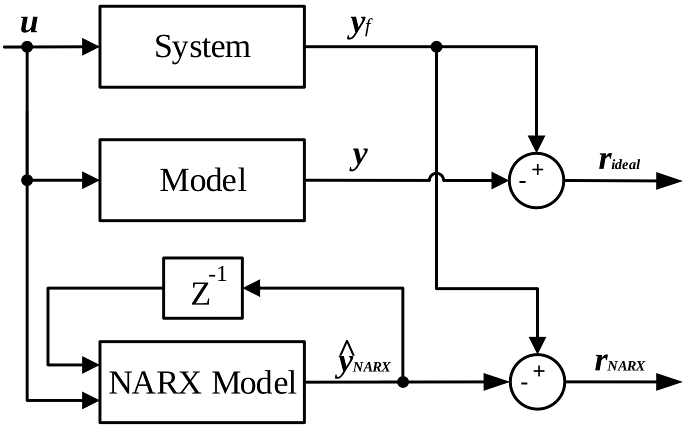


Figure 7.5: Diagram of the residual generation of $\boldsymbol{r}_{NARX}$ & $\boldsymbol{r}_{ideal}$.

Data Driven FDD

The data driven FDD approach in its simplest form utilises measured data of a given set of system states, to estimate the given output(s) containing the faults. Promising results with data driven ANNs used directly as parameter estimators of leakage coefficients, Coulomb frictions and supply pump failures were presented in the articles [T. T. Le, 1997] & [W. J. Crowther, 1998] described in the literature review in chap. 3.

The networks can either be trained with theoretical data or with data collected from a physical system where faults can be introduced in a controlled environment.

7.3 Architecture of NARX

The NARX network is a recurrent dynamic network, with feedback connections closing multiple loops within the different layers in the network. The NARX network can be configured in two different ways, depending on application, the two architectures are shown in fig. 7.6. Where the TDL is a Tapped Delay Line.

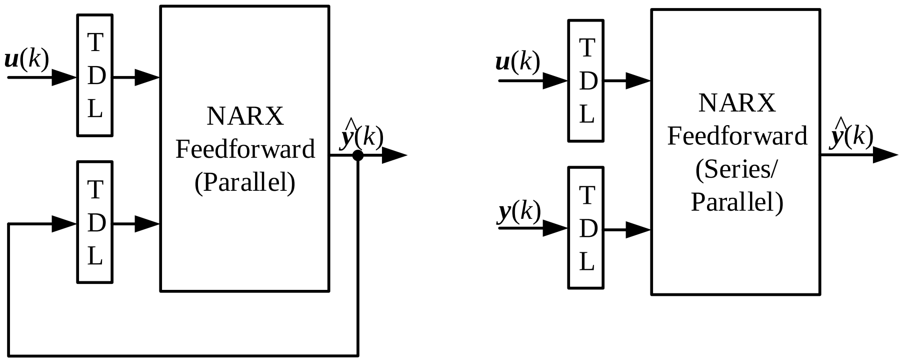

$u(k)$
TDL
TDL
NARX Feedforward (Parallel)
$\hat{y}(k)$
$u(k)$
$y(k)$
TDL
TDL
NARX Feedforward (Series/ Parallel)
$\hat{y}(k)$

Figure 7.6: The two possible architectures for a NARX network, with inspiration from [MathWorks, 2016b].

The **parallel** network uses the output of the NARX network to be an estimate of the output of the nonlinear system. The estimated output is fed back into the feedforward ANN as part of the NARX network.

The **series/parallel** can be used when the true output is available during simulation of the network. The true output from the nonlinear system is used instead of the estimated output feedback, which gives a more accurate prediction [MathWorks, 2016b].

However, in fault detection the series/parallel architecture can have the drawback of being "too accurate", hence predicting the actual output and therefore not detecting the fault.

For this thesis it is desired to predict the *healthy* behaviour of the hydraulic servo system, hence if a fault occurs in the real system the NARX model is not supposed to adapt to the behaviour of the given fault. Therefore the parallel architecture has been chosen.

7.3.1 Input- & Output- Variables

In order to select the NARX input- & output- variables, it is convenient to use some of the information gained in the literature review in chap. 3. In [W. Vianna, 2014] it is proposed to use measurements of fluid temperature, fluid pressure and reservoir level to detect leakages. Likewise in [C. S. Byington, 2004] pressures are used along with servo valve control signal to detect leakages. [T. T. Le, 1997] suggests application of line flows and pressures for leakage detection. In [W. J. Crowther, 1998] it is proposed to apply the pressure drop, valve control signal and the piston position to diagnose incorrect supply pressure, increased actuator cross-line leakage and increased friction. Lastly, [H. Khan, 2005] utilises control signal as input and the actuators velocity as output to detect faults.

Based on this information, it is decided to do a combination of what has been used in the previous research, and what is physical possible on the set-up. Since the set-up will be powered by

a central pumping station, which also supplies other applications, some of the measurements may not give useful information. It is expected, that the central pumping station's reservoir will have a constant temperature, since it is connected to a central cooling system. Furthermore, the reservoir level will not give useful information in regard to the faults selected, since all external leakages from the set-up will be leaked to the reservoir.
The use of line flow sensors might produce a more stable result, however, accurate flow sensors are expensive and will introduce additional effort when designing the pitch system, which is why a minimal number of sensors is desired.
Therefore it is considered to combine measurements of: $p_A, p_B, x_P, \dot{x}_P, x_V$ as output, with the applied reference, $x_{P,ref}$ as input. It has been considered to omit the velocity measurement, since it may be cumbersome and since it is directly connected with the position. It is however, not obvious which states contribute with sufficient knowledge about the system, which is why the contribution when the velocity is available will be investigated.

Scaling of Variables

Since ANNs work with scaled variables, all the variables have to be scaled. The scaling of the variables have been done as described by (7.1).

$$\bar{x}_{P,ref} = \frac{x_{P,ref}}{x_{P,ref,max}} \tag{7.1}$$

This procedure is done for every applied variable and therefore the $scaling$ notation is omitted for every variable in this chapter.

The relevant input and output variables have been selected and normalised, and the detailed structure of the parallel NARX network is shown in fig. 7.7.

Governing Equations of NARX ANN

In fig. 7.7 the input regressor vector for the network is defined by (7.2).

$$\boldsymbol{u}(k) = \begin{bmatrix} x_{P,ref}(k) & x_{P,ref}(k-1) & \cdots & x_{P,ref}(k-d_u) \end{bmatrix}^T \tag{7.2}$$

where d_u is the memory delay.
The output regressor vector can be written as shown in (7.3).

$$\begin{aligned} \hat{\boldsymbol{y}}(k) = \Big[& \hat{p}_A(k) \quad \hat{p}_A(k-1) \quad \cdots \quad \hat{p}_A(k-d_y) \\ & \hat{p}_B(k) \quad \hat{p}_B(k-1) \quad \cdots \quad \hat{p}_B(k-d_y) \\ & \hat{x}_P(k) \quad \hat{x}_P(k-1) \quad \cdots \quad \hat{x}_P(k-d_y) \\ & \hat{\dot{x}}_P(k) \quad \hat{\dot{x}}_P(k-1) \quad \cdots \quad \hat{\dot{x}}_P(k-d_y) \\ & \hat{x}_V(k) \quad \hat{x}_V(k-1) \quad \cdots \quad \hat{x}_V(k-d_y) \Big]^T \end{aligned} \tag{7.3}$$

Where the memory delays should be chosen as shown in (7.4).

$$\begin{aligned} d_u &\geq 1 \\ d_y &\geq 1 \\ d_u &\leq d_y \end{aligned} \tag{7.4}$$

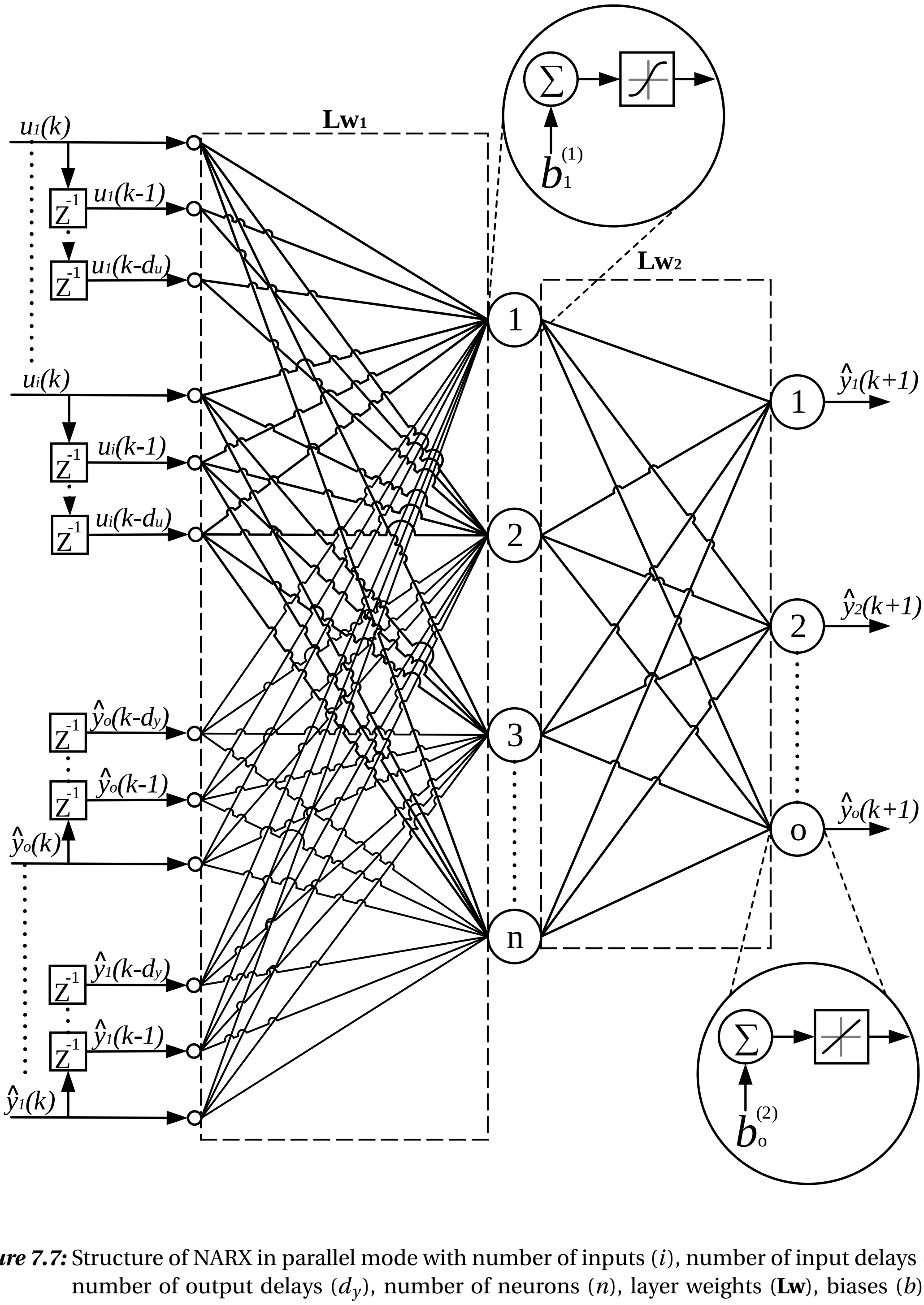


Figure 7.7: Structure of NARX in parallel mode with number of inputs (i), number of input delays (d_u), number of output delays (d_y), number of neurons (n), layer weights (**Lw**), biases (b) and number of outputs (o).

The main purpose of the NARX is to realise a nonlinear approximation from i-dimensional input space to o-dimensional output space, where i is the number of inputs and o is the number of outputs. This is described by the total network function shown in (7.5) which describes the network from fig. 7.7.

$$\hat{\boldsymbol{y}}(k+1) = \mathbf{L}_{w2} \tanh\left(\mathbf{L}_{w1}\xi(k) + \boldsymbol{b}^{(1)}\right) + \boldsymbol{b}^{(2)} \tag{7.5}$$

where

$$\mathbf{L}_{w1} = \begin{bmatrix} w_{1,1}^{(1)} & \cdots & w_{1,(d_u+1)i+(d_y+1)o}^{(1)} \\ \vdots & \ddots & \vdots \\ w_{n,1}^{(1)} & \cdots & w_{n,(d_u+1)i+(d_y+1)o}^{(1)} \end{bmatrix} \qquad \mathbf{L}_{w2} = \begin{bmatrix} w_{1,1}^{(2)} & \cdots & w_{1,n}^{(2)} \\ \vdots & \ddots & \vdots \\ w_{o,1}^{(2)} & \cdots & w_{o,n}^{(2)} \end{bmatrix} \tag{7.6}$$

$$\boldsymbol{b}^{(1)} = \begin{pmatrix} b_1^{(1)} \\ \vdots \\ b_n^{(1)} \end{pmatrix} \qquad \boldsymbol{b}^{(2)} = \begin{pmatrix} b_1^{(2)} \\ \vdots \\ b_o^{(2)} \end{pmatrix} \tag{7.7}$$

The combined regressor, $\xi(k)$, is shown in (7.8) and given by (7.2) and (7.3).

$$\xi(k) = \begin{pmatrix} \boldsymbol{u}(k) \\ \hat{\boldsymbol{y}}(k) \end{pmatrix} \tag{7.8}$$

The hyperbolic tangent (sigmoid) function or activation function described by (C.4) in app. C is used to normalise the output into the interval $[-1,1]$. The utilisation of the tanh function, introduces the nonlinear functionality of the NARX network. Furthermore, the training method utilised (explained in sec. 7.3.2) requires computation of the gradient of the error function, hence differentiability of the error function has to be guaranteed. This is why the sigmoid activation function is chosen.

The main purpose of the training algorithm is to optimise all the weights shown in (7.6). This process is described later in the chapter.

Number of Neurons

In [W. J. Crowther, 1998], [T. T. Le, 1997], [S. He, 2000], [A. El-Betar, 2006] and [J. Anzurez-Marin, 2009] a manual "trial and error" approach was used to determine an appropriate number of neurons. Their objective was to have good generalisation behaviour (avoiding overfitting) and accurate representation of the system (avoid underfitting), while decreasing the convergence period. However, [M. Karpenko, 2002], Yi-Hui [2007] and [Hosovsky, 2011] have presented approaches with utilisation of different optimisation algorithms to perform this decision.

There does not exist any specific design rules on the matter of choosing the number of either neurons or delays. However, there does exist a sweet-spot between underfitting and overfitting by applying an ANN model with sufficient complexity with respect to the problem. This is elaborated further in sec. 8.2.

In the preliminary test, the number of neurons has been selected with inspiration from the default values determined by MathWorks, this will later be expanded with the use of an optimisation algorithm.

Number of Delays

The number of input and output delays (d_u and d_y) describes the length of the discrete time series $y(k)$ for every step during training, where the discrete time shift operator Z^{-1} denotes a discrete time shift as described in (7.9).

$$y(k)Z^{-1} = y(k-1) \tag{7.9}$$

The expression can be described by (7.10).

$$\hat{y}(k+1) = f\left(x(k), x(k-1), \cdots, x(k-d_u), y(k), y(k-1), \cdots, y(k-d_y)\right) \tag{7.10}$$

The sizes of the delays have initially been determined by the same procedure as for the neurons.

7.3.2 Training Method

The NARX network is designed with a set of parameters, which are designed in an adaptation process based on information about the desired measurable inputs and outputs. This method was initially proposed by [Hecht-Nielsen, 1988] and the terminology used was to "*train the network*".

[R. Isermann, 2010] describes that the automated design process consists of the *training* stage previously mentioned and then a *generalisation* stage where the NARX network is used to simulate/predict new data, which has not necessarily been part of the training dataset. This is done to determine the performance of the NARX network for new and unknown data. Lastly, it should be noted that if the measurable input & output variables are not correlated in any logical manner, the NARX network cannot be expected to perform well.

There exists three main types of training methods:

Supervised Training

Supervised training or "training with a teacher", provides the ANN with input data and the correct corresponding output data. The input data is fed through the ANN and the outputs are compared to the correct outputs. If they agree, no changes are made. If they do not agree, the weights within the ANN are adjusted to reduce error on a given training set, ensuring a better likelihood for a correct answer for future similar input data. The weights are adjusted by the use of a gradient based optimisation algorithm, which will be explained later.

Unsupervised Training

Unsupervised training provides the ANN only with input data. The ANN has to self-organise the data, depending on the given structure of the input data. The structure of the input data would normally consist of redundant data or specific clusters. This is typically used to organise large amount of data, where a specific output is not known.

Reinforcement Training

Reinforcement training is a combination of the previous two types. The ANN is provided with the input data, but not the actual output. Instead the ANN is only told if the output is correct or not. If the output is wrong then the weights are adjusted. This is a trial and error approach, where the ANN trains until it reaches the actual output and is effective in learning tasks where the output is either a success or failure.

The chosen training method is **Supervised Training**, since this is generally used for prediction of time series and since the desired output is known.

The objective of training the ANN is to minimise the errors of outputs, hence increasing the accuracy of the derived network. To achieve this, a number of different training algorithms are available within the embedded MATLAB neural network toolbox.

Levenberg-Marquardt

Levenberg-Marquardt adaptively varies between the gradient descent update (generally effective far away from the solution point) and the Gauss-Newton update (generally effective near the solution point) [Arora, 2012]. The Levenberg-Marquardt Algorithm (LMA) gives fast convergence and is in general the most versatile algorithm.

Bayesian Regularisation

Bayesian regularisation is a statistical method suited well for small and/or noisy datasets due to its statistical properties. This may help reduce overfitting of the measured output. It updates weights and biases by LMA.

Scaled Conjugate Gradient

Scaled conjugate gradient is typically applied to relatively simple problems, due to its increasing convergence period as complexity increases.

Based on these descriptions and recommendations from MATLAB, and that LMA has been proven to be fast and stable [Arora, 2012], it is concluded that the LMA is best suited for this problem. When noisy measurements are included it can be considered to apply the Bayesian Regularisation as a second choice.

The Levenberg-Marquardt Algorithm

The following section is made with inspiration from [M. T. Hagan, 1994] and to understand the full process of the LMA, knowledge about the following algorithms is required: Steepest Decent Algorithm (SDA), Newton's Method, Gauss-Newton's Algorithm (GNA). For brevities sake these will not be explained here, but can also be found in [M. T. Hagan, 1994].

LMA is a combination of SDA and GNA, which results in an inheritance of the convergence speed from the GNA, but with the stability from the SDA. However, the LMA tends to converge a bit slower than the GNA, but still much faster then the SDA.
The idea behind the LMA is to make a combination of the two algorithms within the training process. If the algorithm reaches an area with a non-smooth curvature it will switch to the SDA until a more smooth curvature is reached. At this area a proper quadratic approximation can be made, and the LMA switches to the GNA speeding up the convergence.

The advantage of the GNA compared to the standard Newton method, is that the GNA does not require the calculation of second order derivatives in order to compute the Hessian matrix. Instead, the GNA introduces the Jacobian matrix, which can be used to approximate the Hessian matrix. This is beneficial if the second derivatives are computationally difficult to obtain.

To minimise the ANN model error, a summed squared error cost function (7.11) is introduced. The cost function is updated once all the available data has been applied, and the result of this is used to update the parameters.

$$SSE\left(\xi, \mathbf{L}_w, \boldsymbol{b}\right) = \frac{1}{2} \sum_{p=1}^{P} \sum_{m=1}^{M} \boldsymbol{e}_{p,m}^2 \left(\xi, \mathbf{L}_w, \boldsymbol{b}\right) \tag{7.11}$$

where

ξ	is the regressor vector.
$\boldsymbol{b}$	is the bias vectors.
$\mathbf{L}_w$	is the weight matrices.
$\boldsymbol{e}_{p,m}$	is the error at output m when pattern p is applied.
p	is the index of patterns, from pattern 1 to total number of patterns P.
m	is the index of the outputs, from output 1 to total number of outputs M.
SSE	is the sum of squared errors.

The error is described by (7.12).

$$\boldsymbol{e}_{p,m} = \boldsymbol{y}_{p,m} - \hat{\boldsymbol{y}}_{p,m} \tag{7.12}$$

where

$\boldsymbol{y}$	is the desired output vector.	[-]
$\hat{\boldsymbol{y}}$	is the actual output vector.	[-]

Depending on this error the weights are adjusted by error back-propagation described in the following.

Hessian Approximation & Adapting the Weights

The LMA is designed like the quasi-Newton methods where the algorithm is approaching second-order training speed, but without having to calculate the Hessian matrix of the ANN function. The cost function has the form of a sum of squares given by (7.11) and the approximated Hessian is given by (7.13) [M. T. Hagan, 1994].

$$\mathbf{H} \approx \mathbf{J}^T \mathbf{J} + \mu \mathbf{I} \tag{7.13}$$

where μ is the combination coefficient and is always positive, $\mathbf{I}$ is the identity matrix. $\mathbf{J}$ is the Jacobian matrix which holds the first derivatives of the ANN errors, with respect to the weights, as shown in (7.14).

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \cdots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \cdots & \frac{\partial e_{1,2}}{\partial w_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial e_{1,M}}{\partial w_1} & \frac{\partial e_{1,M}}{\partial w_2} & \cdots & \frac{\partial e_{1,M}}{\partial w_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial e_{P,1}}{\partial w_1} & \frac{\partial e_{P,1}}{\partial w_2} & \cdots & \frac{\partial e_{P,1}}{\partial w_N} \\ \frac{\partial e_{P,2}}{\partial w_1} & \frac{\partial e_{P,2}}{\partial w_2} & \cdots & \frac{\partial e_{P,2}}{\partial w_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial e_{P,M}}{\partial w_1} & \frac{\partial e_{P,M}}{\partial w_2} & \cdots & \frac{\partial e_{P,M}}{\partial w_N} \end{bmatrix} \tag{7.14}$$

The number of weights (N) are determined by the network design, and when introducing several patterns and outputs the Jacobian matrix grows by $P \times M \times N$. For large sized training patterns this can give memory issues, but due to the amount of training sets in this work and the relatively restricted size of the ANNs, this is not considered an issue.
Note that the Jacobian includes differentiation of each computed error, hence the entire dataset is applied for each update iteration. This is referred to as *Batch* training.
Furthermore, the gradient can be calculated as (7.15).

$$\boldsymbol{g} = \mathbf{J}\boldsymbol{e} \tag{7.15}$$

Where the error vector is given by (7.16).

$$\boldsymbol{e} = \begin{bmatrix} e_{1,1} & e_{1,2} & \dots & e_{1,M} & \dots & e_{P,1} & e_{P,2} & \dots & e_{P,M} \end{bmatrix}^T \tag{7.16}$$

The Jacobian matrix is calculated by utilising SDA, which is also known as the error back-propagation algorithm. However, it should be noted that in the LMA, the back-propagation process has to be repeated for all the different outputs separately.

The LMA can be described as the Newton-like update equation shown in (7.17), [M. T. Hagan, 1994].

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \left(\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I}\right)^{-1} \mathbf{J}_k \boldsymbol{e}_k \tag{7.17}$$

Note, that when μ in (7.17) is approaching zero, the LMA is approaching Newton's Method, and when μ is large the LMA approaches the SDA with a small step size.Since Newton's method is fast and accurate near a solution point, the goal is to shift to GNA as fast as possible. This means that μ must be decreased after a successful step, and only increase if the cost function will increase in the next step. By doing so, the cost function is always reduced at each iteration of the LMA.
The procedure of the algorithm is illustrated in the following.

Illustrative Example of an Iteration in LMA

The flow diagram of the algorithm is shown in fig. 7.8.

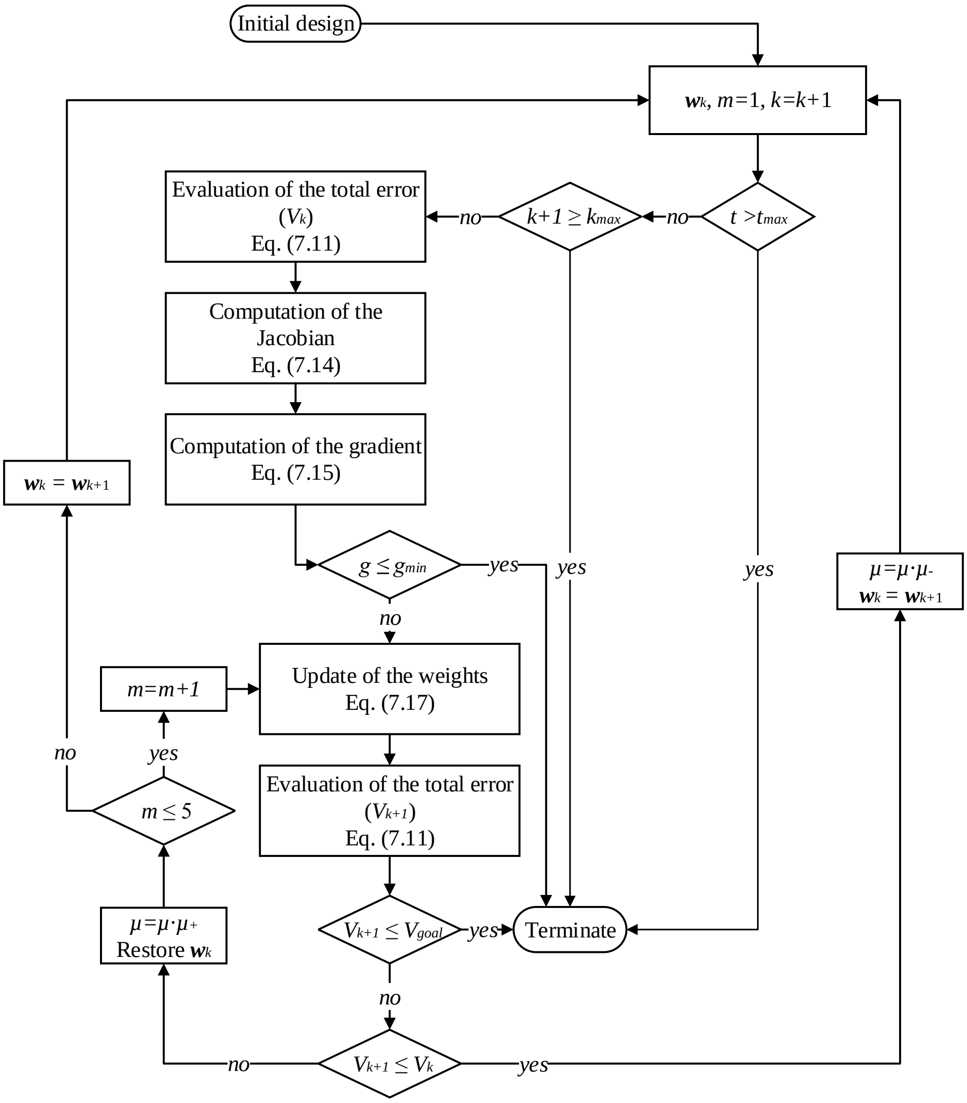

Initial design
$\mathbf{w}_k$, m=1, k=k+1
Evaluation of the total error (V_k) Eq. (7.11)
no
$k+1 \geq k_{max}$
no
$t > t_{max}$
Computation of the Jacobian Eq. (7.14)
Computation of the gradient Eq. (7.15)
$\mathbf{w}_k = \mathbf{w}_{k+1}$
$g \leq g_{min}$
yes
yes
yes
$\mu=\mu \cdot \mu_-$ $\mathbf{w}_k = \mathbf{w}_{k+1}$
no
m=m+1
Update of the weights Eq. (7.17)
no
yes
$m \leq 5$
Evaluation of the total error (V_{k+1}) Eq. (7.11)
$\mu=\mu \cdot \mu_+$ Restore $\mathbf{w}_k$
$V_{k+1} \leq V_{goal}$
yes
Terminate
no
no
$V_{k+1} \leq V_k$
yes

Figure 7.8: Flowchart of the training process of the Levenberg-Marquardt algortihm, with inspiration from [M. T. Hagan, 1994].

Where the parameters are defined as shown in tab. 7.1.

Max iterations	Initial μ	μ increase	μ decrease	Min gradient	Performance goal	Max time
k_{max}	μ_{ini} [-]	μ_+ [-]	μ_- [-]	g_{min} [-]	V_{goal} [-]	t_{max} [s]
10,000	0.001	10	0.1	1e-6	0	∞

Table 7.1: Values used in the LMA when training ANNs.

The values from tab. 7.1 can be designed to fit the application of the ANN, and thereby the algorithm can be terminated when it is not feasible to optimise any further.

The given values are an example of the NARX design where zero error is preferred. The iteration speed varies depending on ANN design and applied data, and 10,000 iterations has been chosen since the designs capable of completing this amount without reaching the gradient limit is heavily underfitting. The values in the ANN are normalised, hence a gradient below

1e-6 is not considered to be sufficient to continue the LMA for the purpose of this ANN. The maximum time allowed is set to infinity in the preliminary tests to avoid this being a factor on the achieved performance, which is why either gradient or iteration causes termination.
The benefit of this approach is to reveal the obtainable performance of the ANNs which is essential to do a comparison. The actual training period required can then be used to evaluate the effort connected with increased performance.

The process of the LMA shown in fig. 7.8 can be described by the following steps.

Step 1 Generation of an initial random set of weights.
Step 2 Reset m to default. Count iteration number, k and store the current design.
Step 3 The design is used to evaluate the model error from (7.11), the Jacobian from (7.14) and the gradient from (7.15).
Step 4 The computations from **Step 3** are applied to (7.17) which adjusts the weights.
Step 5 The new weights are used to evaluate the total error from (7.11).
Step 6 If a stopping criterion is reached the process terminates.
Step 7.a If the current total error is increased as a result of the update, then reset the weight vector to the previous values and change the combination coefficient (μ) by μ_+. When m is below 6 then go to **Step 4** and update the design again. When m is above 5 go to **Step 2**.
Step 7.b If the current total error is decreased as a result of the update, then accept the step and change the combination coefficient by μ_-. Then go to **Step 2**.

Required Training Period

The initial training period for the NARX ANN was: ≈39 min (update of 270 weights and 5 biases). The LMA was terminated by reaching the minimum allowed gradient.

7.3.3 Preliminary Design of NARX for Residual Generation

Based on the previous described theory and decisions, the developed NARX ANN design is summarised in the list below.

(e.1) Choose suitable design for the ANN (NARX)

- **(e.1.1)** Define number of neurons in the hidden layer: $n = 5$
- **(e.1.2)** Define number of input delays: $d = 8$
- **(e.1.3)** Define number of output delays: $d = 8$
- **(e.1.4)** Training method: Levenberg-Marquardt with back-propagation

(e.2) Define relevant measurable input variables

- **(e.2.1)** Valve position reference ($x_{P,ref}$)

(e.3) Define relevant measurable output variables

- **(e.3.1)** Valve position (x_V)
- **(e.3.2)** Lumped pressures (p_A, p_B)
- **(e.3.3)** Piston position and velocity ($x_P, \dot{x}_P$)

(e.4) Normalisation of the variables

- **(e.4.1)** Pressures: $p_{A,max} = p_{B,max} = 180$ bar
- **(e.4.2)** Actuator position: $x_{P,max} = 0.5$ m
- **(e.4.3)** Actuator velocity: $v_{P,max} = 0.17$ m/s
- **(e.4.4)** Valve position: $x_{V,max} = 1$

7.4 Architecture of the Feedforward ANN

The structure of the applied feedforward (FF) networks, specifically Feedforward ANN (FFANN) and Focused Time-Delay ANN (FTDANN) are identical with the exception of delay functions being included in the FTDANN, where previous values of the input makes the network dynamic, hence better possibility of capturing the transient relationships between inputs and outputs. The architecture with delays is illustrated in fig. 7.9.

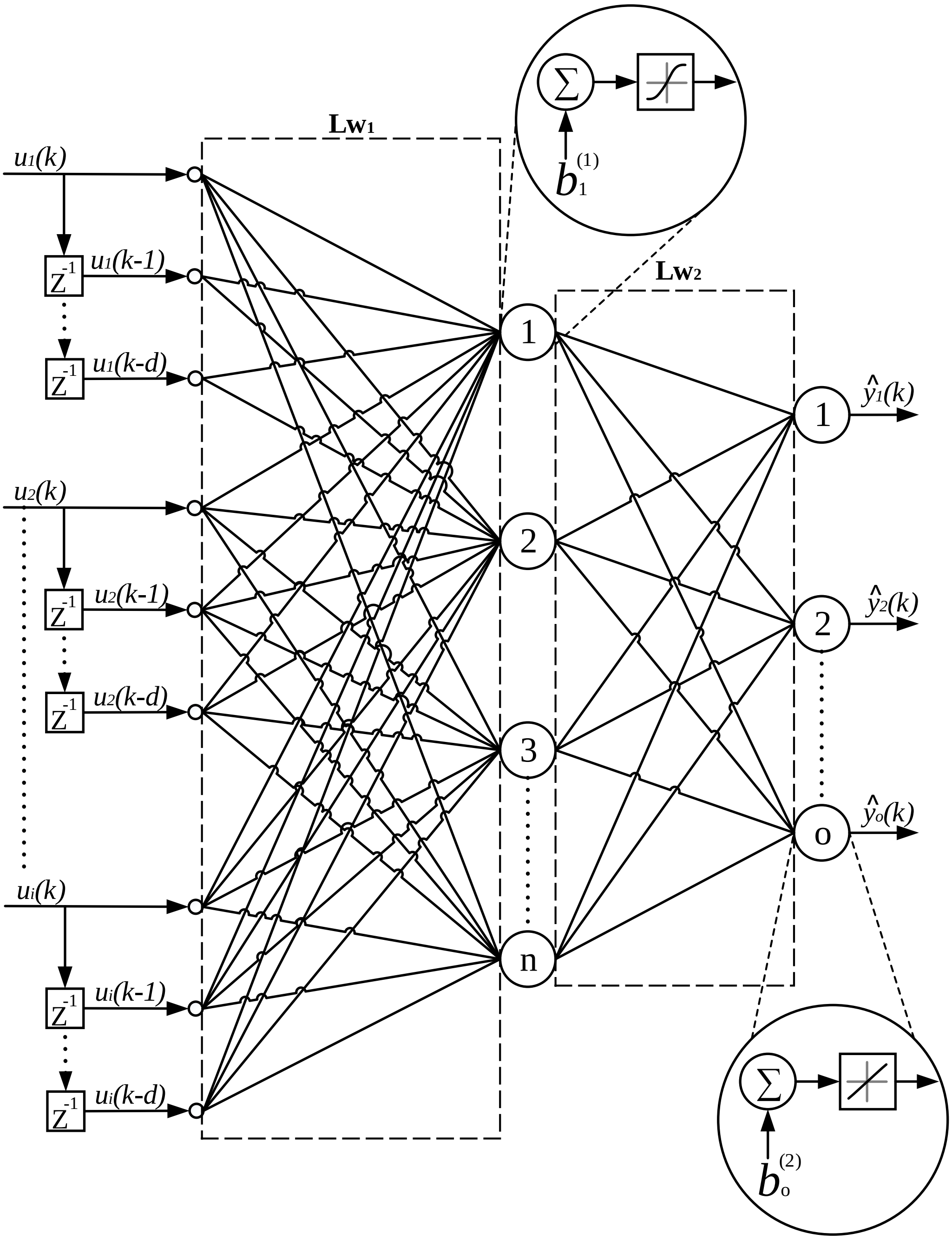


Figure 7.9: Structure of feedforward ANN with number of inputs (i), number of delays (d), number of neurons (n), layer weights (**Lw**), biases (b) and number of outputs (o). It should be noted, that if d = 0 the structure represents a FFANN.

This architecture has been chosen due to its simplicity, and prior proven performance in estimation of parameters in a hydraulic servo system [C. S. Byington, 2004], [W. J. Crowther, 1998] & [A. El-Betar, 2006]. Furthermore, the universal approximation theorem given by Theorem 1 and [MathWorks, 2016a] states that a FF network with one hidden layer and one output layer, with enough neurons in the hidden layer, will be able to fit any finite input-output mapping problem.

7.4.1 Input- & Output- Variables

The selection of input- & output- variables is equal to the one presented in sec. 7.3.1. This is done with the purpose of making it possible to compare if there is a significant difference between forwarding the actual residual of the states in contrast to simply forwarding the measurements from the system directly to the feedforward ANNs.
The training data has been collected through simulation by means of the simulated fault approach described in chap. 6.

The output variables are chosen to directly reflect the faults desired predicted. For the preliminary test, the three leakage flows are therefore the outputs.

The input regressor vector for the network is described by (7.18).

$$\begin{aligned}\boldsymbol{u}(k) &= \begin{bmatrix} u_1(k) & u_2(k) & \cdots & u_i(k) \end{bmatrix}^T \\ &= \begin{bmatrix} p_A(k) & p_B(k) & x_P(k) & \dot{x}_P(k) & x_V \end{bmatrix}^T \end{aligned} \tag{7.18}$$

Where i denotes the number of inputs and k is the current time increment.
The output regressor vector is defined to include the parameters desired estimated. This is given by (7.20).

$$\hat{\boldsymbol{y}}(k) = \begin{bmatrix} \hat{y}_1(k) & \hat{y}_2(k) & \cdots & \hat{y}_i(k) \end{bmatrix}^T \tag{7.19}$$

$$= \begin{bmatrix} \hat{Q}_{int,L}(k) & \hat{Q}_{A,ext,L}(k) & \hat{Q}_{B,ext,L}(k) \end{bmatrix}^T \tag{7.20}$$

The total network function for the FFANN and FTDANN is described by (7.21), which represents the structure from fig. 7.9.

$$\hat{\boldsymbol{y}}(k) = \mathbf{L}_{w2} \tanh\left(\mathbf{L}_{w1}\xi(k) + \boldsymbol{b}^{(1)}\right) + \boldsymbol{b}^{(2)} \tag{7.21}$$

Where the weight matrices are defined by (7.22). Similarly the biases are given by (7.23).

$$\mathbf{L}_{w1} = \begin{bmatrix} w_{1,1}^{(1)} & \cdots & w_{1,(d+1)i}^{(1)} \\ \vdots & \ddots & \vdots \\ w_{n,1}^{(1)} & \cdots & w_{n,(d+1)i}^{(1)} \end{bmatrix} \qquad \mathbf{L}_{w2} = \begin{bmatrix} w_{1,1}^{(2)} & \cdots & w_{1,n}^{(2)} \\ \vdots & \ddots & \vdots \\ w_{o,1}^{(2)} & \cdots & w_{o,n}^{(2)} \end{bmatrix} \tag{7.22}$$

$$\boldsymbol{b}^{(1)} = \begin{pmatrix} b_1^{(1)} \\ \vdots \\ b_n^{(1)} \end{pmatrix} \qquad \boldsymbol{b}^{(2)} = \begin{pmatrix} b_1^{(2)} \\ \vdots \\ b_o^{(2)} \end{pmatrix} \tag{7.23}$$

where n denotes the number of neurons in the hidden layer, d denotes the input delays and o denotes the number of outputs. These weights and biases are adapting while the network is training with the back-propagation algorithm, which minimises the square of errors.

The regressor $\xi(k)$ is given by (7.24).

$$\xi(k) = \begin{pmatrix} \boldsymbol{u}(k) \\ \boldsymbol{u}(k-1) \\ \vdots \\ \boldsymbol{u}(k-d) \end{pmatrix} \tag{7.24}$$

When using the dimensions for the presented matrices and vectors, the output of (7.21) becomes a $o \times 1$ vector.

Number of Neurons

The initial design of the FFANN has $n = 50$ neurons and the FTDANN has $n = 12$ neurons. The different designs are based on an manual iterative process where a trade-off between increased performance versus training period is considered. Furthermore, this contributed knowledge about the capabilities of the chosen ANNs.
The FTDANN has significantly fewer neurons since it utilises $(d+1) \cdot i$ input values, i.e increasing the number of weights.

Number of Delays

The FTDANN is designed with a discrete time delay of $d = 8$ samples for each input state, where the sample rate is 1 kHz. This is therefore only a period of 8 ms, however gives useful information about the change of the states.
This is based on the trade-off between increased performance versus training period found through manual tuning.

Training Method & Required Training Period

The training algorithm is chosen based on the arguments presented in sec. 7.3.

The training period for the FFANN was: ≈6 min (update of 250 weights and 50 biases).
The training period for the FTDNN was: ≈28 min (update of 540 weights and 12 biases).

7.4.2 Preliminary Design of Data Driven ANN for FDD

Based on the previous described theory and decisions, the design procedure is summed in the list below.

(f.1) Choose suitable design for the ANN (FFANN or FTDANN)

- **(f.1.1)** Define number of neurons in the hidden layer: $n = 50$(FFANN) $n = 12$(FTDANN)
- **(f.1.2)** Define number of delays: $d = 8$(FTDANN)
- **(f.1.3)** Training method: Levenberg-Marquardt back-propagation

(f.2) Define relevant measurable input variables

- **(f.2.1)** Valve position (x_V)
- **(f.2.2)** Lumped pressures (p_A, p_B)
- **(f.2.3)** Piston position and velocity $(x_P, \dot{x}_P)$

(f.3) Define relevant measurable output variables

- **(f.3.1)** Internal leakage flow $(Q_{L,int})$

(f.3.2) External leakage flows ($Q_{L,ext,A}$, $Q_{L,ext,B}$)

(f.4) Normalisation of the variables

(f.4.1) Pressures: $p_{A,max} = p_{B,max} = 180$ bar
(f.4.2) Actuator position: $x_{P,max} = 0.5$ m
(f.4.3) Actuator velocity: $v_{P,max} = 0.17$ m/s
(f.4.4) Valve position: $x_{V,max} = 1$

7.5 Data Driven FDD Results

The preliminary designs of the data driven FDD with and without delays in the ANN are evaluated by simulation. This is done with the test sequence described in sec. 6.1 where a sine wave is used as position reference.

Performance Evaluation

To quantify the performance of each ANN the root mean square errors (RMS) of the estimates are evaluated by (7.25). This is a common approach to give a measure of model estimation accuracy and has furthermore been applied by [W. J. Crowther, 1998] for evaluation of ANNs used for leakage estimation.

$$\text{RMS} = \sqrt{\frac{\sum_{k=1}^{S}(\hat{y}_k - y_k)^2}{S}} \tag{7.25}$$

where

k	is the current discrete time increment.	[-]
S	is the number of samples.	[-]

In general the RMS value over time smoothens sudden peak differences between estimate and actual value. On the contrary a continuous error will accumulate over time, hence increasing the mean. This quality is considered beneficial when evaluating the leakage faults, since it is preferred to diagnose a fault over longer periods of time, and a continuous error estimate could result in misinterpretations, whereas a sudden peak would be filtered. The lower the RMS error the better.
The issue with this value is that it can be difficult to interpret if a given value is statistic significant and that the prediction can be trusted.
Nevertheless, RMS errors are suited for comparison purposes which is of interest in this project.

The three estimated parameters for the two FDDs and the actual simulated parameters are shown in fig. 7.10. During the start-up phase (the two first seconds) of the simulation, inaccurate results are expected since there may exist a difference in the initial conditions. To minimise this phenomenon the first two seconds have been omitted in some simulations.

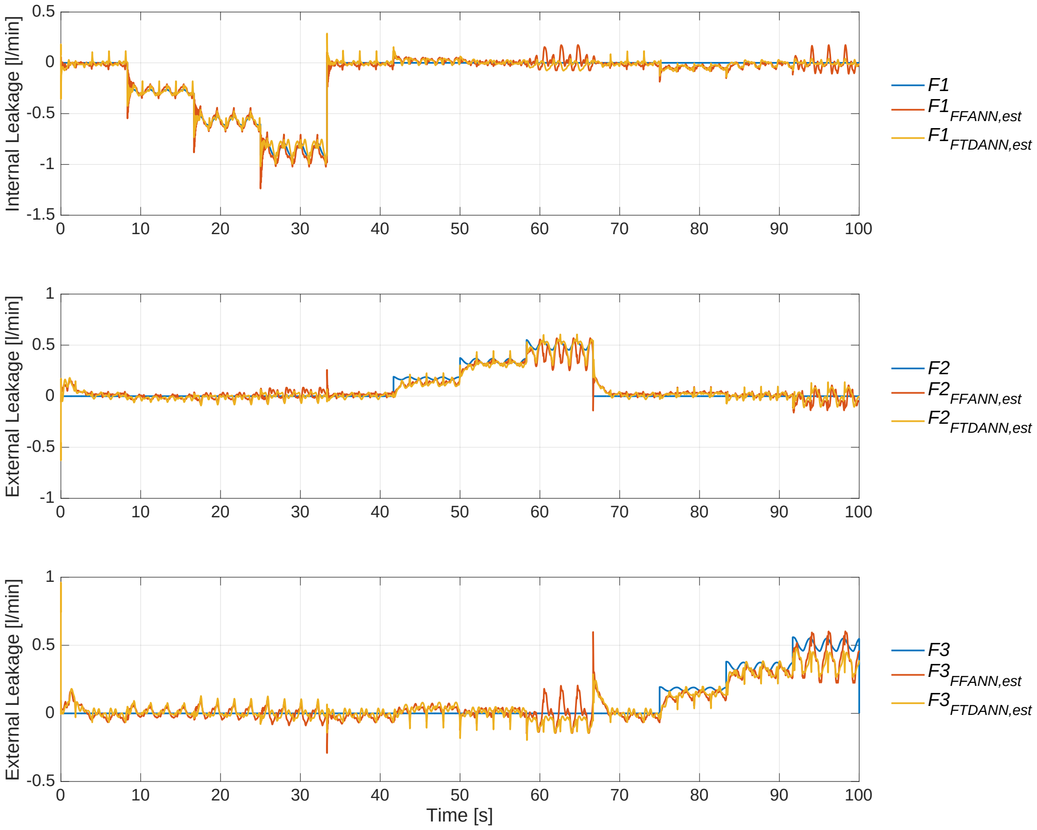


Figure 7.10: The simulated estimated leakage flows (denoted: $F1_{est} - F3_{est}$) by using FFANN & FTDANN with data driven FDD, compared with the actual simulated leakage flows.

In fig. 7.10 it is observed that the estimated flows have a tendency of oscillating near the correct value. Therefore, it is proposed to visualise the difference of performance clearer by removing some of the oscillations. This is done by introducing an exponential moving average (EMA), since a similar approach was done in [L. An, 2014b]. This weights the most recent values higher than previous i.e. has a faster response to changes while removing high frequent oscillations. The expression is given by (7.26).

$$\begin{aligned}\mathrm{EMA}(x(k)) &= \gamma x(k) + (1-\gamma)\mathrm{EMA}(x(k-1)) \\ &= \gamma x(k) + \gamma(1-\gamma)x(k-1) + (1-\gamma)^2\mathrm{EMA}(x(k-2)) \\ &= \gamma\left[x(k) + (1-\gamma)x(k-1) + (1-\gamma)^2 x(k-2) + \cdots + (1-\gamma)^{k-1}x(1)\right] + (1-\gamma)^k x(0) \\ \gamma &= \frac{2}{Period+1}\end{aligned} \tag{7.26}$$

where

EMA	is the value of the exponential moving average.	[-]
γ	is the exponential multiplier.	[-]
$Period$	is the period for the moving average, 1kHz· 2 s.	[-]
x	is the variable which is averaged.	[-]

From the difference equation in (7.26) it can be seen, that as time passes, the EMA becomes the weighted average of an increasing number of previous measurements $x(k-n)$, and the weight γ assigned to the past observations. This difference equation can in terms be seen as a discrete exponential function.

The EMA can be designed with appropriate time periods depending on the length and increments of the time series. For these particular datasets it is decided to accumulate the average over a period of 2 seconds and the results when applying this are depicted in fig. 7.11.

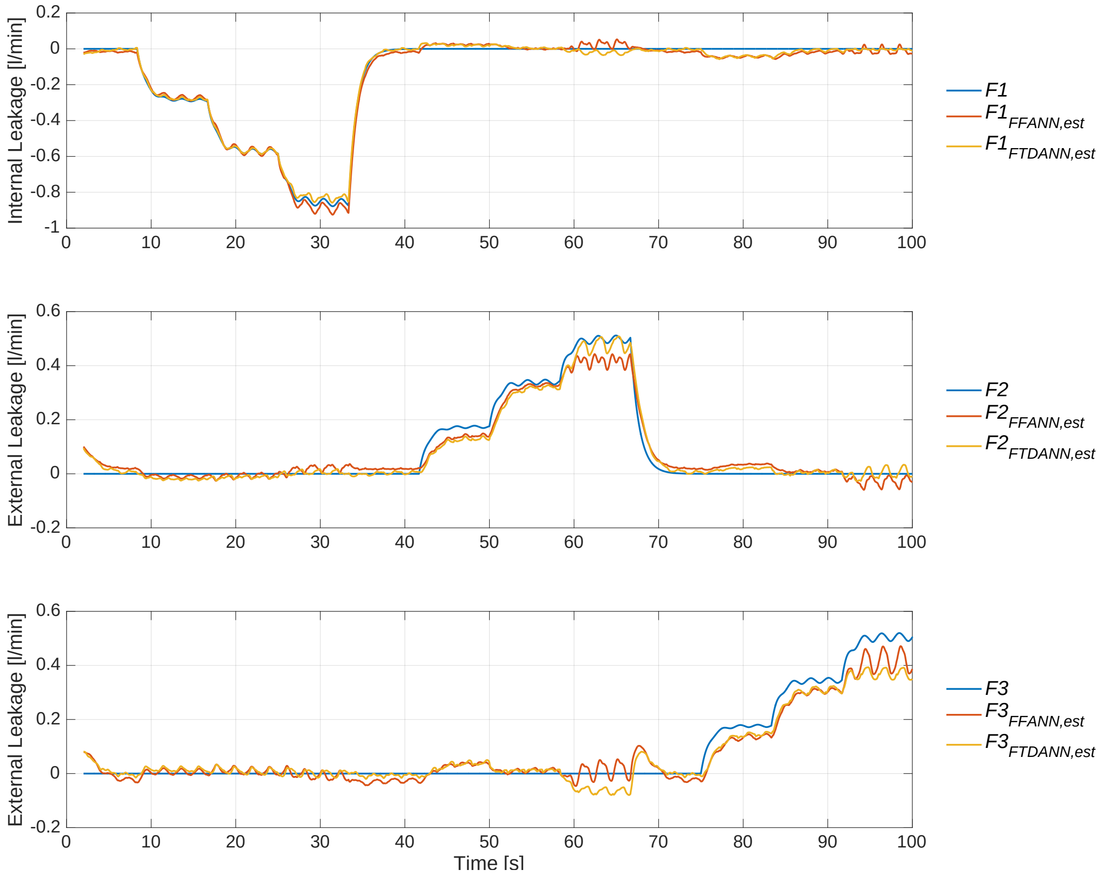


Figure 7.11: 2 second EMA of estimated leakage flows from data driven FFANN & FTDANN compared with 2 second EMA of actual simulated flow.

The RMS errors of FFANN & FTDANN are shown in fig. 7.12 and these yield that in 2 out of 3 faults the FTDANN has best performance and that $F3$ is the most difficult to estimate accurately for both ANNs. Based on the RMS errors it is decided to apply the FTDANN for further work.

In the following the performance of the model-based FDD scheme is investigated.

7.6 Model-Based FDD Results

To test the performance of the model-based FDD scheme, it is initially desired to investigate the performance of the estimator. This is done by using the afore mentioned ideal residuals (r_{ideal}) and the residuals generated by the NARX (r_{NARX}). Furthermore, the difference between actual output and estimated output (7.27) is graphically displayed to illustrate the estimation

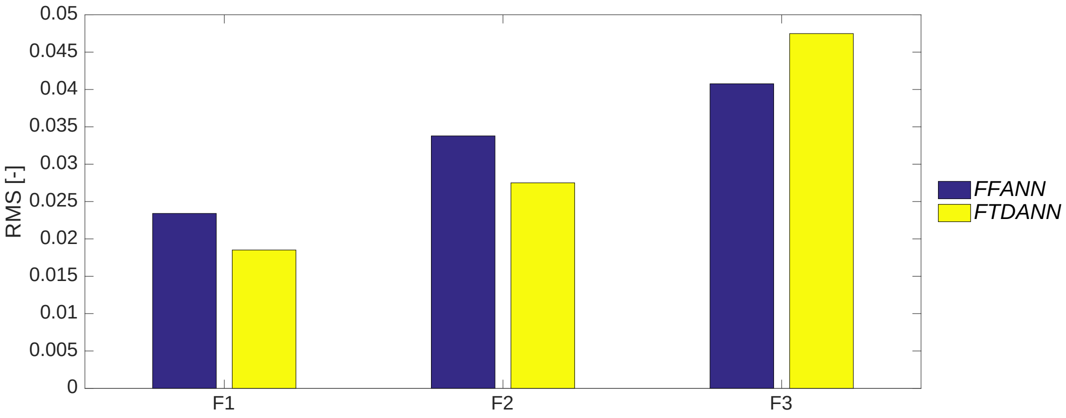


Figure 7.12: Root-mean-square error between estimated- and actual faults from results shown in fig. 7.11.

accuracy.

$$\Delta y = r_{ideal} - r_{NARX} \quad ; \quad r_{NARX} = y_f - \hat{y}_{NARX} \quad ; \quad r_{ideal} = y_f - y$$
$$\Delta y = y - \hat{y} \tag{7.27}$$

where

y	is the actual system output without a fault.	[-]
y_f	is the actual system output with a fault.	[-]
$\hat{y}_{NARX}$	is output estimated by the NARX ANN.	[-]

The purpose of the comparison is to obtain a measure for the designed ANN's capability of replicating the behaviour of the system. The residuals for pressures are shown in fig 7.13. The position in fig. 7.14 and velocity in 7.15. The valve position residual is shown in fig. 7.16. It should be noted that the two first seconds of each data series have been omitted, due to differences of initial conditions and since this part is not relevant for analysis purposes.
The residuals are generated with the fault sequence defined in sec. 6.1.
The interpretation of the Δy is simply that values equal to zero represent a perfect model, and that oscillation around zero reveals an accurate mean residual with some level of uncertainty.

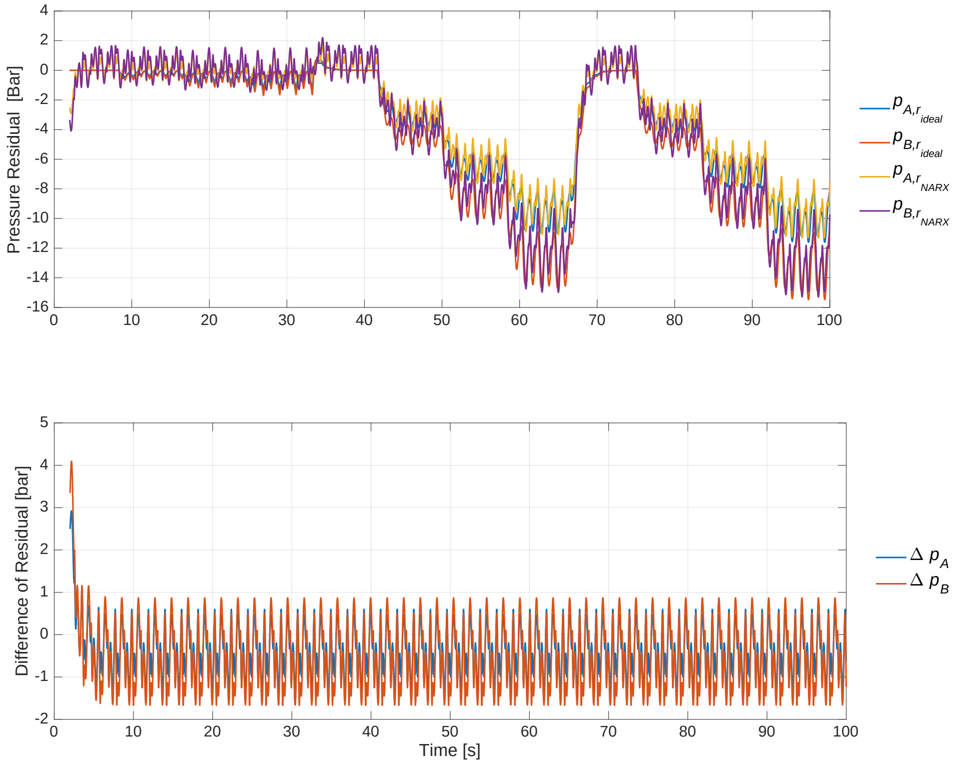


Figure 7.13: Simulated difference in pressure residuals generated with a sine input trajectory.

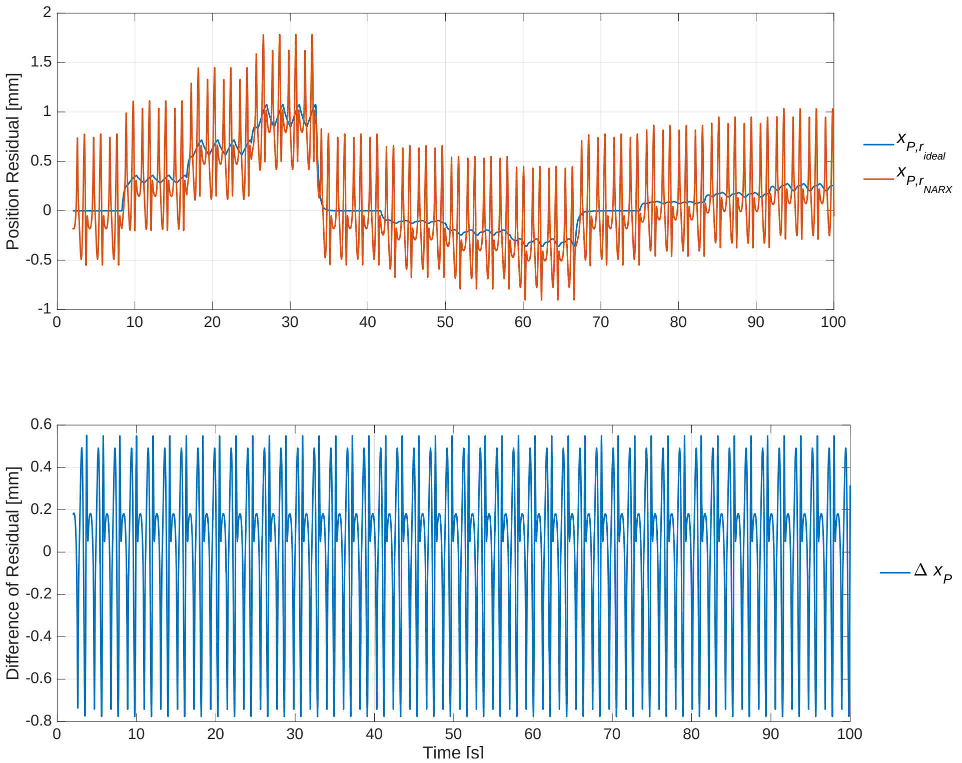


Figure 7.14: Simulated difference in position residuals generated with a sine input trajectory.

It is observed from the results in fig. 7.13, that the mean pressure residuals have an offset of ≈0.5 bar with oscillations of ≈ ±1.25 bar amplitude. It is expected that the FTDANN is capable

of recognising the overall tendency and to some extend filter the residual. The residual is therefore not filtered, with the purpose of minimising the design effort.
Furthermore, during the start-up phase (first five seconds) the pressures do not fit, which as mentioned earlier is caused by initial conditions on the pressures, which are not included in the NARX model. This is expected to affect the diagnosis during the start-up phase, but is not considered a major issue for the FDD performance.
The NARX does mimic the overall behaviour of the pressures in the actual system.

It is observed from the results in fig. 7.14, that the position estimate does include both offset of ≈-0.2 mm and oscillations of ≈ ±0.5 mm. However, as is the case with pressures, the NARX does mimic the overall behaviour of the position of actual system, even though it has never seen the fault sequence before.

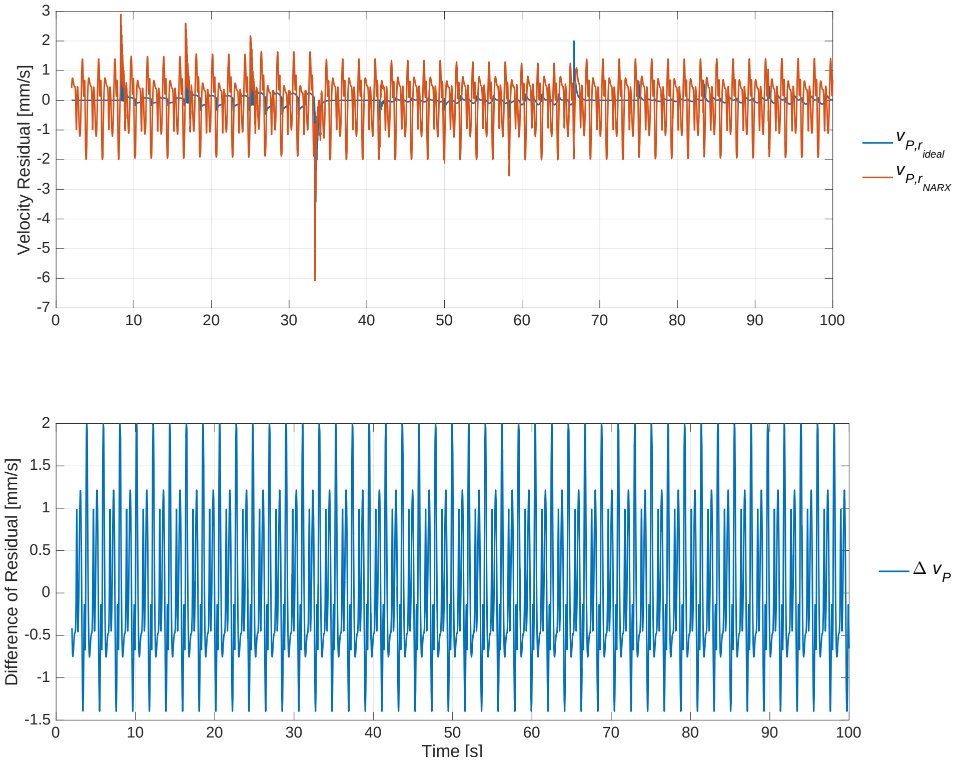


Figure 7.15: Simulated difference in velocity residuals generated with a sine input trajectory.

It is observed from the results in fig. 7.15, that the velocity estimate has an offset of ≈0.3 mm/s with an uncertainty of ≈ ±1.7 mm/s. Again the overall behaviour is sufficiently estimated.

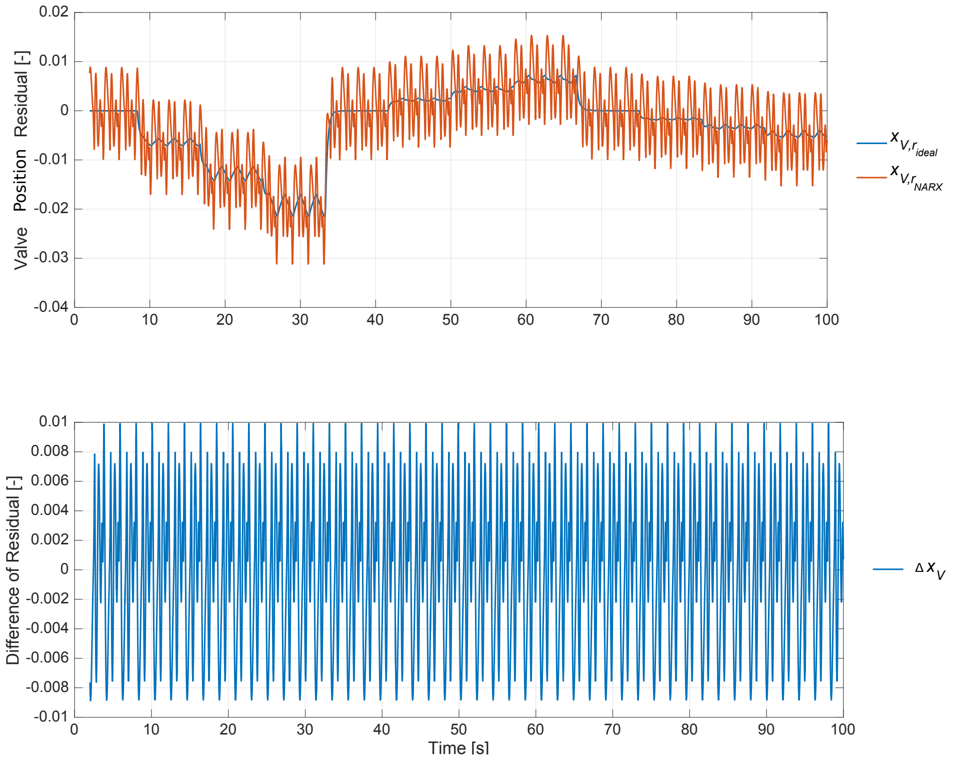


Figure 7.16: Simulated difference of valve position residuals generated with a sine input trajectory.

The valve position from fig. 7.16 has an estimation oscillation of ≈ ±0.01 and is a result of the performance of the Closed-Loop design. This reveals whether the actual system is compensating as usual or if adjustments are performed to track the position reference. The benefit of this residual is that it can be extracted directly from the control system and no special sensors are needed.

The general tendency from the above graphs are, that the residuals generated by NARX ANN is oscillating near the correct value. The magnitude of the oscillations is significant compared with the absolute residual when a fault is introduced. However, the NARX ANN is able to estimate the system states to some degree without having seen the faults beforehand, which is an indication that the NARX ANN behaves in a similar manner as the model. It has been decided to investigate FDD performance when applying the actual residuals and comparing the results with the NARX estimated residuals.

The preliminary design of the FTDANN is applied to diagnose the residuals. The approach for this was described in sec. 7.4.
The results presented in sec. 7.5 revealed some oscillations on the estimated parameters. These tendencies were also present in the results with the model-based scheme, which is why only the EMA results will be presented. The obtained results with the generated residuals can be seen in fig. 7.17.

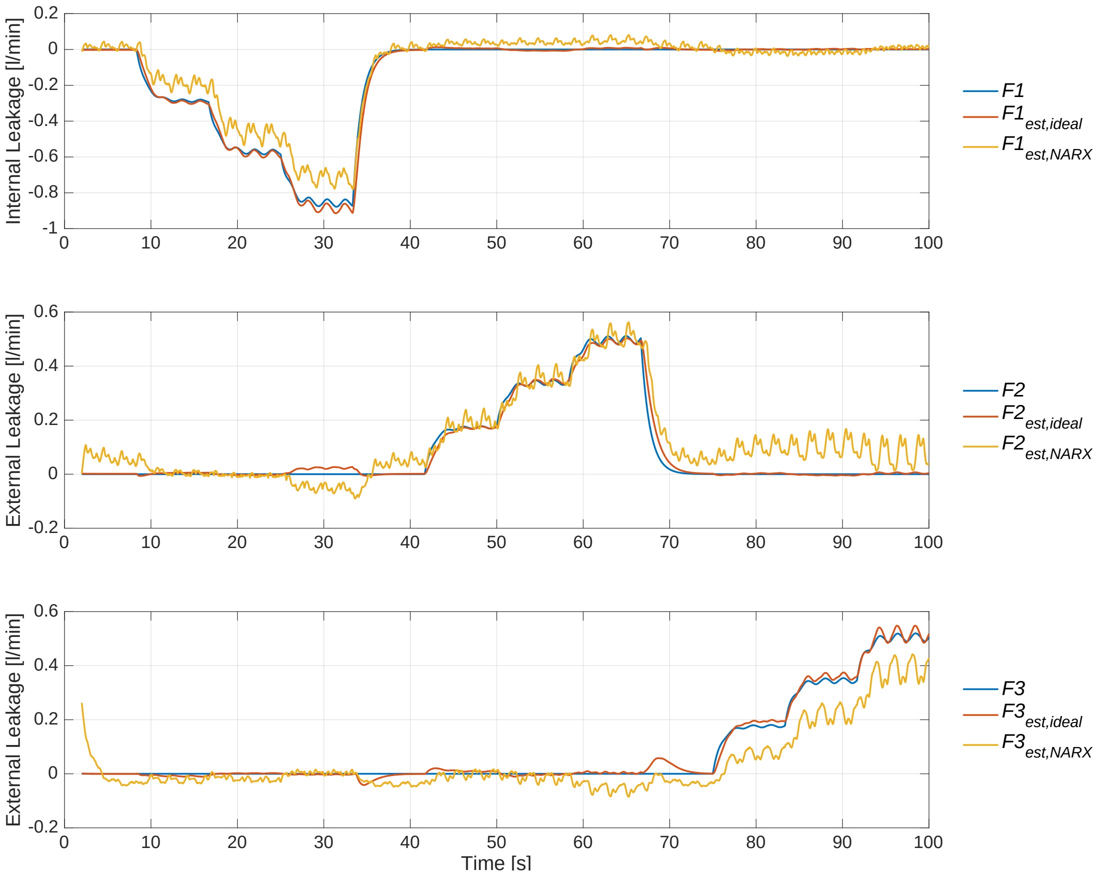


Figure 7.17: Two second EMA of estimated leakage flows ($F1_{est} - F3_{est}$) with either ideal or NARX residuals ($\boldsymbol{r}_{ideal}, \boldsymbol{r}_{NARX}$) applied to the FTDANN, compared with two second EMA of actual simulated leakage flows.

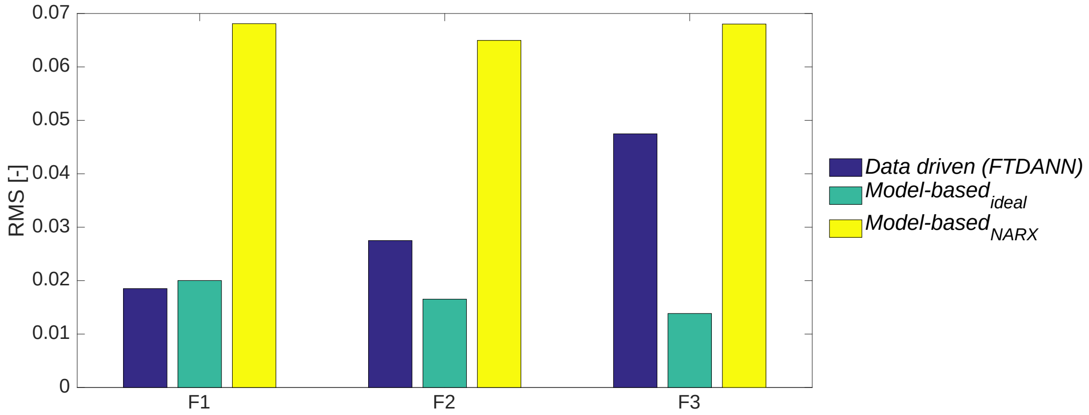


Figure 7.18: Root mean square error between estimated- and actual faults from results shown in fig. 7.17. For ease of comparison, the RMS errors from the data driven diagnosis shown in fig. 7.11 are included.

The RMS errors from fig. 7.18 show that the ideal residuals give superior results compared with application of NARX residuals or data-driven FDD. Furthermore, the FTDANN is better than the NARX model-based scheme, i.e. the estimator must be enhanced in order for this scheme to be feasible.

7.7 Part Conclusion

Two FDD schemes have been designed and tested through simulation. The theoretical results yielded promising potential for both FDD designs, but only one is selected for further work.

The model-based scheme with utilisation of the exact model showed superior results, and although it is not expected that a model with such accuracy can be realised in reality, the model-based FDD is considered to have potential. With the main reason being that the algorithm is "helped" to understand how an abnormality looks like via modelled residuals. Instead of the data driven approach, where raw features are fed to the ANN and is expected to somehow give a valid diagnosis.

Furthermore, it has been proven in literature that model-based FDD schemes include some level of robustness towards measurement noise and model uncertainties [Ehsan Sobhani-Tehrani, 2009].

The data driven scheme did yield promising estimates of the leakage and due to its simplicity, the design of such a FDD scheme is straightforward. Similar data driven schemes have been tested in [T. T. Le, 1997], [W. J. Crowther, 1998] & [A. El-Betar, 2006], which is why it is considered less interesting from a research point of view, compared with the model-based scheme. Therefore, the model-based scheme is chosen for further development.

Finally, simulation showed that the FTDANN slightly outperforms the FFANN when estimating leakage flows in a hydraulic servo system and the FTDANN is therefore applied in the scheme.

CHAPTER 8

Expansion of the Novel Fault Detection and Diagnosis Scheme

Contents

Based on the findings from chap. 7 a Fault Detection and Diagnosis (FDD) scheme has been selected, and in the following chapter this will be expanded to include the remaining faults and the amount of training data will furthermore be increased.
The data structures for training of the nonlinear autoregressive exogenous (NARX) ANN and the focused time-delay network (FTDANN) will be illustrated and elaborated. This structure is based on the work presented in [A. El-Betar, 2006] combined with information from [M. H. Beale, 2016] & [H. Su, 1992].

The objective is to investigate under what conditions the developed scheme is valid. Therefore the fault diagnosis estimated by the FTDANN and the residuals generated by the NARX ANN will be investigated in different scenarios which will be described in this chapter.

This is an investigation of the theoretical performance and will initially be evaluated without any measurement noise or uncertainty. However, a Gaussian distributed white noise on the measurements will later be applied to evaluate the ANNs robustness towards measurement noise.

8.1 Analysis of the Additional Faults

The faults of interest are listed again for convenience.

$F0$ is no fault/a healthy system.
$F1$ is internal rupture/leakage between volume A & B.
$F2$ is external rupture/leakage from volume A.
$F3$ is external rupture/leakage from volume B.
$F4$ is pressure transducer failure.
$F5$ is a stuck servo valve.

The impact caused by $F1$-$F3$ was described in chap 6, which is why only $F4$ & $F5$ are analysed in the following.

Pressure Transducer Failure

$F4$ is emulated by either receiving maximum or minimum measurable output from the transducer. This is based on the normal occurrence of failure in such a component [Pedersen, 2016].

Therefore, the pressure output on the transducer connected with volume A will be constantly either 1 or 250 bar, as shown in (8.1).

$$p_A = \begin{cases} p_A & \text{for } F4 = 0 \\ 250 \vee 1 \text{ bar} & \text{for } F4 = 1 \end{cases} \tag{8.1}$$

$F4$ is non-dimensional and binary, with 1 resulting in an erroneous transducer signal. The estimation of this failure ($\hat{F}4$) will therefore be trained to output either 0 or 1.

Valve Failure

$F5$ is emulated by disconnecting the control signal to the valve at some given point in time. It is then assumed that the valve spool will be affected by the spring and thereby return to neutral position. This is described by (8.2).

$$u_V = \begin{cases} u_V & \text{for } F5 = 0 \\ 0 & \text{for } F5 = 1 \end{cases} \tag{8.2}$$

The actual valve position and valve reference are available on the test rig, and the fault is expected detectable only by these variables.

$F5$ is non-dimensional and binary, with 1 indicating a stuck spool. $\hat{F}5$ is the notation for the estimate.

The simulation of the fault indices can be observed on fig. 8.1.

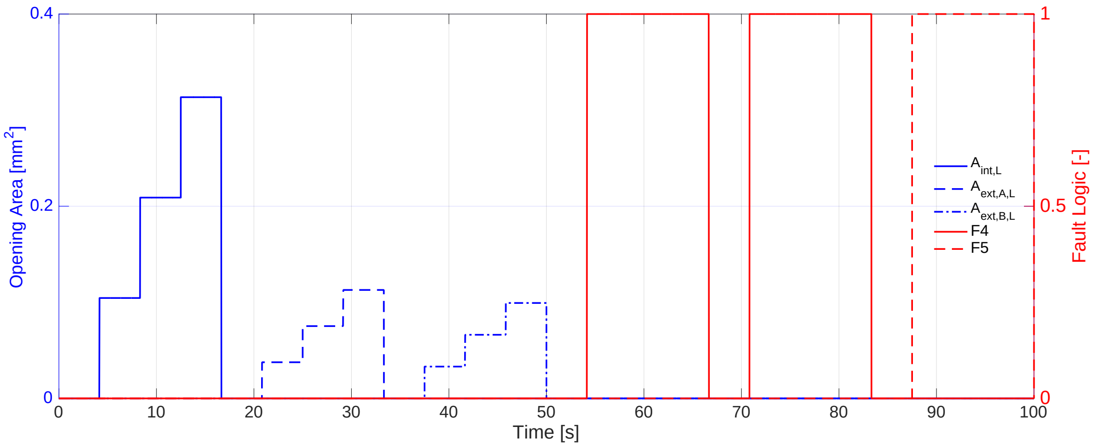


Figure 8.1: The fault sequence used for evaluation of the FDD. Note the faults *F*4 & *F*5 are non-dimensional and furthermore *F*4 is applied twice; one for both max and min pressure fault signal.

Through simulations it was found, that to detect all faults in one scheme, it is necessary to utilise two ANNs for classification of the faults. Therefore, *F*5 is diagnosed with data driven FDD and the remaining faults are diagnosed with a model-based FDD. This expanded structure of the FDD scheme is illustrated in fig. 8.2.

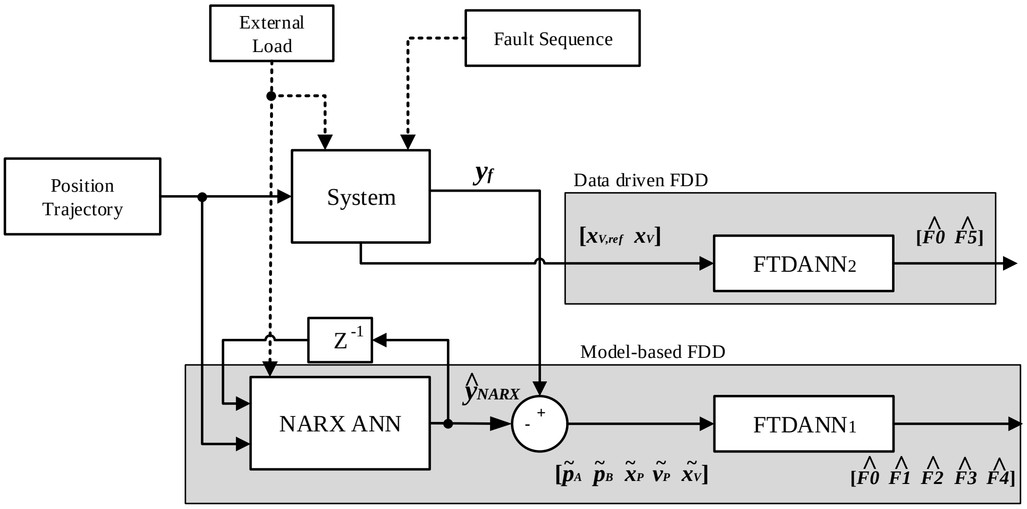


Figure 8.2: The FDD scheme where estimated values are denoted with a hat, i.e. $\hat{y}$ and the difference between estimate and actual values are denoted with a tilde, i.e. $\tilde{p}_A$.

The external load and position trajectories shown in fig. 8.2 will be explained later in this chapter.

It is observed from the figure, that the data driven method is applied for FTDANN$_2$ to detect the valve failure by monitoring the valve position and the referenced position. This is chosen since its simple and sufficient, but also because it is not effective when included in FTDANN$_1$. The remaining faults are diagnosed by FTDANN$_1$ through the model-based scheme.

This topology introduces an issue when an unknown fault occurs. Therefore, it was initially proposed to train FTDANN$_1$ to understand that the occurrence of *F*5 should not cause any

other faults. The reason for doing so, is that the purpose of applying ANNs is to avoid human interference where a list of rules must be designed, i.e. the objective is to let the scheme learn over time if the decision is correct or not. However, this did not give the desired results, which may be caused by the relatively low complexity of the network (only one hidden layer). This issue might be addressed by increasing the amount of hidden layers, which in the literature, e.g. [X. Glorot, 2010] proves enhanced learning capabilities (deep-learning). This will be tested later in this chapter.
Alternatively a manual rule can be implemented in the algorithm in order to make the ANNs communicate. This is defined by the sentence shown in algorithm 1, where the EMA refers to the previously applied Exponential Moving Average.

```
if $0.99 \leq \mathrm{EMA2}(\mathrm{FTDANN}_2) \quad \wedge \quad \mathrm{EMA2}(\mathrm{FTDANN}_2) \leq 1.01$ then
    $\mathrm{EMA2}(\mathrm{FTDANN}_1) = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}$
end
```

Algorithm 1: Manual overwrite of FTDANN_1 output.

This will enable communication between FTDANN_1 & FTDANN_2. However, the proposed manipulation will become problematic if the output of FTDANN_2 is between the limits while another fault is active. This gives a risk of inaccurate fault estimation, but the advantage of algorithm 1 is considered greater than the disadvantage since the output of FTDANN_2 has proved to be reliable.

8.2 Training of the FDD Scheme

In the following section it will be explained how the scheme has been trained and how the design parameters of the ANNs have been optimised. The structure is based on the training approach for a feedforward ANN from [A. El-Betar, 2006] and for a recurrent ANN in [H. Su, 1992].

8.2.1 ANN Training Data Structures

In this section, the design of the data structures applied for training will be addressed. This structure can affect both the performance and the training period of an ANN. Therefore, it is relevant to investigate how this may be done most efficiently.

The data structures of interest are those which occur *concurrently* (in a non specific time sequence), and those which occur *sequentially* (in a specific time sequence).
In the concurrent structure, the succession of each vector is of no importance. Therefore the data can be utilised simultaneously.
The sequential data structure is time-dependent and the succession is therefore important.

Both data structures are beneficial when training a dynamic ANN, i.e. network with either feedbacks or delayed data. The utilisation of concurrent data will make it possible to train a given ANN with all time series in parallel.

The reasoning for this structure is that the ANN can be designed to respond differently to each of the input vectors, as if they were applied to separate parallel ANNs. This feature gives the opportunity to create several different input vectors, where each vector would hold information about a healthy or faulty system, in a variety of different situations.

The approach for implementing the presented matrices into MATLAB, is to create one partly concurrent and partly sequential matrix for the system input sequence, and a similar one for the system output data.

In (8.3), (8.4), (8.5), (8.6), (8.7) & (8.8) the matrices applied when training the ANNs are described. Each structure in the matrices are denoted by $\{x\}$, these are a sequential representation of the measurable data.

$$\mathbf{U}_{\text{NARX}} = \begin{bmatrix} \{x_{P,ref}\}_1 & \{x_{P,ref}\}_2 & \cdots & \{x_{P,ref}\}_m \end{bmatrix} \tag{8.3}$$

$$\mathbf{Y}_{\text{NARX}} = \begin{bmatrix} \{p_A\}_1 & \{p_A\}_2 & \cdots & \{p_A\}_m \\ \{p_B\}_1 & \{p_B\}_2 & \cdots & \{p_B\}_m \\ \{x_P\}_1 & \{x_P\}_2 & \cdots & \{x_P\}_m \\ \{\dot{x}_P\}_1 & \{\dot{x}_P\}_2 & \cdots & \{\dot{x}_P\}_m \\ \{x_{V,ref}\}_1 & \{x_V\}_2 & \cdots & \{x_V\}_m \end{bmatrix} \tag{8.4}$$

$$\mathbf{U}_{\text{FTDANN}_1} = \begin{bmatrix} \{\tilde{p}_A\}_{1,1} & \{\tilde{p}_A\}_{2,1} & \cdots & \{\tilde{p}_A\}_{m,1} & \{\tilde{p}_A\}_{1,2} & \{\tilde{p}_A\}_{2,2} & \cdots & \{\tilde{p}_A\}_{m,\varrho_1} \\ \{\tilde{p}_B\}_{1,1} & \{\tilde{p}_B\}_{2,1} & \cdots & \{\tilde{p}_B\}_{m,1} & \{\tilde{p}_B\}_{1,2} & \{\tilde{p}_B\}_{2,2} & \cdots & \{\tilde{p}_B\}_{m,\varrho_1} \\ \{\tilde{x}_P\}_{1,1} & \{\tilde{x}_P\}_{2,1} & \cdots & \{\tilde{x}_P\}_{m,1} & \{\tilde{x}_P\}_{1,2} & \{\tilde{x}_P\}_{2,2} & \cdots & \{\tilde{x}_P\}_{m,\varrho_1} \\ \{\tilde{\dot{x}}_P\}_{1,1} & \{\tilde{\dot{x}}_P\}_{2,1} & \cdots & \{\tilde{\dot{x}}_P\}_{m,1} & \{\tilde{\dot{x}}_P\}_{1,2} & \{\tilde{\dot{x}}_P\}_{2,2} & \cdots & \{\tilde{\dot{x}}_P\}_{m,\varrho_1} \\ \{\tilde{x}_V\}_{1,1} & \{\tilde{x}_V\}_{2,1} & \cdots & \{\tilde{x}_V\}_{m,1} & \{\tilde{x}_V\}_{1,2} & \{\tilde{x}_V\}_{2,2} & \cdots & \{\tilde{x}_V\}_{m,\varrho_1} \end{bmatrix} \tag{8.5}$$

$$\mathbf{Y}_{\text{FTDANN}_1} = \begin{bmatrix} \{F1\}_{1,1} & \{F1\}_{2,1} & \cdots & \{F1\}_{m,1} & \{F1\}_{1,2} & \{F1\}_{2,2} & \cdots & \{F1\}_{m,\varrho_1} \\ \{F2\}_{1,1} & \{F2\}_{2,1} & \cdots & \{F2\}_{m,1} & \{F2\}_{1,2} & \{F2\}_{2,2} & \cdots & \{F2\}_{m,\varrho_1} \\ \{F3\}_{1,1} & \{F3\}_{2,1} & \cdots & \{F3\}_{m,1} & \{F3\}_{1,2} & \{F3\}_{2,2} & \cdots & \{F3\}_{m,\varrho_1} \\ \{F4\}_{1,1} & \{F4\}_{2,1} & \cdots & \{F4\}_{m,1} & \{F4\}_{1,2} & \{F4\}_{2,2} & \cdots & \{F4\}_{m,\varrho_1} \end{bmatrix} \tag{8.6}$$

$$\mathbf{U}_{\text{FTDANN}_2} = \begin{bmatrix} \{x_V\}_{1,1} & \{x_V\}_{2,1} & \cdots & \{x_V\}_{m,1} & \{x_V\}_{1,2} & \{x_V\}_{2,2} & \cdots & \{x_V\}_{m,\varrho_2} \\ \{x_{V,ref}\}_{1,1} & \{x_{V,ref}\}_{2,1} & \cdots & \{x_{V,ref}\}_{m,1} & \{x_{V,ref}\}_{1,2} & \{x_{V,ref}\}_{2,2} & \cdots & \{x_{V,ref}\}_{m,\varrho_2} \end{bmatrix} \tag{8.7}$$

$$\mathbf{Y}_{\text{FTDANN}_2} = \begin{bmatrix} \{F5\}_{1,1} & \{F5\}_{2,1} & \cdots & \{F5\}_{m,1} & \{F5\}_{1,2} & \{F5\}_{2,2} & \cdots & \{F5\}_{m,\varrho_2} \end{bmatrix} \tag{8.8}$$

where

ϱ_1	is the number of fault modes in FTDANN$_1$, 5.	[-]
ϱ_2	is the number of fault modes in FTDANN$_2$, 2.	[-]
m	is the number of test trajectories, 3.	[-]
NARX	is an index to indicate data applied for NARX network.	[-]
FTDANN$_1$	is an index to indicate data applied to diagnose the faults $F1$-$F4$.	[-]
FTDANN$_2$	is an index to indicate data applied to diagnose the fault $F5$.	[-]
p_A	is the A pressure.	[Pa]
p_B	is the B pressure.	[Pa]
x_P	is the piston position.	[m]
$\dot{x}_P$	is the piston velocity.	[m/s]
$x_{P,ref}$	is the piston position reference.	[m]
x_V	is the valve position.	[-]
$x_{V,ref}$	is the valve signal.	[-]
$F1$	is the internal leakage flow.	[l/min]
$F2$	is the external A leakage flow.	[l/min]
$F3$	is the external B leakage flow.	[l/min]
$F4$	is the non-dimensional pressure failure.	[-]
$F5$	is the non-dimensional stuck valve failure.	[-]

The training matrices have the characteristic that all columns can be swapped provided its counterpart in the corresponding matrix is swapped in a similar manner. This allows for training of ANN with application of all the time series input and output data series simultaneously, which not only shortens the training period but also decreases the risk of overfitting. Furthermore, the issues connected with input and output fitting with ANNs are elaborated in the following.

The Bias-Variance Dilemma

The Bias-Variance dilemma from [R. Isermann, 2010] describes a trade-off between the bias error and the variance error. This rises from supervised training algorithms which will prevent the ANN from generalising beyond their training. The dilemma is illustrated in fig. 8.3, where the bias error is a systematic deviation from the true output of the system. Since it is a systematic deviation, it will cause a distortion from the true output in a predictable direction. This occurs when the ANN model does not have enough flexibility (number of neurons and delays) to fit the real system (underfitting). The bias error decreases, as the ANN complexity rises. This increases the variance error which is an indication that the ANN attempts to fit the noise instead of the actual output. This is referred to as overfitting and is caused by the ANN adapting to the exact tendencies in the training data set.

The Bias-Variance dilemma is handled by training the ANN through a two step procedure. The first step is the training of the ANN, where the weights of the ANN are optimised to give the best possible fit to the training data. The second step is the generalisation, here the ANN is used to simulate new data, which have not been part of the training data set. A well generalised ANN will have good estimations of the new unseen data, and the training data. If the trained ANN is capable of estimating the training data precisely but fails to estimate the new unseen data, the ANN have been trained to overfit the data. The application of this procedure is described in the following section.

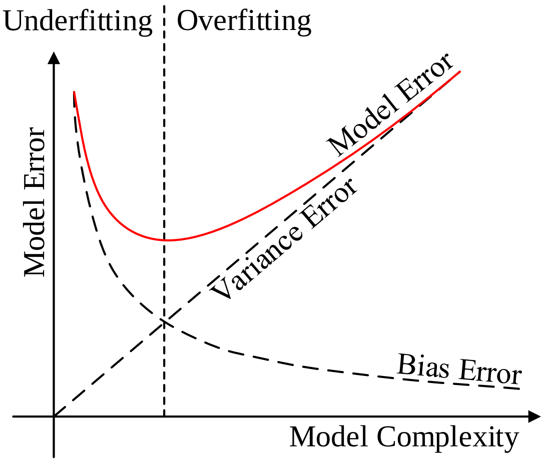


Figure 8.3: The trade-off between the bias- and variance- error, with inspiration from [R. Isermann, 2010].

Data Division for Enhanced ANN Generalisation

The ANN's generalisation capabilities may be enhanced by using the general practice from [M. H. Beale, 2016], where the data is divided into three random or predefined subsets: Training set, validation set, and test set, where the ratio of data applied is 70%/15%/15% respectively, based on recommendations from [M. H. Beale, 2016]. However, other division fractions have been applied by [P. S. Crowther, 2005], where it is also noted, that the division of data may be a dynamic process depending on the application.
The training set is used for calculating and optimising the ANN weights and biases.
The validation set is used to evaluate the generalisation when the ANN is presented to unseen data from the validation set.
The test set can by used to evaluate the performance of the design during the training process.
The performance evaluation of an ANN can be illustrated by plotting the training-, validation- and test set mean squared error as the training of the ANN progresses. An example of such a plot is shown in fig. 8.4, where an epoch defines the number of times the ANN has been presented to the entire data set.

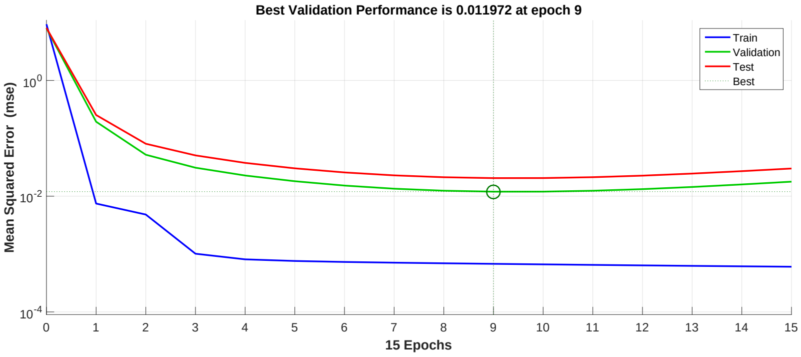


Figure 8.4: Illustrative example of training-, validation- and test mean square error performance plot.

Fig. 8.4 shows that after 9 epochs the ANN slowly starts overfitting, which is revealed by the rising validation error. When the validation and test error reaches a minimum and the design

cannot be improved further, the design of weights and biases is saved.
The explained procedure allows for the overall performance of the ANN with the given data set to be determined.

The division of data cannot address the Bias-Variance dilemma completely since it cannot adjust the complexity of the ANN model. Therefore it is proposed to perform a generic optimisation with the purpose of locating the optimum model complexity.

8.2.2 Optimal ANN complexity for the FDD scheme

Based on the previous discussion about the ANN *Bias-Variance dilemma* and since there does not exist any design rules for ANNs, it is proposed to apply an optimisation algorithm. This is primarily chosen due to the prior presented theory from [R. Isermann, 2010], stating that there is a clear connection between ANN performance and ANN complexity. It is therefore expected that an optimisation will be able to locate the most suitable design in a more time efficient manner compared with simply brute forcing all possible designs in the set.

In this thesis it is not the objective to search for the global optimum design, however, it is desired to rapidly locate a design that fit the specific FDD scheme. Therefore, an evaluation of ANN performance after a given period of training (10 min) is utilised, i.e. a design with fast convergence and sufficient performance is found. For further aspects it is suggested to remove this limit.
The design is found via the genetic algorithm (GA) by MathWorks, which has been chosen since its heuristic and since the cost function is not guaranteed to be convex.

The basic theory and explanations for the applied optimisation is presented in app. C.2, and the specific optimisation settings are presented below.

The optimisation vectors are given by (8.9).

$$\boldsymbol{x}_{\text{NARX}} = \begin{bmatrix} d_u & d_y & n \end{bmatrix}^T \qquad \boldsymbol{x}_{\text{FTDANN}} = \begin{bmatrix} d_u & n \end{bmatrix}^T \tag{8.9}$$

where

d_u is the number of input delays.
d_y is the number of output delays.
n is the number of neurons.
$\boldsymbol{x}$ is the design parameter vector.

The initial guess for the input vectors are given in (8.10).

$$\boldsymbol{x}_{\text{NARX},0} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T \qquad \boldsymbol{x}_{\text{FTDANN},1,0} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T \qquad \boldsymbol{x}_{\text{FTDANN},2,0} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T \tag{8.10}$$

Furthermore the upper and lower bounds have been specified as seen in (8.13).

$$1 \leq x_{\text{NARX}}(1) \leq 10 \quad ; \quad 1 \leq x_{\text{FTDANN}_1}(1) \leq 30 \quad ; \quad 1 \leq x_{\text{FTDANN}_2}(1) \leq 30 \tag{8.11}$$

$$1 \leq x_{\text{NARX}}(2) \leq 10 \quad ; \quad 1 \leq x_{\text{FTDANN}_1}(2) \leq 8 \quad ; \quad 1 \leq x_{\text{FTDANN}_2}(2) \leq 8 \tag{8.12}$$

$$1 \leq x_{\text{NARX}}(3) \leq 6 \tag{8.13}$$

The upper and lower bounds have been specified from knowledge about previously successful designs for similar problems. This was found in the articles [W. J. Crowther, 1998] & [M. Karpenko, 2002] during the literature review in chap. 3.

The fitness function is defined to be the sum of squared errors (from (7.11) in sec. 7.3.2) (8.14) after a 10 min period. This enhances the convergence period of the GA and furthermore ensures that an ANN design with fast convergence is obtained.

$$f(\boldsymbol{x}) = \frac{1}{2} \sum_{p=1}^{P} \sum_{m=1}^{M} \boldsymbol{e}_{p,m}^{2}(\boldsymbol{x}) \tag{8.14}$$

where

$\boldsymbol{e}_{p,m}$	is the error at output m when pattern p is applied.
p	is the index of patterns, from pattern 1 to total number of patterns P.
m	is the index of the outputs, from output 1 to total number of outputs M.
$\boldsymbol{x}$	is the design parameter vector.

A total of five fitness values are calculated for each generation, and the algorithm is restricted to optimise for a total of 50 generations. This is chosen so that the algorithm is not limited by number of iterations, which is confirmed by the convergence plot in fig. 8.5.

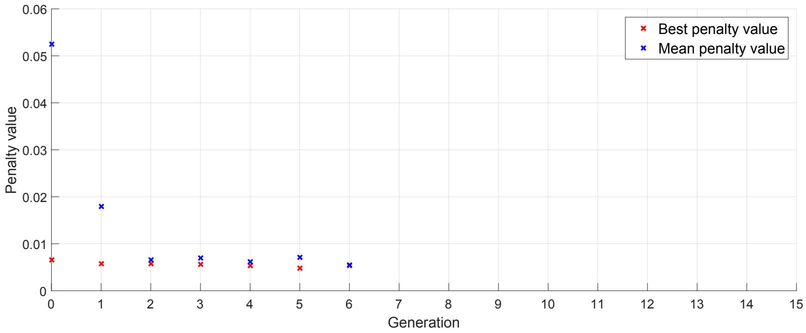


Figure 8.5: Convergence plot of the GA algorithm during optimisation of the FTDANN$_1$.

The optimum vector designs can be seen in tab. 8.1.

Network	d_u	d_y	n
NARX	9	2	4
FTDANN$_1$	6	-	6
FTDANN$_2$	12	-	5

Table 8.1: Results of the GA optimisation.

The optimums are in a range similar to the ones found throughout the literature review and hence the design is considered reasonable. It is not certain that the designs are optimal when the algorithm is permitted longer training periods, but this is irrelevant for the purpose of this thesis. The increase of performance will be evaluated in sec. 8.3 where the performance of the optimum FDD design is compared with the one from the preliminary tests from 7.3.3 & 7.4.2.

8.2.3 Training & Test Sequences

The purpose of the training and test sequences is to validate the FDD scheme at various conditions. Hence, it is not a requirement that this is designed with a realistic representation of the external load and/or position trajectories occurring in a wind turbine.

Instead simple conditions will be defined with the possibility of adding stochastic characteristics. From this consideration a rapid development and testing of the FDD scheme may be achieved, which for this thesis is considered sufficient as a proof of concept.

Applied External Load

The external load has been designed to disturb the hydraulic system with a pseudorandom term so that the FDD scheme cannot simply memorise the output. It is not necessary to design a replicate of an actual wind load, since it is solely desired tested if the FDD scheme can operate at an arbitrary load condition. Which may be confirmed/denied with the proposed load in (8.15).

$$F_L = c_{Gaus,L}\frac{1}{\tau_{s,L}s+1} + a_{b,L}\sin(\omega_{b,L}t) + b_{b,L} \tag{8.15}$$

where

$a_{b,L}$	is the load base amplitude.	7 kN
$b_{b,L}$	is the bias on the base amplitude.	10 kN
$c_{Gaus,L}$	is Gaussian white noise with frequency of 63 rad/s and noise power 5	kN
$\omega_{b,L}$	is the wind load base frequency.	1 rad/s
$\tau_{s,L}$	is the first order system time constant.	0.1 s

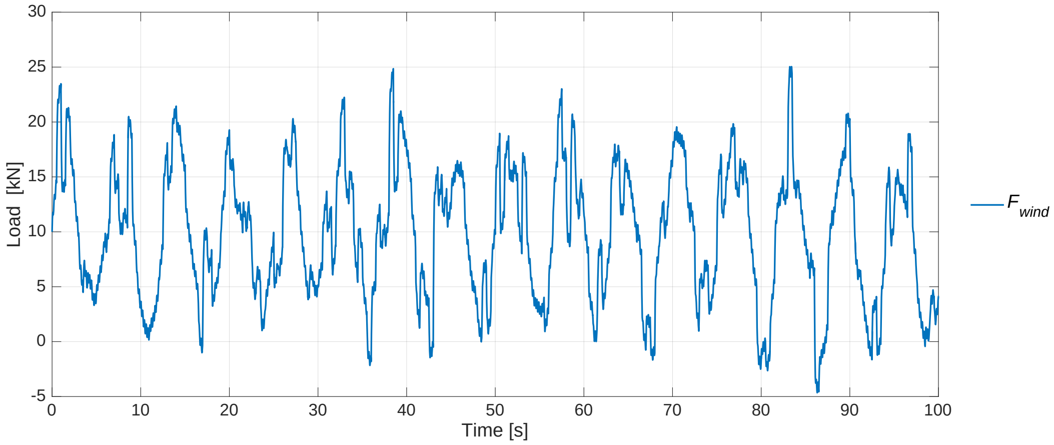


Figure 8.6: An example of the external load.

The load has a repeating characteristic, however, the Gaussian steps result in a noticeable change, which will appear as random disturbances.

Training Trajectories

The training trajectories have been designed to simulate three situation with significantly different behaviour, which will make it possible to test the FDD schemes' versatility.

The three situations are described below.

Sine Wave Reference
A sine wave is chosen due to its repeating dynamic behaviour, which eases the learning problem for the ANNs. A base frequency of 3 rad/s and amplitude of 4% of full stroke (20 mm)

is applied. Similar reference is applied for a similar problem in [L. An, 2014b] thereby making the results comparable.

Sine Wave with Gaussian White Noise Reference
This reference is chosen to maintain the repeating behaviour of the sine wave. The stochastic behaviour allows for investigation of the schemes ability to learn the symptoms at random inputs. Hence, it will not be possible to memorise the exact pattern of the symptoms.
The reference is described by (8.16). The Gaussian white noise step is fed through a first order system to avoid heavy velocity spikes. The reference is inspired by [L. An, 2014b] where an approach with pseudorandom steps was applied for a similar problem.

$$x_{P,ref} = c_{Gaus,P}\frac{1}{\tau_{s,P}s+1} + a_{b,P}\sin(\omega_{b,P}t) \tag{8.16}$$

where

$a_{b,P}$	is the pitch base amplitude.	0.02 m
$c_{Gaus,P}$	is Gaussian white noise with frequency of 4.15 rad/s and noise power 0.0002	m
$\omega_{b,P}$	is the base frequency.	3 rad/s
$\tau_{s,P}$	is the first order system time constant.	0.5 s

Constant Reference
A constant reference is chosen since it will contribute information about the schemes capabilities when the system is not excited.

8.3 Evaluation of FDD Schemes Ability to Recognise Several Fault Patterns

The performance of the developed FDD scheme will be evaluated in a strategic manner that allows interpretation of each component of the designed scheme. Therefore, it will initially be tested if accurate fault diagnosis can be obtained with a classic state estimator resulting in ideal residuals ($\mathbf{r}_{ideal}$) for each of the afore mentioned trajectories.
The performances of the above test sequence are compared with the performance of the preliminary design from sec. 7.6 where only one input was used for training. Thereby the FTDANN's ability to recognise different patterns is evaluated. These results will furthermore be compared with a scheme that does not include velocity measurements. This is used to evaluate the increase/decrease in performance when the velocity measurement is applied.
The schemes ability to generalise is tested by introducing noise on the measurements and evaluating the diagnosis performance.
Finally, the designed NARX estimator is tested by comparing its estimated outputs with the actual outputs.

The estimated faults from the FDD scheme are depicted in fig. 8.7, 8.8 & 8.9. Furthermore, the actual position and reference are shown. It should be noted that the applied controller is a standard P-controller. This is chosen because of its simplicity and since the tracking performance is not relevant at the current stage of the FDD evaluation.

It is observed that the scheme has high performance for all faults shown in fig. 8.7 & 8.8. Therefore it is concluded that the FTDANN has adapted to recognise both known and unknown

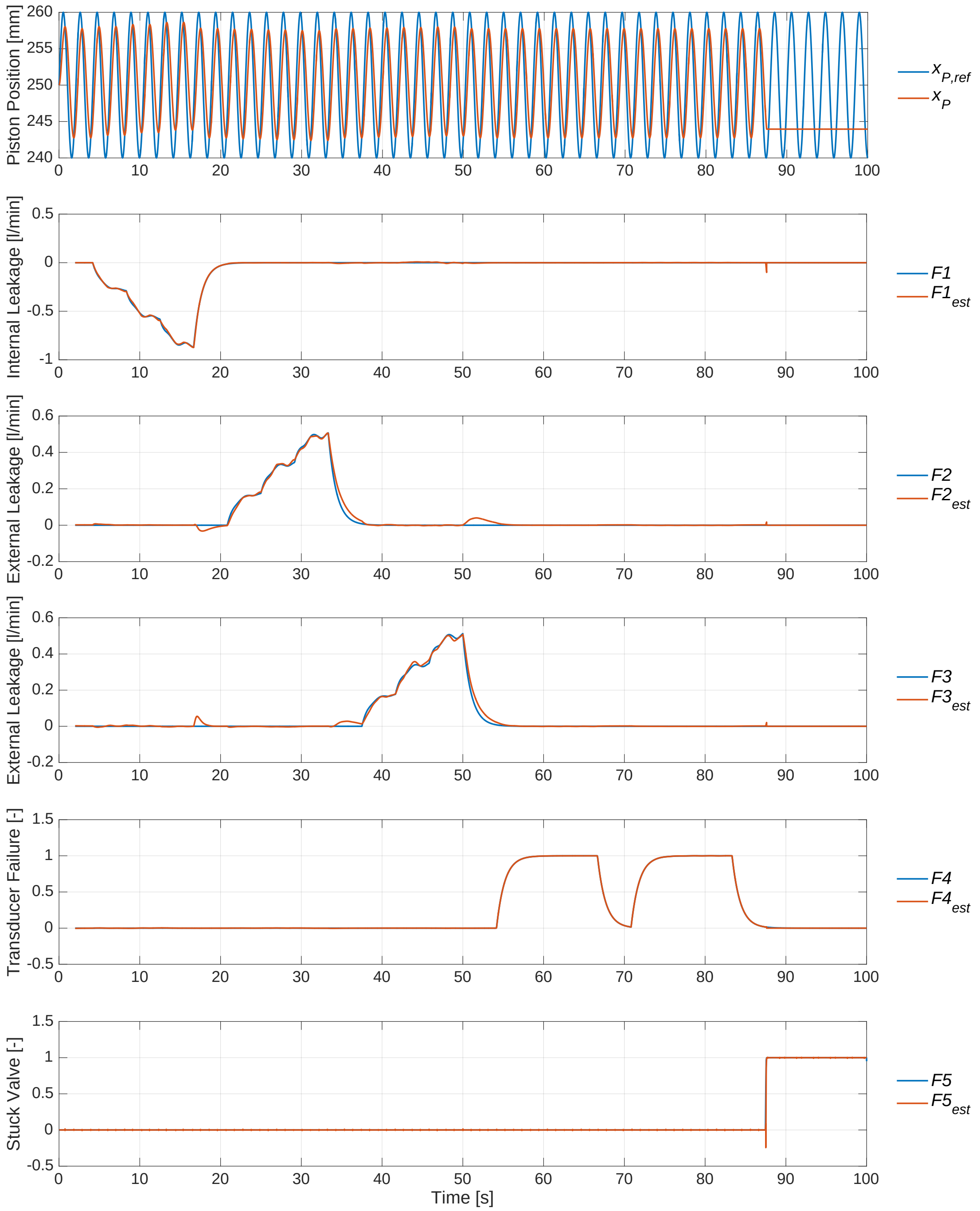


Figure 8.7: Two second EMA of estimated and actual simulated- faults when ideal residuals ($\mathbf{r}_{ideal}$) are applied.

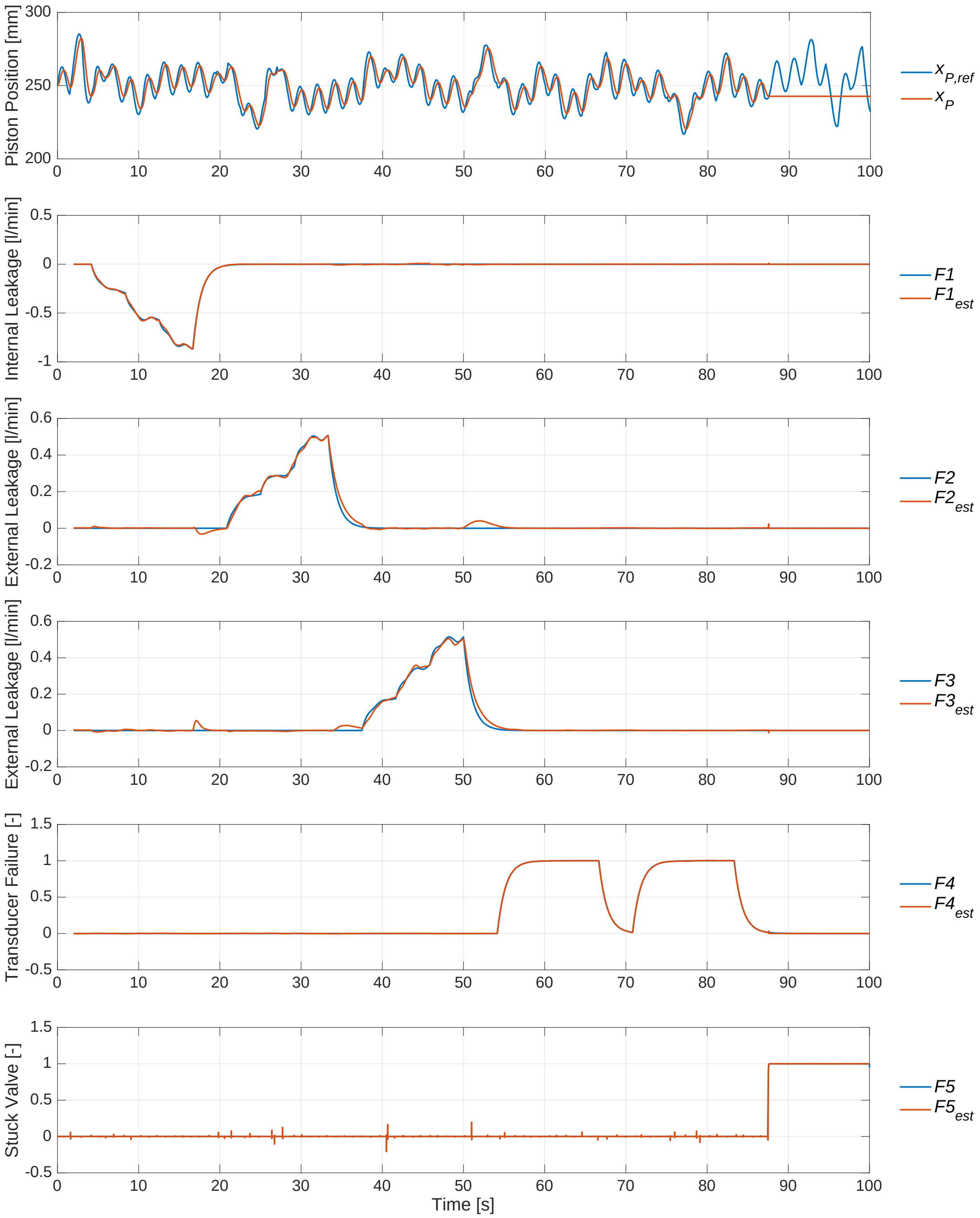


Figure 8.8: Two second EMA of estimated- and actual simulated faults when ideal residuals ($\mathbf{r}_{ideal}$) are applied.

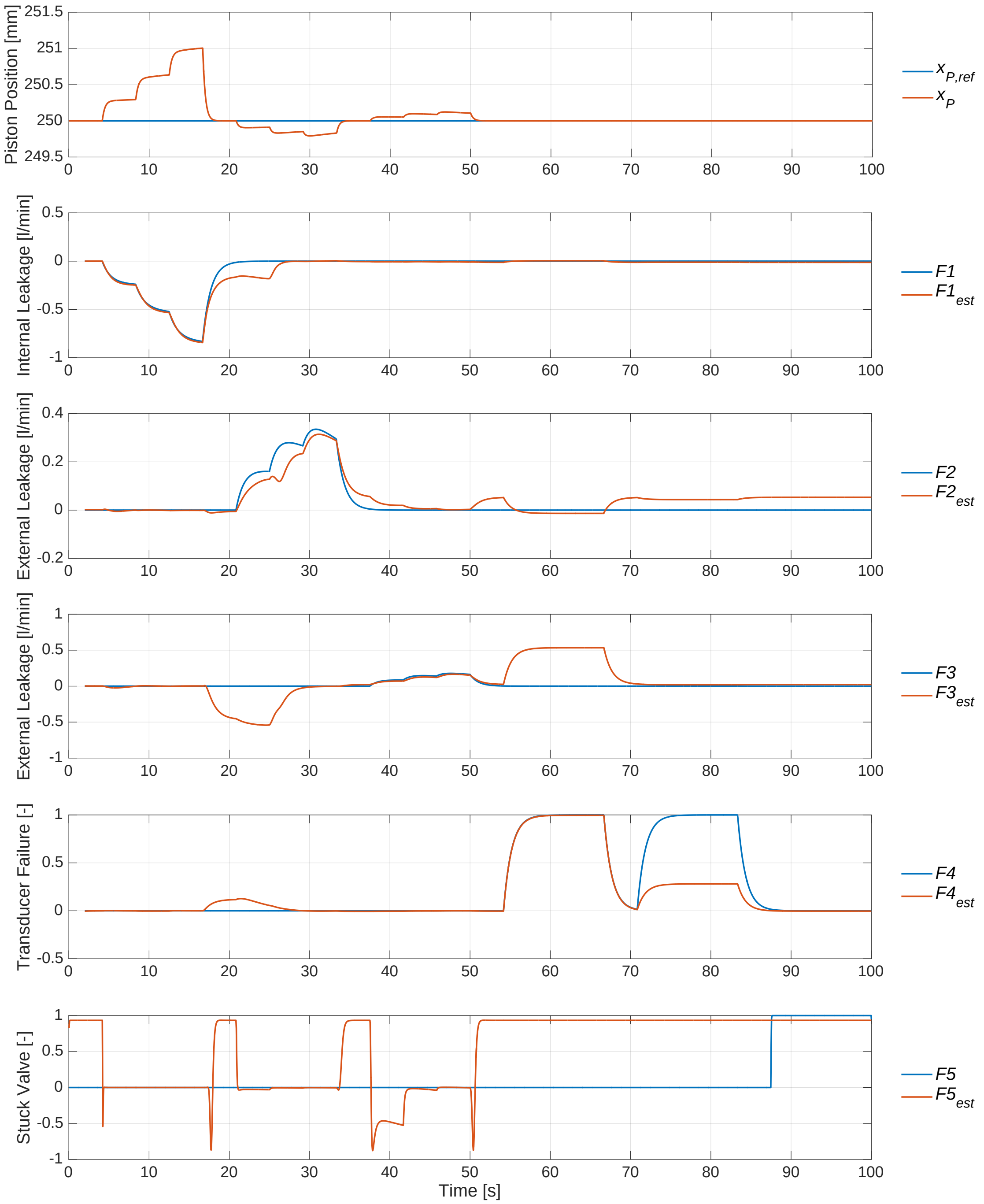


Figure 8.9: Two second EMA of estimated- and actual simulated faults when ideal residuals ($\mathbf{r}_{ideal}$) are applied.

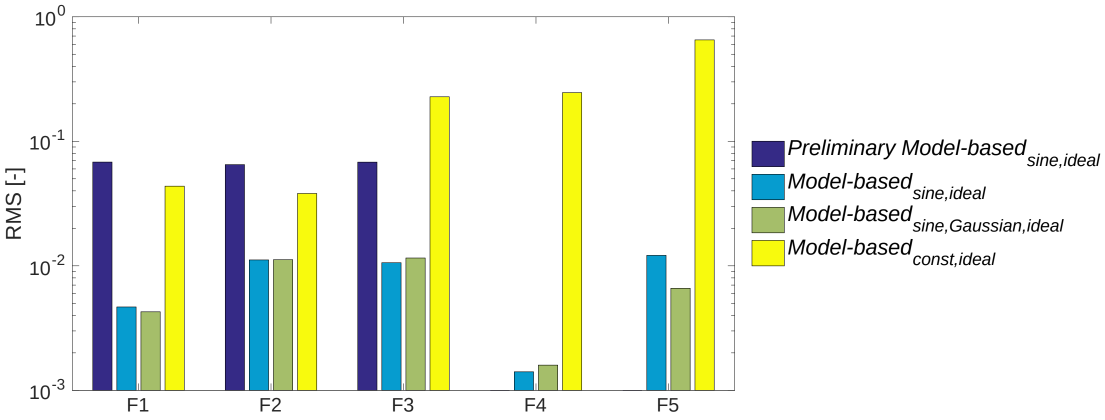


Figure 8.10: RMS error of estimated- and actual faults from results shown in fig. 8.7, 8.8 & 8.9. The y-axis is logarithmic due to the significant difference in RMS values.

patterns of residuals. The preliminary performance of $F1$-$F3$ depicted in fig. 8.10 shows that the expanded- and optimised- scheme has improved significantly.
In fig. 8.9 the results of the situation of constant reference are shown, it is observed that the diagnosis is more difficult when the system is in steady-state. In general this situation is considered to result in insufficient performance, especially the stuck valve estimation is invalid for this situation. This is considered caused by the limited required control signals, which are only applied when the piston drifts due to leakage.

The hypothesis of adapting the FTDANN with multiple input sequences has been confirmed in 2 of 3 cases. The most important result is that the diagnosis is valid even though the position trajectory has a stochastic behaviour.

8.3.1 Evaluation with Noise & without Velocity Measurements

The results in the previous section are simulated without any measurement noise. It is desired investigated if the faults can be diagnosed when a Gaussian white noise is applied to the measurements. This can be used to prove if the trained FTDANN is adequately fitted to the data, i.e. if it is capable of recognising a known sequence with some deviation from the data applied for training.
The noise is introduced as a Gaussian distributed white noise added to the relevant states, which is designed as seen in tab. 8.2.

State	p_A	p_B	x_P	$\dot{x}_P$	x_V
Approximated noise gain [-]	0.75e5	0.75e5	0.0001	0.0001	0.001
Noise power [-]	0.0001	0.0001	0.0001	0.0001	0.0001
Measurement delay [ms]	4	4	0.5	0.5	0.5

Table 8.2: Values connected with the noise and delay on state measurements.

The noise levels have been designed to approximate the expected uncertainty of each measurement.
In reality there will be a delay on the measurements. This is in the range of a few milliseconds, and is therefore of no importance for the estimations of the developed FDD and will not be further investigated.

The generated residuals with noise on measurements are shown in fig. 8.11.

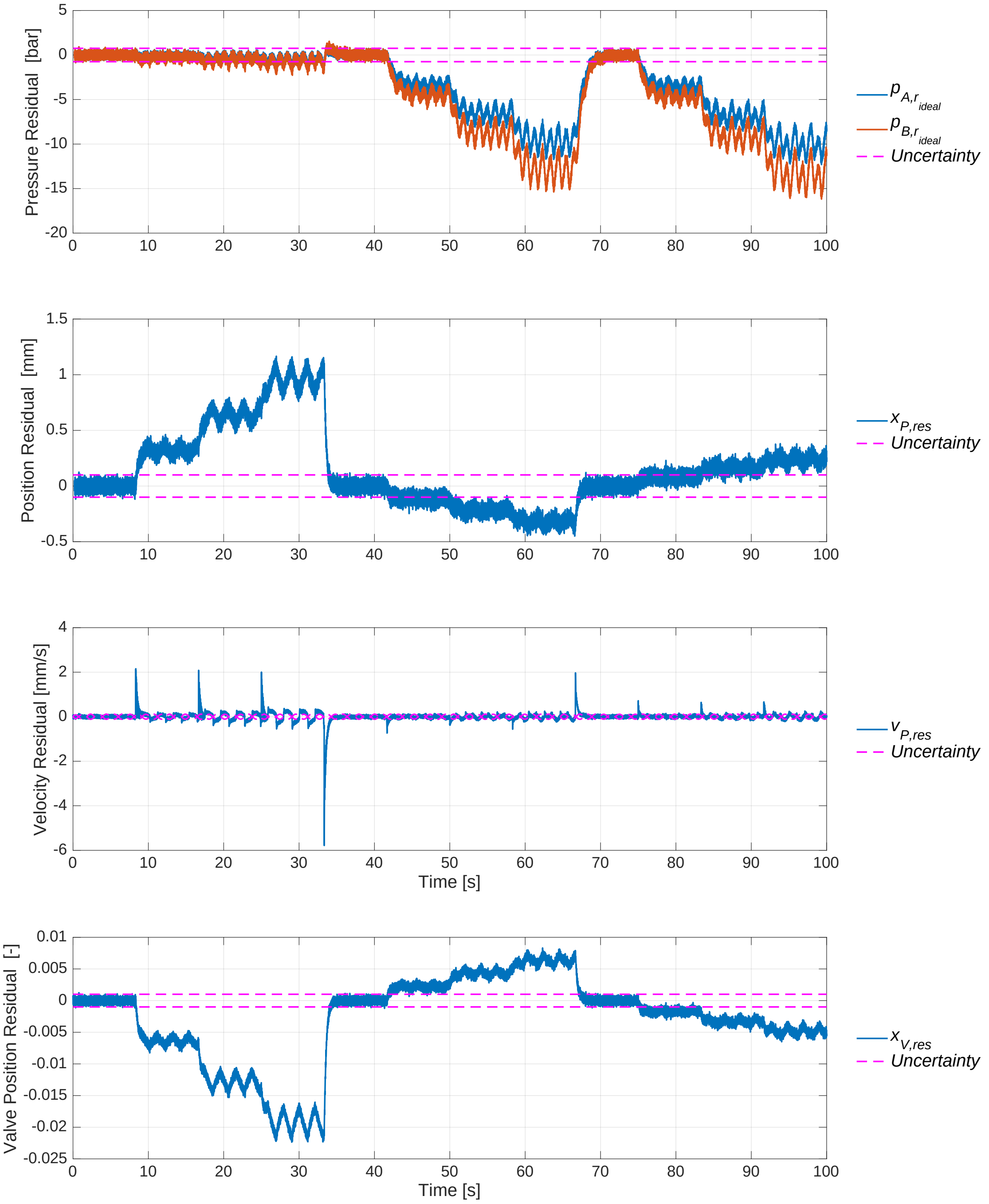


Figure 8.11: Residuals ($\mathbf{r}_{ideal}$) generated with the presented noise levels with a sine wave position reference and the leakage faults.

These residuals are applied to the FTDANN for diagnosis and since the results are quite similar to the ones seen in fig. 8.7, 8.8 & 8.9 the graphs with diagnosis results with noise and without velocity are omitted, but can be found in app. I in fig. I.10, I.11, I.12, I.13, I.14 & I.15.
The RMS values for these tests are given in tab. 8.12 with noise-free results for comparison. The results without application of the velocity measurements are also included in fig. 8.12.

It is observed that there is a difference in RMS errors with and without noise. The noise will in

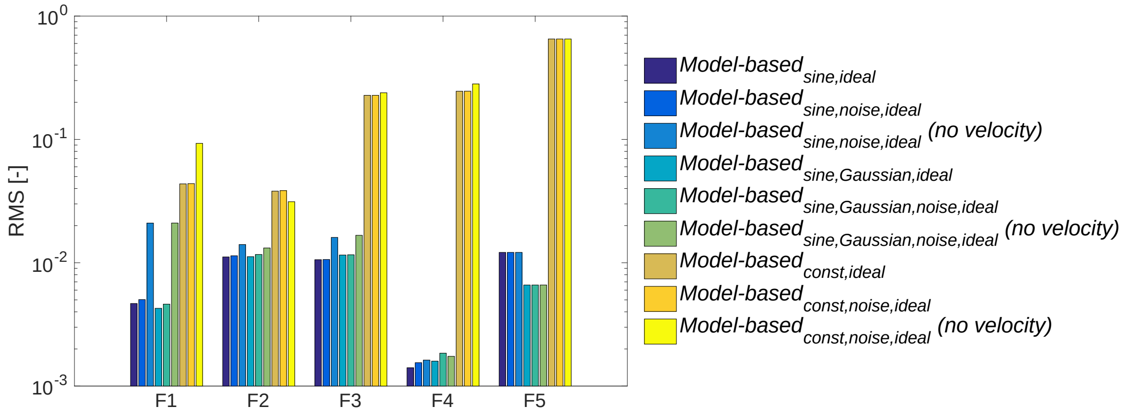


Figure 8.12: RMS error of estimated- and actual faults from results shown in fig. 8.7, 8.8 & 8.9 compared with the same sequence while noise is applied and also for the FDD scheme where the velocity is not included.

general worsen the estimation, however, the worst case RMS difference is around 0.0005 l/min which is considered acceptable.
Generally the noise is not considered to have any significant impact and the FTDANN$_1$ is therefore considered able to reject the simulated noise levels, even though it has not been trained with such data.

The test with noise and without the use of velocity reveals that the internal leakage ($F1$) has worsened significantly. Whereas the remaining faults do only show minor differences.
By analysing the result graphs from app. I in further detail it is observed that the estimation is in general delayed. This may be interpreted in the sense that the significant transient symptoms are dominating in the velocity residual which may also be observed from fig. 8.11. This feature can therefore be used to detect the fault more rapidly.
The delayed estimation will give a greater RMS error, but will not necessarily cause unreliable diagnosis.
For the specific faults it is therefore not considered essential to have velocity measurements.

8.3.2 Evaluation of Residuals Generated by NARX ANN

In the previous section it was proved that the faults could be estimated for three different situations both with and without noise. This was estimated with a classical state estimator used to generate the residuals.
In the following it will be discussed whether the NARX ANN is capable of estimating the system states with sufficient accuracy compared with the state estimator. This is used to analyse the possibility of applying NARX ANN for residual generation.

The generated residuals are investigated in a similar manner as presented in sec. 7.6 where the estimated states were evaluated. The three trajectories are applied, and by analysing the difference between the actual and estimated states, the accuracy can be evaluated. For the sake of brevity the result graphs are not presented here, but can be found in app. I in fig. I.1, I.2, I.3, I.4, I.5, I.6, I.7, I.8 & I.9.

The simulation results shows insufficient performance of the NARX ANN. Especially the stochastic position reference shows ±30 bar deviations on the chamber pressure estimates and

in general the NARX ANN has insufficient accuracy. This indicates that the NARX ANN has not been able to learn the actual system behaviour.
The decreased performance is connected with the increased training data, and the proposed training and NARX design is therefore not sufficient for residual generation valid for all input trajectories. However, if the FDD scheme is designed to diagnose exclusively when a certain input sequence occurs, it would be possible to train the NARX to this specific situation.

Based on this it cannot be proved that the training method can make the NARX ANN adapt to changing input sequences for the given system. In other words, further work must be done if this type of estimator is desired to work under any given conditions.
In the following section it will be investigated whether the NARX can be trained with a single repeating input sequence and with introduction of the external load defined in sec. 8.2.3.

8.4 Evaluation of the Fault Detection and Diagnosis Scheme with External Load

The NARX ANN is desired investigated while an external load is applied. The applied load was described in sec. 8.2.3 and it is expected that the load force is measurable. This is done to test if it is possible to teach the impact of a given external load to the ANN. The position trajectory is chosen as a sine wave, since the residual analysis revealed highest performance for this input.

The training procedure of the NARX ANN in this case is with the designed wind load applied as an exogenous input together with the reference signal.
The diagnosis results when using the residuals generated with the external load along with a sine position trajectory are presented in fig. 8.13.

The results shows similar tendencies to the ones previously shown in fig. 8.7. However, as seen in fig. 8.14 the overall performance of leakage estimations has decreased as a consequence of the external load.
The internal leakage is considered valid since it tracks the actual leakage when introduced, furthermore the inaccurate estimates occurs at $F4$ and only for a short temporary period.
The external leakage shows promising results, but it is not possible to distinguish which chamber is leaking. This indicates that the symptoms on the residuals are similar when a load is introduced. $F4$ & $F5$ are not affected in any significant manner by the introduced load, although fig. 8.14 shows that $F4$ has become more uncertain.

Based on these results it is possible to train the NARX ANN towards stochastic load conditions provided a repeating sine trajectory is applied and force measurement is available.

8.5 Evaluation of Gradual Leakage

It is desired to evaluate if the designed FDD scheme is capable of detecting gradual internal leakage over a long-period of time (1 week). If this can be diagnosed it is expected that certain symptoms are static and not only transient, since the fault does not have the abrupt behaviour from the previous tests. Furthermore, this test is used as a proof of concept to investigate whether the FDD scheme can recognise the symptoms in an arbitrary time frame.
The gradual leakage is assumed to occur linearly over time, and is therefore simulated by slowly

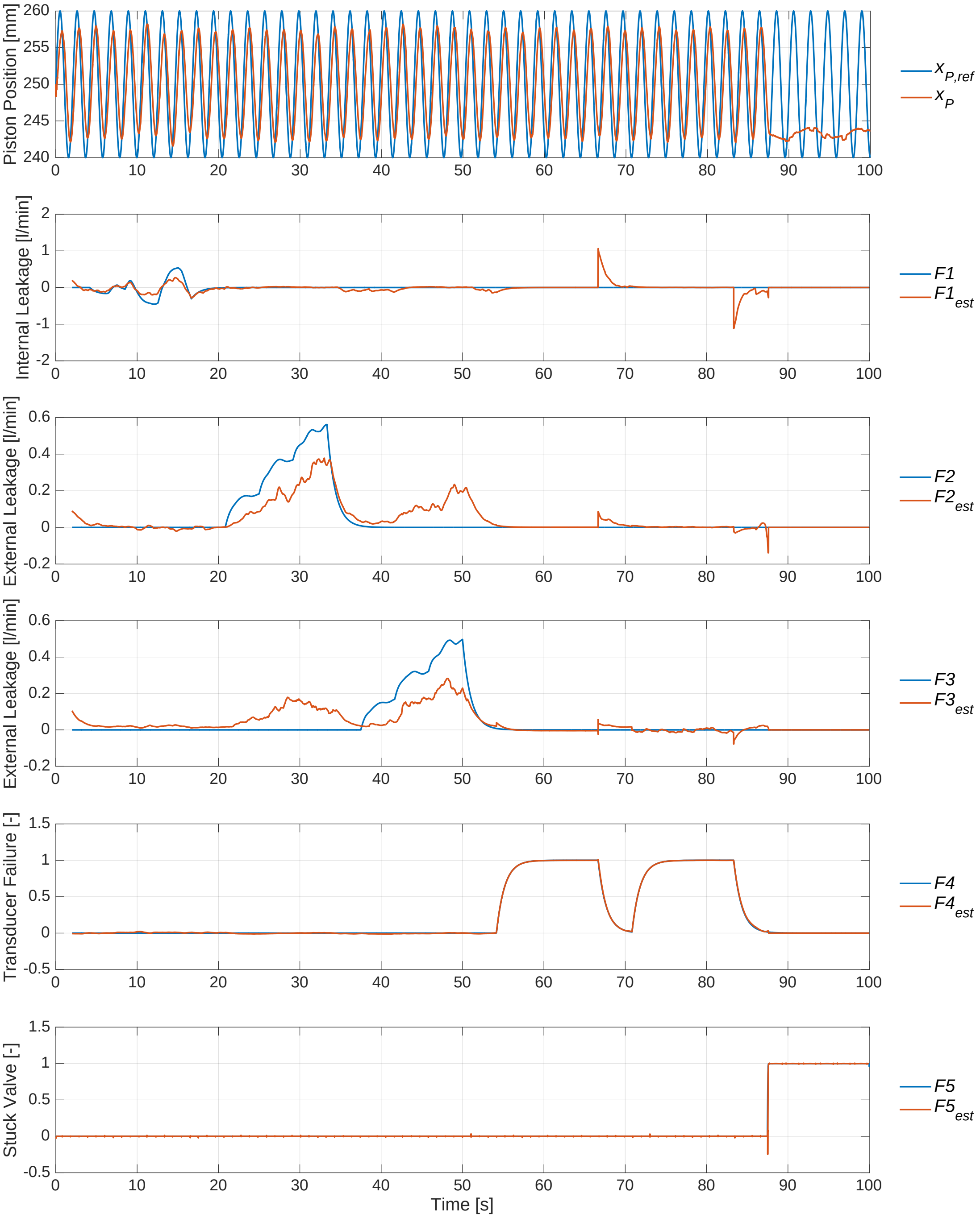


Figure 8.13: 2 second EMA of estimated- and actual simulated faults when applying the NARX ANN as estimator with a sine wave input and wind load trajectory.

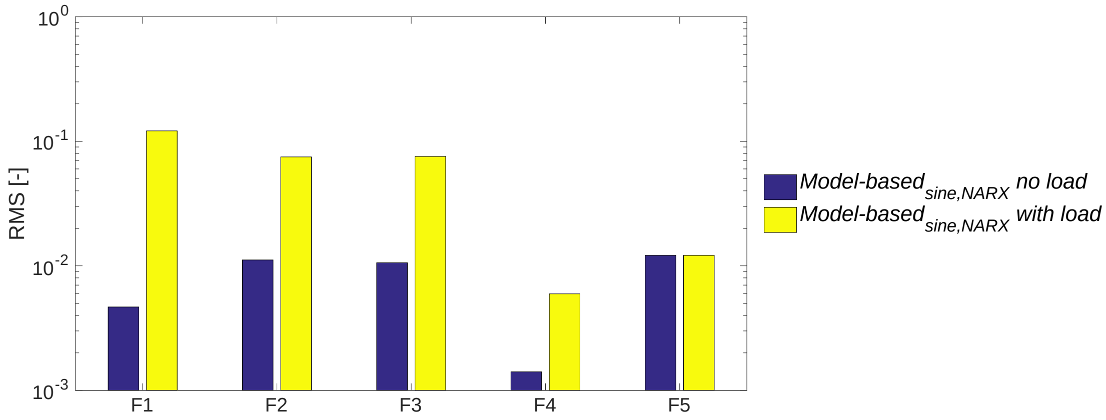


Figure 8.14: RMS error of estimated- and actual faults from the results shown in fig. 8.13 & 8.7.

opening the leakage valve in the model with a slope of 0.3 mm^2/7 days. In reality this is not necessarily the actual occurrence of the fault, but is however an approximation of wear in the system.

The gradual leakage tests are performed both with and without wind load since it was proved that the internal leakage could be detected with the NARX estimator during a sinusoidal reference with a measurable load.

The estimated and actual simulated faults are logged every two seconds and an EMA with a period of 2000 seconds is applied. This is done to limit the size of the data file, and the EMA is designed to filter this data sufficiently.

The results from the gradual leakage tests can be seen in fig. 8.15 & 8.16.

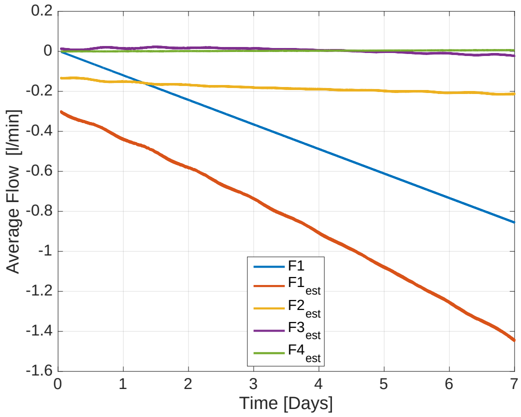


Figure 8.15: Simulated internal leakage flow compared with all fault estimations from FTDANN$_1$ without load.

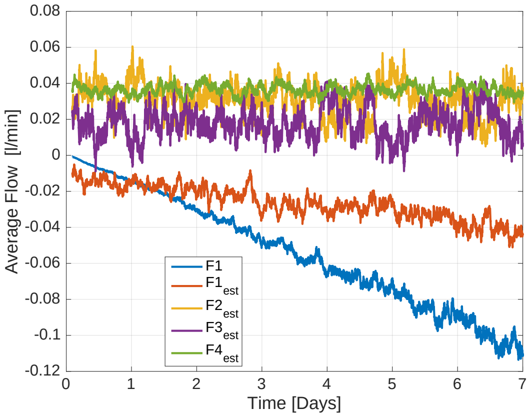


Figure 8.16: Simulated internal leakage flow compared with all fault estimations from FTDANN$_1$ with load.

The results for the test without any load gives a clear indication that an internal leakage is gradually increasing in the system. However, the estimation has an offset of ≈0.3 l/min. This is considered acceptable, since this type of fault indication is only proposed to be used for planning preventive maintenance on the given application. The estimation of external leakage in chamber A has an offset and slope which could cause misinterpretation. However, the estimation of $F1$ does show the most significant deviation from the normal value, and it can be concluded that the system is gradually failing.

The estimations while an external load is applied are not as accurate as the test without load. However, both the actual and estimated internal leakage are drifting, where the other fault estimations are at a constant offset. The results are on the periphery of what can be considered accurate enough to validate the FDD scheme during external load and gradual increase of $F1$. The argument speaking in favour of the validity is the estimated slope of $F1_{est}$.

Based on the diagnosis results without a load it is concluded that FTDANN$_1$ has been able to learn the correlation between the symptoms and the magnitude of the internal leakage. The estimates do not have equal accuracy as for the much shorter step tests presented in sec. 8.3, but it is expected that the proposed FDD scheme will be able to give a fair estimate of the faults even during long periods of time. Furthermore, the gradual tests are performed with the NARX estimator and if this is improved or a classic state estimator is used, the diagnosis is expected to perform better.

8.6 Evaluation of Data Driven FDD Schemes Ability to Recognise Several Fault Patterns

The model-based scheme from the above analysis relies on a model which was expected to be realised with application of a NARX ANN. This could not be validated at multiple input sequences, and it has therefore been considered to modify the scheme from fig. 8.2 by making it only based on data driven FDD. This is illustrated by fig. 8.17.

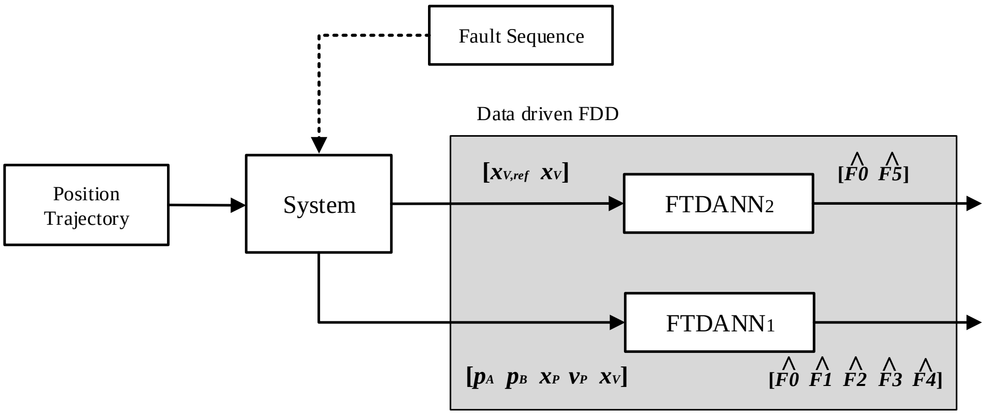


Figure 8.17: The expanded data driven FDD scheme, where the design of FTDANN$_1$ is re-trained with the actual system states.

FTDANN$_1$ is trained in a similar manner as was the case with a model-based topology which was explained in sec. 7.4 & 8.2. The three training trajectories without load or noise are used for evaluation. The results are shown in fig. 8.18. Graphs can be found in app. I in fig. I.16, I.17 & I.18.

The results in fig. 8.18 show that the RMS of the fault estimates from the data driven scheme is higher than the prior presented model-based accuracy. Only $F5$ is equal in both schemes. This is caused by application of equal FTDANN$_2$ in both schemes.

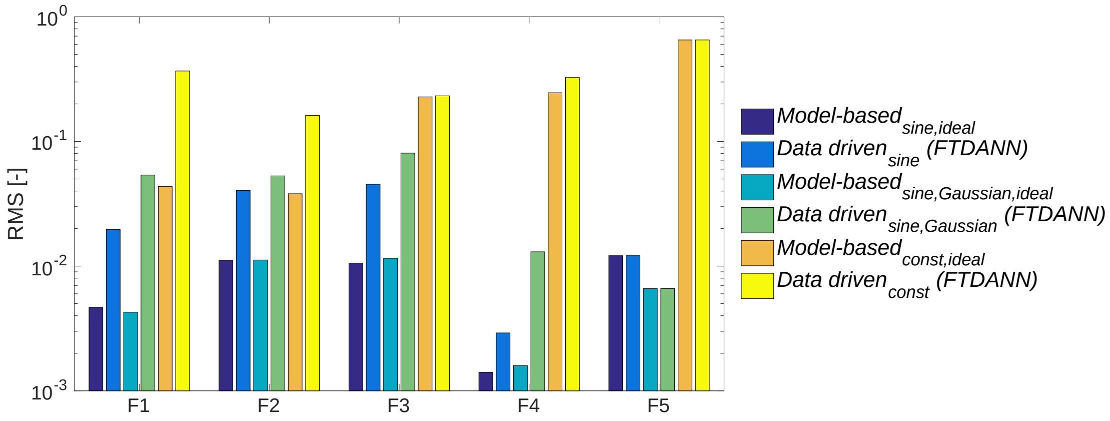


Figure 8.18: RMS error of estimated- and actual leakage from results shown in fig. 8.7, 8.8 & 8.9 compared with the same sequence applied to the data driven scheme shown in fig. 8.17.

It is concluded that the most promising FDD scheme is the model-based. This requires an experimentally verified state estimator, which for the given system is considered achievable provided the test from chap. 11 can be conducted.

In the following it is investigated if the FTDANN can be improved by adding an additional hidden layer.

8.7 Evaluation of Multilayer Focused Time Delay ANN

To investigate if the performance of the FTDANN is linked to the number of neurons in a single layer, or the number of layers in the design of the ANN, a Multilayer Focused Time Delayed Feed Forward ANN (MFTDANN) has been designed and tested. The MFTDANN is replacing FTDANN_1 from fig. 8.2 & 8.17.

A MFTDANN is designed for both the data driven and model-based scheme where the residuals are generated by the classic state estimator. Furthermore, the tests have been conducted without the overwrite rule described by algorithm 1. By excluding the overwrite rule, the MTDFANN enhanced learning capabilities (deep-learning) may be investigated. In fig. 8.19, a simplified schematic of the MFTDANN is illustrated. The schematic is based on the theory presented in sec. 7.4, but with the addition of a layer.

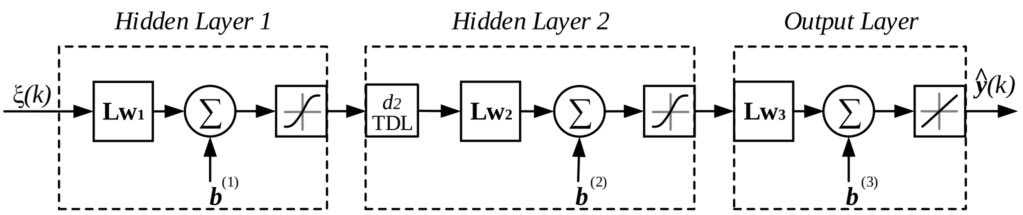


Figure 8.19: Simplified structure of Multilayer Focused Time Delayed ANN.

The MTDFANN design parameters are based on the ones applied for the FTDANN found in the study presented in sec. 8.3. This is done since it is not considered feasible to apply the GA optimisation on the MFTDANN, due to the increased time required for ANN training and hence also for GA convergence.

The total network function for the MFTDANN can be described by (8.17), which represents the structure from fig. 8.19.

$$\hat{\boldsymbol{y}}(k) = \mathbf{L}_{w3} \tanh\left(\mathbf{L}_{w2}\xi_{TDL}(k) + \boldsymbol{b}^{(2)}\right) + \boldsymbol{b}^{(3)} \tag{8.17}$$

The regressors $\xi(k)$ & $\xi_{TDL}(k)$ are given by (8.18).

$$\xi_{TDL}(k) = \begin{pmatrix} \tanh\left(\mathbf{L}_{w1}\xi(k) + \boldsymbol{b}^{(1)}\right) \\ \tanh\left(\mathbf{L}_{w1}\xi(k-1) + \boldsymbol{b}^{(1)}\right) \\ \vdots \\ \tanh\left(\mathbf{L}_{w1}\xi(k-d_2) + \boldsymbol{b}^{(1)}\right) \end{pmatrix} \quad ; \quad \xi(k) = \begin{pmatrix} \boldsymbol{u}(k) \\ \boldsymbol{u}(k-1) \\ \vdots \\ \boldsymbol{u}(k-d_1) \end{pmatrix} \tag{8.18}$$

The matrices and vectors are given by (8.19) & (8.20).

$$\mathbf{L}_{w1} = \begin{bmatrix} w^{(1)}_{1,1} & \cdots & w^{(1)}_{1,(d_1+1)i} \\ \vdots & \ddots & \vdots \\ w^{(1)}_{n_1,1} & \cdots & w^{(1)}_{n_1,(d_1+1)i} \end{bmatrix} \quad \mathbf{L}_{w2} = \begin{bmatrix} w^{(2)}_{1,1} & \cdots & w^{(2)}_{1,(d_2+1)n_1} \\ \vdots & \ddots & \vdots \\ w^{(2)}_{n_2,1} & \cdots & w^{(2)}_{n_2,(d_2+1)n_1} \end{bmatrix} \quad \mathbf{L}_{w3} = \begin{bmatrix} w^{(3)}_{1,1} & \cdots & w^{(3)}_{1,n_2} \\ \vdots & \ddots & \vdots \\ w^{(3)}_{o,1} & \cdots & w^{(3)}_{o,n_2} \end{bmatrix} \tag{8.19}$$

$$\boldsymbol{b}^{(1)} = \begin{pmatrix} b^{(1)}_1 \\ \vdots \\ b^{(1)}_{n_1} \end{pmatrix} \quad \boldsymbol{b}^{(2)} = \begin{pmatrix} b^{(2)}_1 \\ \vdots \\ b^{(2)}_{n_2} \end{pmatrix} \quad \boldsymbol{b}^{(3)} = \begin{pmatrix} b^{(3)}_1 \\ \vdots \\ b^{(3)}_{o} \end{pmatrix} \tag{8.20}$$

where n_1 denotes the number of neurons in hidden layer 1, n_2 denotes the number of neurons in hidden layer 2, d_1 denotes the input delays, d_2 denotes delays of hidden layer 2 and o denotes the number of outputs and i denotes the number of inputs. It should be noted that the output from the first hidden layer is stored $d_2 + 1$ times.
The training procedure is Levenberg-Marquardt with back-propagation presented in sec. 7.3.2.

Network	d_1	d_2	n_1	n_2	Training period
FTDANN$_1$	6	-	6	-	28 min
MFTDANN (Data-Driven)	6	6	6	6	5 h 23 min
MFTDANN$_{r_{ideal}}$	6	6	6	6	8 h 22 min

Table 8.3: Comparison of FTDANN and MFTDANN design and training period.

The evaluation of the multilayer schemes is done by comparing them with both the model-based scheme in sec. 8.3 without algorithm 1 and the data driven scheme in sec. 8.6 without algorithm 1. In this manner it is possible to investigate if the additional layer is an improvement of the previous most promising designs. The three training trajectories without load or noise are applied.
In fig. 8.3 the diagnosis results for each scheme can be seen. The corresponding graphs can be found in app. I.2 in fig. I.19, I.20, I.21, I.22, I.23 & I.24.

The RMS error results for the model-based scheme shows that the addition of a hidden layer does not enhance the performance significantly. It is therefore not considered a better alternative than applying the FTDANN$_1$ with algorithm 1. Furthermore, the results from fig. 8.10 where algorithm 1 was applied, show that this is superior with respect to the MFTDANN

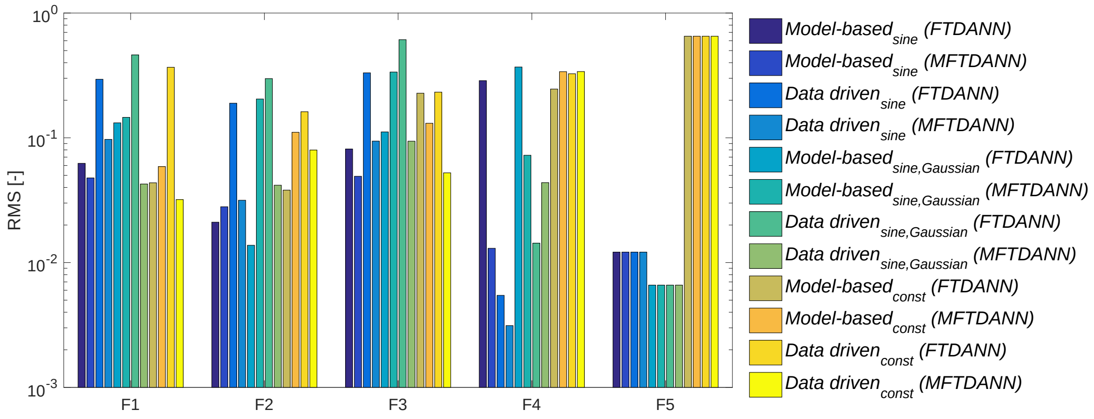


Figure 8.20: The RMS error of estimated- and actual leakage when FTDANN is applied without algorithm 1, compared with the RMS of the same sequence applied to the data driven and model-based FDD schemes designed with a MFTDANN.

scheme. Therefore, and due to the increased training period, multilayer diagnosis is not considered feasible in this scheme. The most promising model-based scheme is therefore, two FTDANNs with the proposed overwrite algorithm 1, although this involves manipulation of the outputs.

The results for the data driven scheme shows, that the MFTDANN outperforms the standard FTDANN in almost every fault indices. However, the increased performance comes at a cost of increased training period by a factor of 11. Depending on the application, the training period may or may not be considered an issue, since less data may be used, or better computer hardware is available, hence the absolute training period could be reduced. This issue is addressed in [X. Glorot, 2010] stating, that standard gradient descent training is not suited for deep feedforward ANN. Therefore, it is not considered feasible to increase the amount of layers any further, due to the increase of the training period.
The data driven results are considered to validate that an additional layer can enhance the diagnosis performance. Hence, further work should be addressed to investigate the benefits of increasing the amount of layers further while changing the training approach.
The main benefit of this MFTDANN data driven scheme is its independence of residual generation, and therefore a reduction of engineering effort may be realised.

8.8 Part Conclusion

A systematic test procedure has been performed to evaluate the capabilities of various FDD schemes.

The model-based scheme was trained to recognise different series of input residuals and faults with different characteristics were successfully diagnosed. This is an indication of the schemes versatility and it is proposed to investigate to what extend the list of faults can be expanded.
The stuck valve required additional attention and an alternative solution was suggested.

The model-based scheme proved to be sufficient with respect to generalisation, i.e. it functioned as intended to unseen data. Furthermore, a comparison of the scheme with and without application of the velocity was performed. This yielded that the benefits of applying velocity did not outweigh the additional complications connected with such measurements, for the specific application of interest in this project.

The NARX ANN was not able to predict the desired outputs with different series of exogenous inputs.
During the work of this thesis it was investigated if the NARX ANN could be trained in a similar manner as the system identification problem for a hydraulic servo system described in [L. Li, 2013]. This was done with a slow and fast series of step inputs (to excite the valve dynamics and the actuator dynamics separately). However, this did not yield any promising results.
Further work must be addressed to design a NARX estimator for this specific system, alternatively a classic state estimator may be applied.

Introduction of a stochastic load complicated the FDD problem and resulted in poorer performance on $F2-F3$, even though the force acting on the actuator was measurable. The estimations of the remaining faults were with an acceptable RMS error.

A simulation with a gradual increase of $F1$ showed promising estimation without load, even though the FDD scheme had not been trained with this exact type of fault. This indicates that the scheme is capable of recognising the static values of the symptoms and connecting them to the different faults with correct severity and time of occurrence. When a load was introduced the estimations were questionable.

A pure data driven scheme was proposed, due to the complications with the NARX estimator. This resulted in significantly poorer diagnosis performance, however, with the benefit of eliminating the need for an estimator.

An additional hidden layer was added to FTDANN_1 in the data driven and model-based topology. This did not give enhanced performance in model-based FDD. However, the data driven scheme was improved significantly, and the requirement for a manual overwrite was reduced. However, the required training period with the Levenberg-Marquardt algortihm was considered infeasible if additional complexity was desired. Therefore, a deep layer network scheme based on reinforcement learning could serve as a future aspect to design a general solution to the FDD problem.

The performance of several FDD schemes has been evaluated via simulation. It is proposed to test the model-based scheme with a sine trajectory on the actual test rig to investigate if similar results can be obtained practically. This is presented in chap. 11.

Part III

Supervisory Control and Data Acquisition Design

CHAPTER 9

ANALYSIS & DESIGN OF CONTROL SYSTEM

Contents

In the following chapter, control strategies for the test rig are designed. These will be used to control the piston position of the test hydraulic servo system (HSS) and the force of the load HSS.
The purpose of the control strategies is to be able to use the test rig in a similar manner as done for the Fault Detection and Diagnosis (FDD) simulations where a desired load can be implemented. The subscript t and l are used to denote test- and load HSS respectively.
The derivation and verification of the linear model can be found in app. A.

9.1 Analysis of the Control Problem

The two actuators in the system are controlled independently with one servo valve for each actuator. Due to their physical coupling the independent control introduces an additional problem which need to be addressed. To solve this problem, it is proposed to apply feedforwarded information from the position servo reference when controlling the force in the load actuator. The solution is shown in sec. 9.3.
The control problem is considered as two SISO systems with an unknown disturbance (from the opposing actuator). This is done to simplify the control design while fulfilling the control specifications which are explained later in this chapter.

In chap. 8 it was found that promising results were achieved when a sinusoidal reference with or without a stochastic term and a constant load were applied.

The TF's of the systems are desired investigated at different piston position to confirm that a single linear controller is feasible in the entire working area. The bode plots corresponding to the TFs for position and force at different piston positions are presented in fig. 9.1 & 9.2.

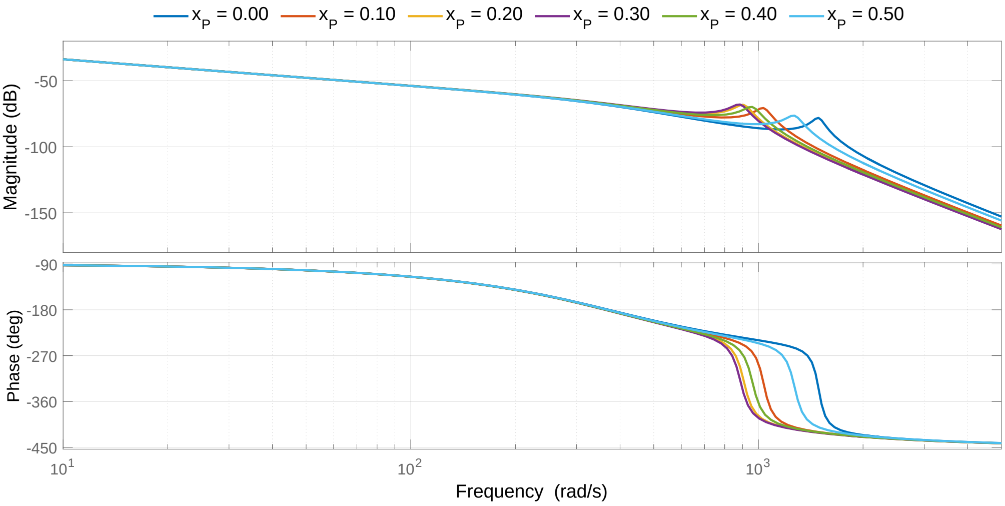


Figure 9.1: Open-Loop frequency response of the piston position in the test actuator, for different linearisation points.

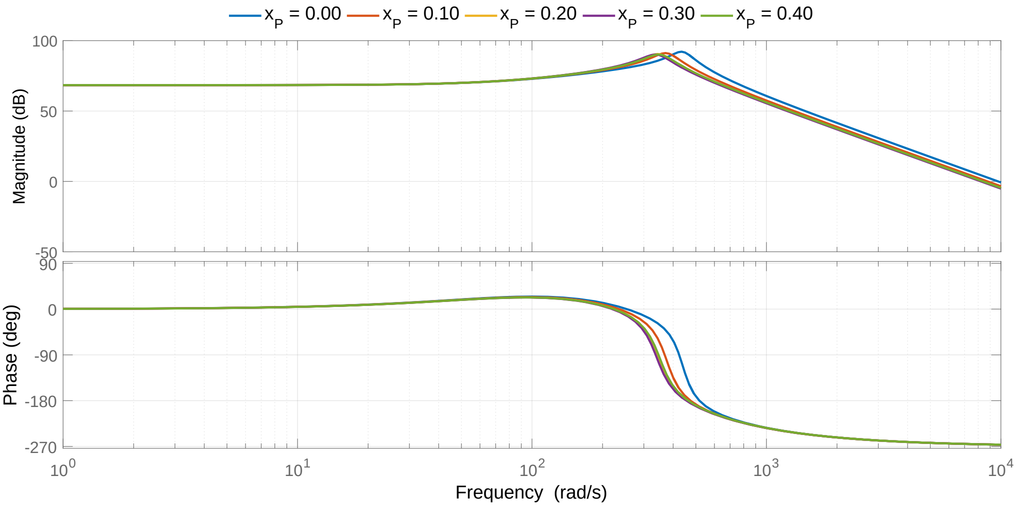


Figure 9.2: Open-Loop frequency response of the piston force in the load actuator, for different linearisation points.

The derivation of the TFs for the bode plots in fig. 9.1 & 9.2 can be found in app. A. It is observed that the frequency response of the two respective hydraulic servo systems does not vary significantly although the linearisation point is changed. Therefore it is verified that a standard linear controller for both systems is feasible.

The derived TFs will be applied when designing linear control strategies in the following.

9.2 Position Control of Test Actuator

Tracking of the position reference is done by applying a feedback of the test side piston position, comparison of these gives e_t which is directed to the developed control strategy. The structure of this is shown in fig. 9.3.

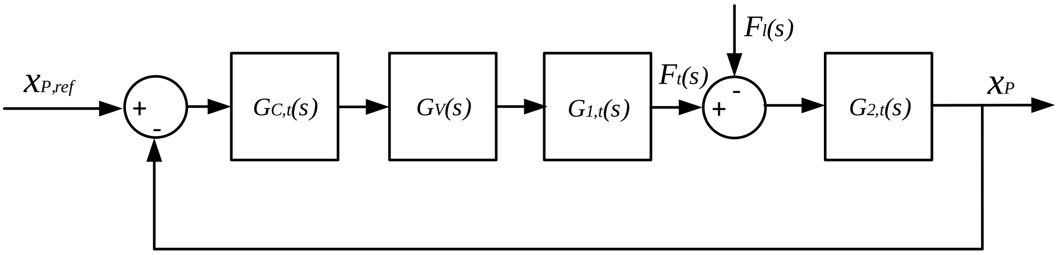


Figure 9.3: Block diagram of test side position servo control. The load actuator force is considered as an unknown disturbance.

The TFs from fig. 9.3 can be described by (9.1), (9.2) (from sec. 5.1), (9.3) (from app. A) & (9.4) (from app. A).

$$G_{C,t}(s) = \frac{u_{V,t}(s)}{e_t(s)} = K_p \tag{9.1}$$

$$G_V(s) = \frac{x_{V,t}(s)}{u_{V,t}(s)} = \frac{1}{\frac{s^2}{\omega_{n,V}^2} + 2\zeta_V \frac{s}{\omega_{n,V}} + 1} \tag{9.2}$$

$$G_{1,t}(s) = \frac{F_t(s)}{x_{V,t}(s)} = \frac{A_{A,t} k_{Q,t} (ms + B)}{ms^2 + Bs - k_{QP,t} ms - k_{QP,t} B + A_{A,t} A_{P,t}} \tag{9.3}$$

$$G_{2,t}(s) = \frac{x_P(s)}{F_t(s)} = \frac{1}{(ms + B)s} \tag{9.4}$$

The mass (m) & damping (B) applied in the TFs are the combined mass and damping acting on the moving parts.

The argumentation for applying a P-controller and design of (9.1) are presented below.

9.2.1 Argumentation for Controller Choice and Tuning of Controller Gains

The purpose of the position controller is to be able to track a sinusoidal reference with a frequency of 0.1 Hz and it is not considered an issue if it does not have tracking capabilities below 2% of full stroke, corresponding to 8 mm.
The FDD scheme is designed to mimic the implemented controller, i.e. it is trained in Closed-Loop. Therefore, the design of the controller is irrelevant as long as the algorithm is presented with data generated with the actual controller.

The control may therefore be realised by a P-controller designed, so that the system is stable at the worst case linearisation point. This is ensured by tuning the controller gain to give an Open-Loop phase margin (PM) of atleast 45° and gain margin (GM) of atleast 6 dB by using the linear model. This results in some robustness towards disturbances or changes of system parameters.
Furthermore, it is irrelevant to design a controller with a bandwidth above that of the servo valve (60 Hz), and since the position servo is initially only required to track a sine wave

with frequency of 0.1 Hz, the controller can be designed with a Closed-Loop bandwidth approximately twice of this.

The requirements to the controller are:

(g.1) Closed-Loop bandwidth: $\omega_{bv,CL}$>0.2 Hz
(g.2) Gain margin: GM>6dB
(g.3) Phase margin: PM>45°

The tuned controller gain for the linearisation point, can be seen in tab. 9.1, and the chosen point is explained in app. A. The frequency response of the compensated system and the designed controller can be seen in fig. 9.4.

Controller Parameters			**Compensated System**		
x_{P0} [m]	$\dot{x}_{P0}$ [m/s]	K_p [-]	GM [dB]	PM [°]	$\omega_{bw,CL}$ [Hz]
0.15	0.03	40	23	89	9.5

Table 9.1: Position controller values and resulting gain margin, phase margin and Closed-Loop bandwidth at the linearisation point.

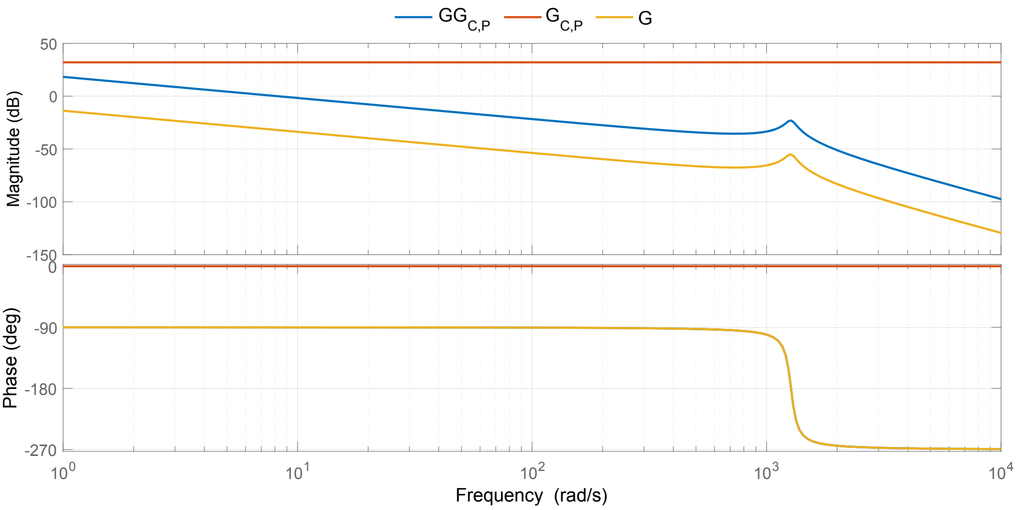


Figure 9.4: Open-Loop frequency response of the test actuator, with and without the designed controller.

From fig. 9.4 it is observed that the gain of the position servo has been increased and that the compensated and uncompensated resonance peak occurs at equal frequencies with the same magnitude.

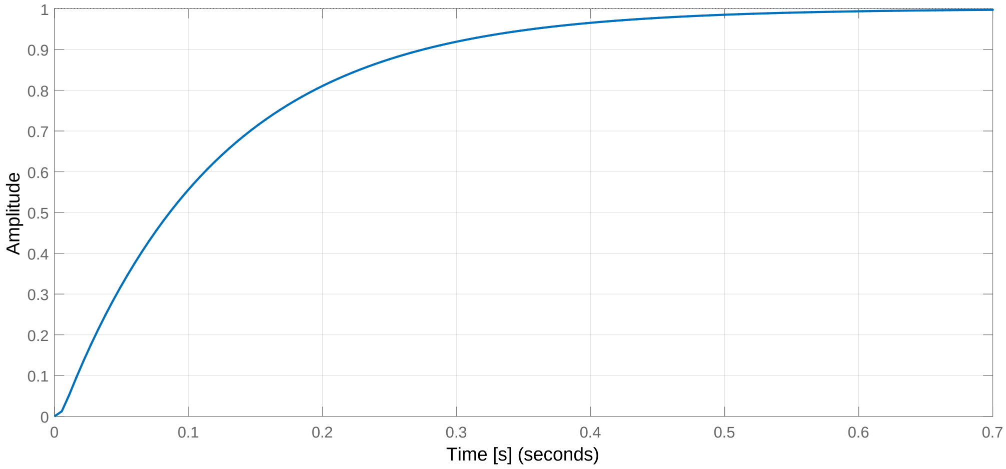


Figure 9.5: Closed-Loop step response of the compensated test actuator.

The simulated position tracking performance can be found in app. D.

9.3 Force Control of Load Actuator with Velocity Feedforward

The two hydraulic actuators are both mounted on the same slider. Therefore it is proposed to feedforward the desired velocity of the test actuator to the load actuator. This will cause the load actuator to become a slave to the test actuator which acts as the master.
The desired load force will be added by manipulating the load actuator control signal additionally.

The generated force from the system (estimated from pressure measurements) is used as a feedback and a controller is designed to track the desired force. The general structure of the control system can be seen in fig. 9.6.

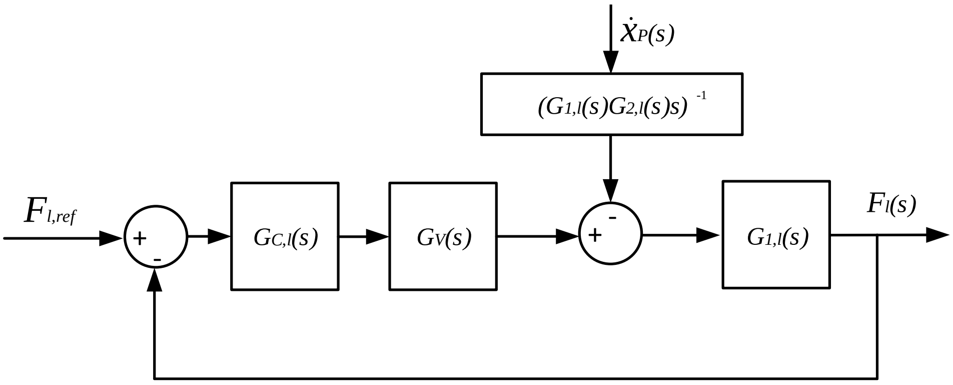


Figure 9.6: Block diagram of load side force servo control with velocity feedforward from the test side.

The TFs from fig. 9.6 which are not previously described can be defined by (9.5), (9.7) & (9.6).

$$G_{C,l}(s) = \frac{u_{V,l}(s)}{e_l(s)} = K_p \frac{(1+z_0 s)}{(1+p_0 s)} \tag{9.5}$$

$$G_{1,l}(s) = \frac{F_l(s)}{x_{V,l}(s)} = \frac{A_{A,l} k_{Q,l}(ms+B)}{ms^2 + Bs - k_{QP,l}ms - k_{QP,l}B + A_{A,l}A_{P,l}} \tag{9.6}$$

$$G_{2,l}(s) = \frac{x_P(s)}{F_l(s)} = \frac{1}{(ms+B)s} \tag{9.7}$$

$$(G_{1,l}(s)G_{2,l}(s)s)^{-1} = \left(\frac{\dot{x}_P(s)}{F_l(s)}\frac{F_l(s)}{x_{V,l}(s)}\right)^{-1} = \frac{x_{V,l}(s)}{\dot{x}_P(s)}$$

Theoretically the inverse of (9.6) can be used to decouple the two actuators, by compensating the valve displacement in the load actuator to follow the velocity of the test actuator. This cannot be implemented in reality since the inverse of $G_{1,l}(s)G_{2,l}(s)s$ is a non-proper TF and furthermore, the valve displacement cannot be directly manipulated.

Therefore, to approximate the inverse of the TF and to manipulate the control signal ($u_{V,l}$) instead of the actual valve displacement ($x_{V,l}$) the TF in (9.8) is defined and the DC-gain of it is given by (9.9). The modified structure is illustrated in fig. 9.7.

$$G_{V,l}(s) = \left(sG_V(s)G_{1,l}(s)G_{2,l}(s)\right)^{-1} \tag{9.8}$$

$$H_l = G_{V,l}(0) \tag{9.9}$$

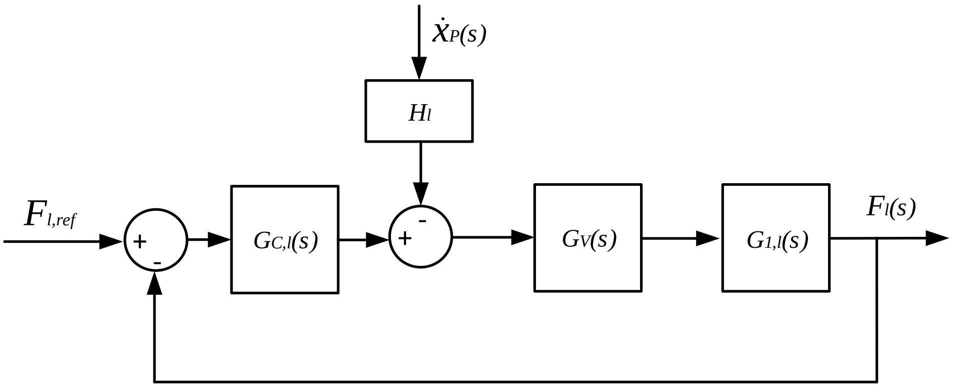


Figure 9.7: Modified block diagram of load side force servo control.

Determination of the desired velocity can be obtained from the defined position reference, if the signal is continuous and hence differentiable. If this cannot be guaranteed, i.e. in the case of a step input, another method must be applied.

For the requirements of the force controller in this project it is sufficient to apply the DC-gain of the test actuator velocity TF as described by (9.11).

$$G_{V,t}(s) = sG_V(s)G_{1,t}(s)G_{2,t}(s) \tag{9.10}$$

$$H_t = G_{V,t}(0) \tag{9.11}$$

This is combined with (9.9) thereby a gain (Γ) which is a function of the moving direction is defined as seen in (9.12).

$$\Gamma = \begin{cases} H_{+,t}H_{+,l} = 0.2331 & \text{for } u_{V,t} \geq 0 \\ H_{-,t}H_{-,l} = 0.2593 & \text{for } u_{V,t} < 0 \end{cases} \tag{9.12}$$

In (9.12) the + & - subscript defines if positive or negative valve reference is used during the derivation of the TF's. The specific gains can be found in app. A along with the different linearisation points which are shown in sec. A.1.

9.3.1 Argumentation for Controller Choice and Tuning of Controller Gains

A lag controller has been chosen since it is desired to reduce the gain and phase around the resonance peak. In this manner the desired gain margin may be achieved and the control design effort is minimal. During the design the following requirements for the controller have been considered:

(h.1) Closed-Loop bandwidth: $\omega_{bv,CL}$>15 Hz
(h.2) Rise time: τ_{CL}<0.1 s
(h.3) Gain margin: GM>6dB
(h.4) Phase margin: PM>45°

The required Closed-Loop bandwidth and rise time are a consequence of the designed load trajectory presented in sec. 8.2.3. This was designed with a 0.16 Hz sine wave and pseudorandom steps of 10 Hz with a time constant of 0.1 seconds (time until 63.2% of the step is reached). The design of the lag compensator fulfilling this is given in tab. 9.2.

Controller Parameters					**Compensated System**			
$x_{P0}[m]$	$\dot{x}_{P0}[m/s]$	K_p [-]	z_0 [-]	p_0 [-]	GM [dB]	PM [°]	$\omega_{bw,CL}$ [Hz]	τ_{CL} [ms]
0.25	0.03	5e-6	0.0027	0.027	47	∞	63	4.5

Table 9.2: Force controller values and resulting gain margin, phase margin, Closed-Loop bandwidth and rise time at the linearisation point.

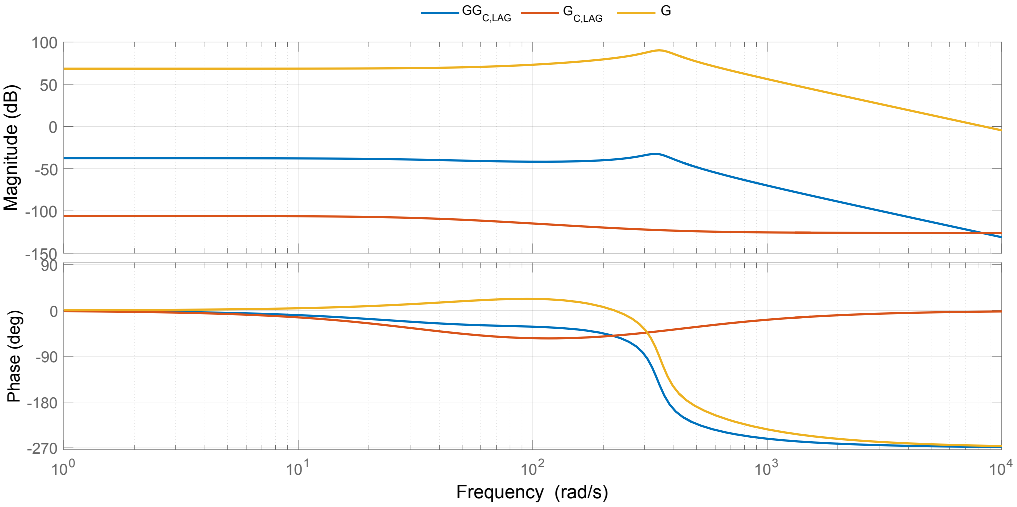


Figure 9.8: Open-Loop frequency response of the load actuator, with and without the designed controller.

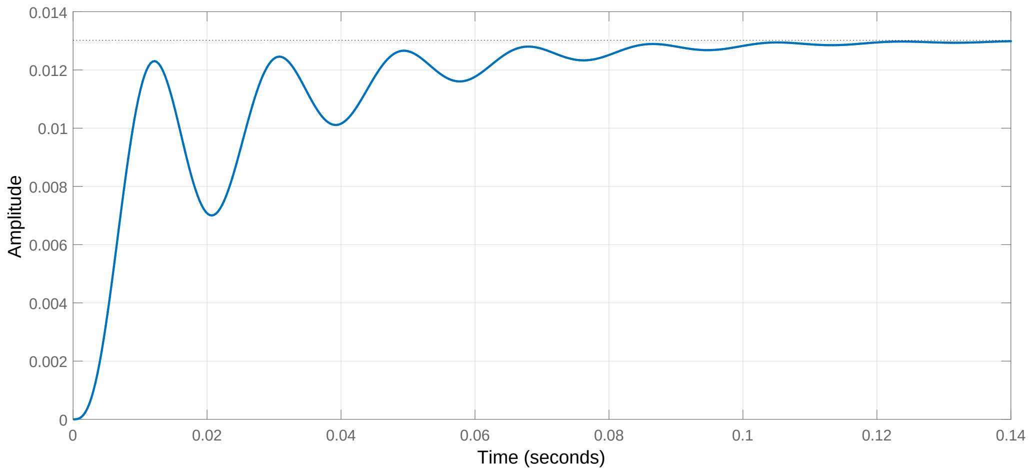


Figure 9.9: Closed-Loop step response of the compensated load actuator.

The simulated force tracking performance can be found in app. D.

9.4 The Governing Control Equations

The presented controllers can be combined in a single system diagram with the decoupling as shown in fig. 9.10.

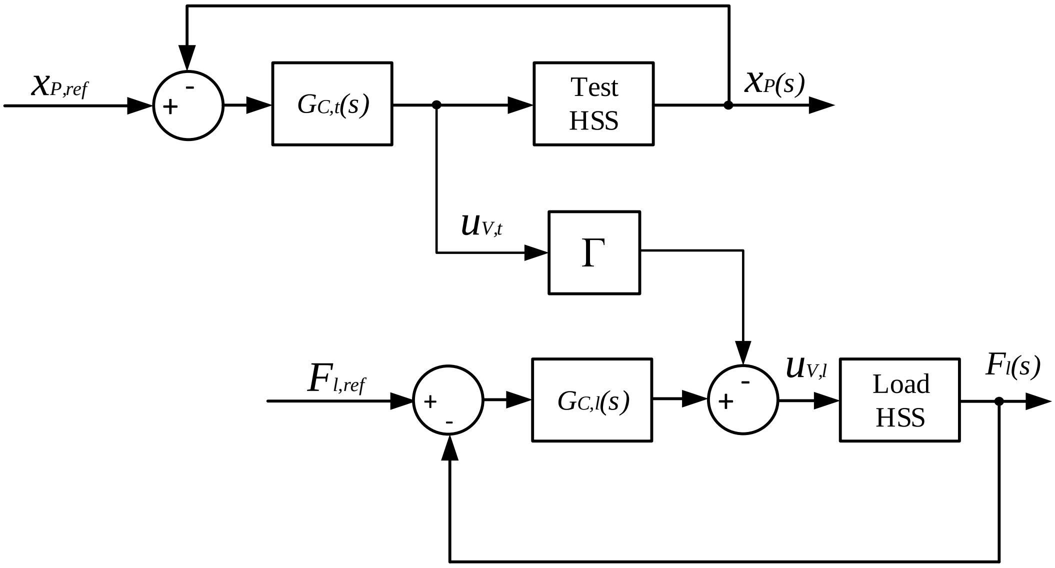


Figure 9.10: Final structure of the control strategy.

The system TF's changes as a function of direction of piston movement. Therefore the varying gain, Γ, is introduced to accommodate for this.

The test and load side valve control signals can then be defined as (9.13) & (9.14).

$$u_{V,t} = G_{C,t}(s)e_t(s) \tag{9.13}$$

$$u_{V,l} = G_{C,l}(s)e_l(s) - \Gamma u_{V,t} \tag{9.14}$$

The theoretical performance for both the position- and force controller with and without feedforward can be found in app. D.

9.5 Part Conclusion

A position and force controller have been designed for the test and load actuators respectively. These were designed at the linearisation points corresponding to the lowest eigenfrequency in order to ensure stability in the expected working area.
The controllers have been proved input/output stable at the chosen linearisation point, and the design criteria have been met. This was done by evaluating the gain margin, phase margin, closed-loop bandwidth and rise time of the two separated compensated systems.
The choice of controller was based on finding the simplest solution to the control problem, while still meeting the system requirements.

The simulated tracking performance is illustrated in app. D, where the improvement caused by the feedforward gain is obvious.

CHAPTER 10

Implementation of Supervisory Control and Data Acquisition System

Contents

This chapter describes the implementation and functions of the Supervisory Control and Data Acquisition (SCADA) system, which will be used on the laboratory set-up.

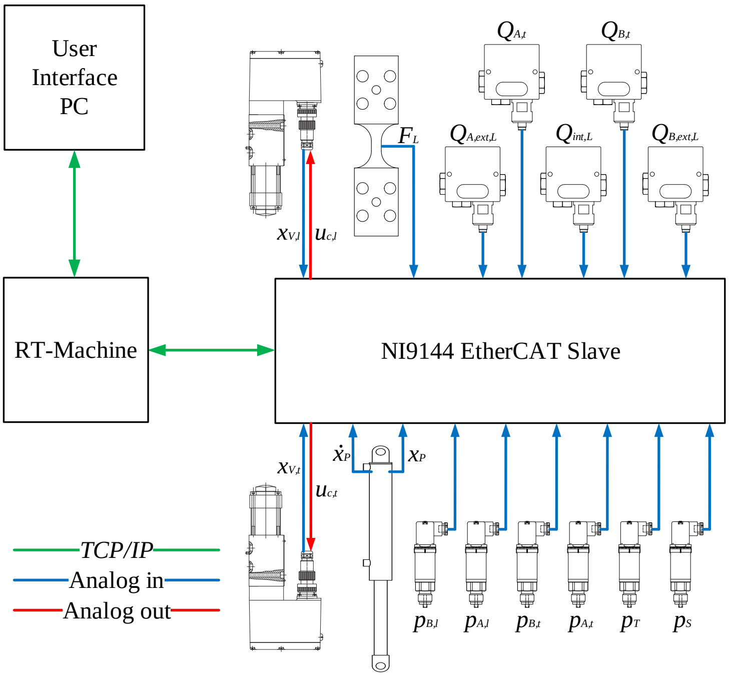


Figure 10.1: Communication of hardware used on the set-up, where t & l denotes test and load side respectively.

The SCADA system will be used to collect data for verification of the mathematical model, and of the proposed fault detection and diagnosis scheme. The SCADA system is designed as an EtherCAT master/slave Ethernet fieldbus system. The master unit is a LabVIEW Real-Time Desktop PC (RT-Machine) which utilises deterministic software/Digital based control. The slave unit is a National Instruments (NI) *NI9144* 8-Slot EtherCAT Chassis, which will hold different NI C Series I/O Modules, where for this thesis an analogue input, an analogue output, and a digital I/O have been utilised. A functional connection diagram of the measurement system can be seen in fig. 10.1. The implemented control algorithm runs on the RT-Machine which is connected to the EtherCAT slave. This holds the Data Acquisition Cards (DAQ) from which measured inputs may be read and output signals may be written. All commands and set-point changes to the RT-Machine are send from the user interface computer (UI PC), where real time is not needed.

The following section contains a description of the implemented software, how the RT-Machine operates, and how the system is controlled from the UI PC.

10.1 Architecture of Implemented Software

The developed software implemented on the RT-Machine, is a customised version of *LabVIEW Real-Time Control (NI-DAQmx)* template provided by National Instruments. The overall architecture of the different loops, communication, data logging and system health monitoring can be seen in fig. 10.2.

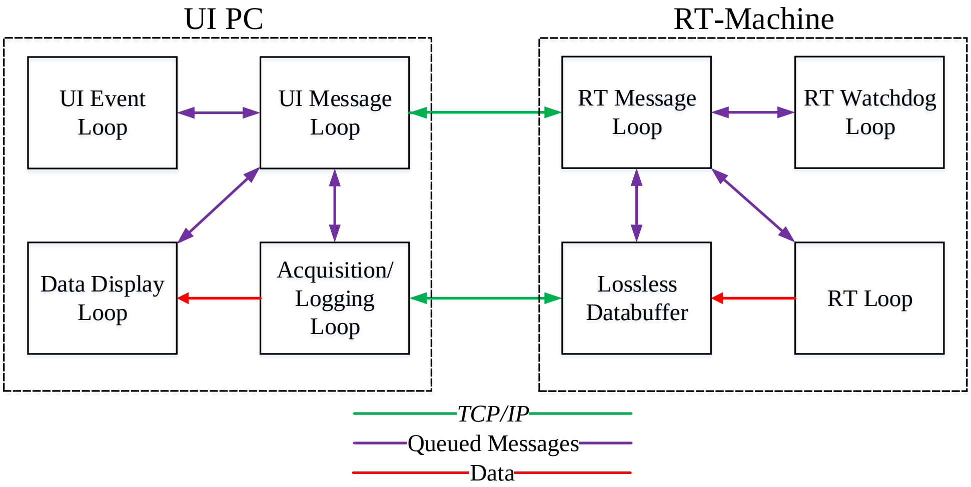


Figure 10.2: Top level software architecture of the implemented SCADA system.

The following subsections will explain the different parallel loops on both the UI PC, and the RT-Machine.

10.1.1 User Interface PC

The first layer of the developed software is the UI PC. The UI PC is a Windows based PC, which runs a standard version of LabVIEW. Here the user/operator can set different control variables, give commands to the RT-Machine, activate the data-logger and survey the systems health.

All the parallel loops are connected through a message queue, which can transmit commands between the different loops. This feature removes the need for interrupting the loops, when a command is send. Instead the message is introduced when the loop has completed the iteration, and is preparing the next iteration. The UI PC software architecture consists of four parallel loops, which can be described as the following.

UI Event Loop

The Event Loop surveys all user activity, by checking if one or more of the user input variables have changed value or state. If a change has been made, a message is created and send to the UI Message Loop along with the new user input data, as a package. The created message acts as a header for the package. Every user event will have a unique message, making the user input data distinguishable.

UI Message Loop

The Message Loop receives all messages send from all the loops running on the UI PC along with all messages send from the RT-Machine. If a message is received from the UI Event Loop, the message header is interpreted and sorted. Next it is pushed into the message queue to the local or remote receiver, along with the data attached to it. The use of the queued message approach allows the UI PC to use a high level of data integrity, but still have a low influence on the parallel execution of the other loops

UI Acquisition/Logging Loop

The Acquisition/Logging Loop reads all the data from a First In First Out (FIFO) data buffer send from the RT-Machine. The data is read in chunks, and send to the UI Data Display Loop. If a message containing a data logging command, along with a user defined logging interval, the received data is also saved to a file. The use of a FIFO buffer allows for lossless data displaying and data logging, as long as the Acquisition/Logging Loop reads the data from the FIFO buffer faster than the RT-Machine is pushing data to the buffer.

UI Data Display Loop

The Data Display Loop receives data from the Acquisition Loop, and displays it to the user. The loop is a low priority loop, and therefore only updates every 200 ms to save CPU time.

Figures of the UI layout, and operation guides can be found in app. G

10.1.2 RT-Machine

The RT-Machine consists of several parallel loops, where the general structure can be seen in fig. 10.2. The different parallel loops are clocked at different frequencies, and have different prioritising, where the RT Loop is considered a high priority loop, and the different user interacting loops are of lower priority. The following description will cover the main features of the different loops and features implemented on the RT-Machine.

RT Message Loop

The RT Message Loop operates in the same manner, as the UI Message Loop. All messages received from the UI PC or local loops on the RT-Machine are sorted, and forwarded to the given process responsible of the task described in the message.

RT Loop

The RT Loop is the main loop of the program which is implemented to have a fixed

frequency of 1 kHz. This means, that each iteration is timed to 0.001 s. If the RT-Machine cannot complete one iteration within the time limit, a finished late error is triggered. The sequence which the RT loops has to complete in each iteration can be explained as the following:
The measurements are sampled, and the raw data collected is converted into SI units. Since the measurements may by subjected to noise, digital Butterworth filters are implemented and may be used by the SCADA software, if the user finds it necessary. The data is checked up against predefined hardware protection values, and if the data exceeds one or more thresholds, an error is triggered. If no error is trigged, the data is fed into a state machine, where it may be used for control purposes. In fig. 10.3 a visualisation of the designed state machine can be seen.

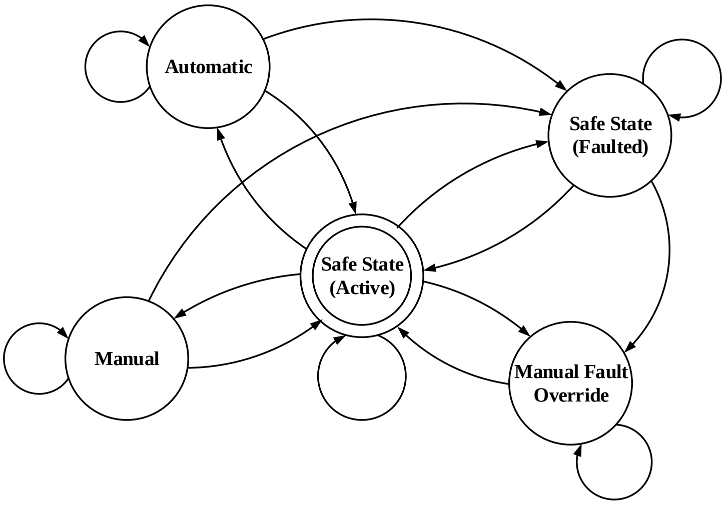


Figure 10.3: Visualisation of the implemented state machine.

The state machine is partly controlled by the user and partly by the program it terms of safety. The different states can be described as:

Safe State (Active)

The Safe State (Active) is the start up state (default state), which is the state that will be active when the UI PC is connected to the RT-Machine, and no errors have been triggered. This state ensures that so called *Safe Output Values* are generated, which for the two outputs to the MOOG valves are 0V. This state can only be active if no errors have been triggered, hence the state can be changed directly by the UI PC.

Safe State (Faulted)

The Safe State (Faulted) is activated if the RT-Machine encounters an error. This is done immediately and automatically by the RT-Machine to prevent damage to the hardware and test rig. The state will be locked until the error has been eliminated or the Manual Fault Override state is activated by the user. The *Safe Output Values* are also used in this state, ensuring a fast cut-off of the pressure lines to the system.

Manual Fault Override State

The Manual Fault Override state may be activated by the user, allowing for manual control over the system so a given fault may be isolated. However, this state only allows up to 10% of the full output.

Manual State

The Manual state may only be activated by the user, and allows for full output range. This state is designed if manual tests, or a simple movement of the cylinder piston is required.

Automatic State

The Automatic state like the Manual State may only be activated by the user. This state allows the system to be controlled by a pre-implemented feedback trajectory controller. In the case of this thesis, this mode will contain two controllers, one for controlling the position of the test cylinder, and one for controlling the load delivered by the load cylinder. In both cases the controllers are trajectory controlled, by pre-made trajectories, which are loaded on connection to the RT-Machine. The discretisation of the controllers is done by Tustins Approximation, and is explained in app. E.

RT Watchdog/Error Handling Loop

The Watchdog and Error Handling Loop is used to automatically survey the system for any triggered faults. The Error Handler part of the loop is the custom made limit checking, which is used for hardware and personal protection. If one or more limits are exceeded the Error Handler will force the RT-Loop state into Safe State (Faulted), and a message is send to the UI PC explaining the error. The Watchdog timer works by having the RT-Machine constantly receive confirmation that the computer is still functioning properly, hence no software fault has occurred, and that the device is still connected to the host UI PC. If the Watchdog does not receive this confirmation within a pre-set timeout period, the Safe State (Faulted) will be activated and the *Safe Output Values* written to the outputs and a message will be send to the UI PC. These safety features are necessary since the system may be used for testing experimental controllers and in the case of this thesis fault diagnosis techniques.

RT Lossless Data Buffer

The RT Lossless Data Buffer collects, and temporary stores all measurement data. Since the RT Loop samples at 1 kHz a buffer must be used to prevent data losses, when transmitting the data to the UI PC. The Lossless Data Buffer is a FIFO buffer hosted on the RT-Machine, where the size of the FIFO is designed such that the read rate by the UI Acquisition/Logging Loop can keep up with the 1 kHz write rate of the RT-Loop, hence, a buffer overflow is not encountered. The idea of the FIFO buffer is shown in fig. 10.4, where the input data is enqueued one sample per iteration by the RT Loop, and the dequeuing is done in chunks by the UI Acquisition/Logging Loop.

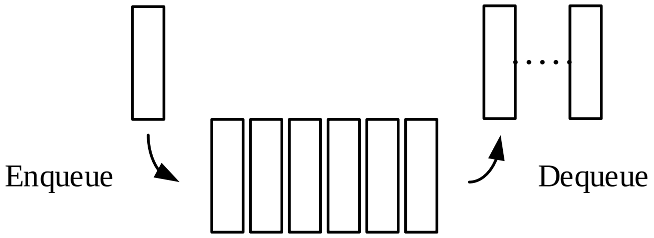


Figure 10.4: Simple FIFO buffer example.

10.1.3 Implementation of the Fault Detection and Diagnosis Scheme

The Fault Detection and Diagnosis (FDD) scheme may be implemented in several different ways. Two different approaches will be investigated in this thesis. The first is an off-line implementation, where the data is collected through the data logger and then analysed off-line by the FDD scheme. The second approach is an on-line implementation, where the FDD scheme is implemented alongside the RT state machine. Since the first method does not require additional coding, it will not be further explained, however, a short explanation of the on-line implementation is given in the following.

Data Exchange from LabVIEW to Simulink through UDP Communication

To speed up the on-line FDD, a User Datagram Protocol (UDP) is proposed. The UDP enables data to be exchanged from LabVIEW to Simulink with the sampling time limitations set by the operating system, hence a LabVIEW implementation of the ANN is not necessary. However, the UDP can be a risky choice if the data being exchanged is of critical character, since the protocol does not guarantee a "safe" delivery of data being send. Hence, the UDP is only recommended to send short, and non-critical messages [Instruments, 2006]. Since the FDD data being transmitted is not of critical character in a system functionality content, this is not considered a major issue, for the proof of concept. Furthermore, since the UDP does not have much communication control, no explicit connection to the receiver is necessary. The client (Simulink) must simply listen on a specified UDP port, and any data broadcast to that port is received.
In fig. 10.5 & 10.6 the implementation of the UDP communication can be seen, where both the Simulink and the LabVIEW are running on the UI PC.

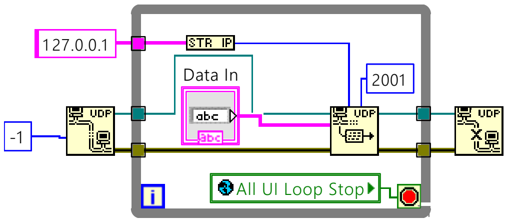


Figure 10.5: Data is prepared as a string, and send as pure ASCII characters from LabVIEW.

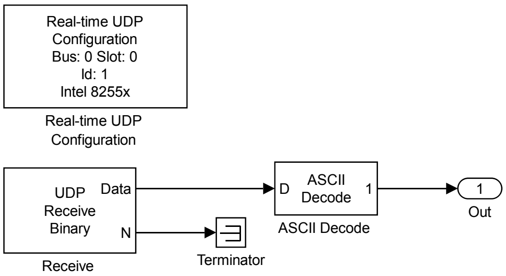


Figure 10.6: Data is received in Simulink as pure ASCII character and is converted back to floating numbers.

The loop running in LabVIEW is executed in parallel with the model in Simulink. These exchange values on each iteration, with an update period of what is allowed by Windows OS on the UI PC. The UI PC is not running in real time, hence the data will be delayed. However, this is not considered as an issue since the data send from the RT-Machine to the UI PC through the data buffer. The data buffer is forwarding chunks to Simulink, where these are unpacked and diagnosed.
The *Data In* object in fig. 10.5 contains all necessary measurements including: $p_{A,t}(k)$,$p_{B,t}(k)$,$x_P(k)$,$\dot{x}_P(k)$,$x_V(k$
This approach is chosen since it reduces the time spend on re-formulating the FDD scheme to other programming environments. However, future work could be addressed towards developing methods to exchange data from LabVIEW to Simulink faster, and with a guarantee of data integrity.

10.2 Part Conclusion

Software for supervisory control and data acquisition has been developed and compiled to the hardware on the test set-up. This serves the main purpose of doing data acquisition from the transducers and controlling the servo valves. Furthermore, safety and condition monitoring considerations have been presented, and incorporated in the architecture of the software, which will ensure that the system operates safely and as intended.
A method for communicating between Simulink and LabVIEW has been proposed, such that the FDD scheme can be executed in the Simulink environment, while receiving real system measurements from the LabVIEW environment. Future work in developing a more safe and faster data exchange method was proposed.

Part IV

Experimental Results & Conclusions

CHAPTER 11

Hydraulic Test Rig & Experimental Tests

Contents

In the following chapter the test rig will be explained, along with a number of experimental tests, used to verify the mathematical model from chap. 5. The main purpose is to investigate if the model is sufficient as state estimator in the developed model-based FDD scheme.
All the applied raw data files are appended along with the scripts designed for data acquisitions and data processing.

11.1 The Hydraulic Test Rig Design

A cross view of the mechanical test rig is illustrated in fig. 11.1 and the hydraulic diagram is depicted in fig. 11.2.

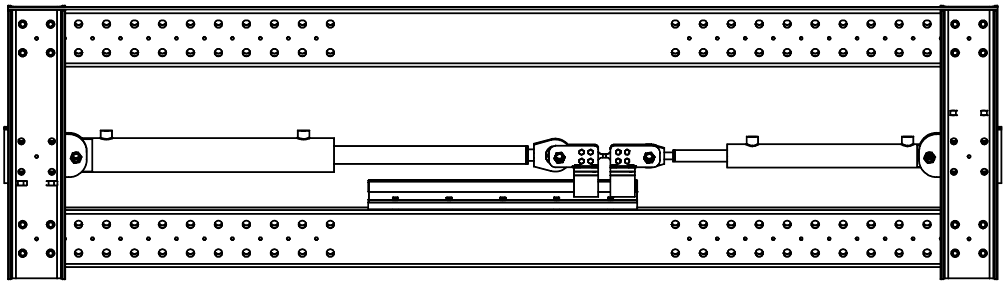


Figure 11.1: Cross view of the mechanical structure of the test rig.

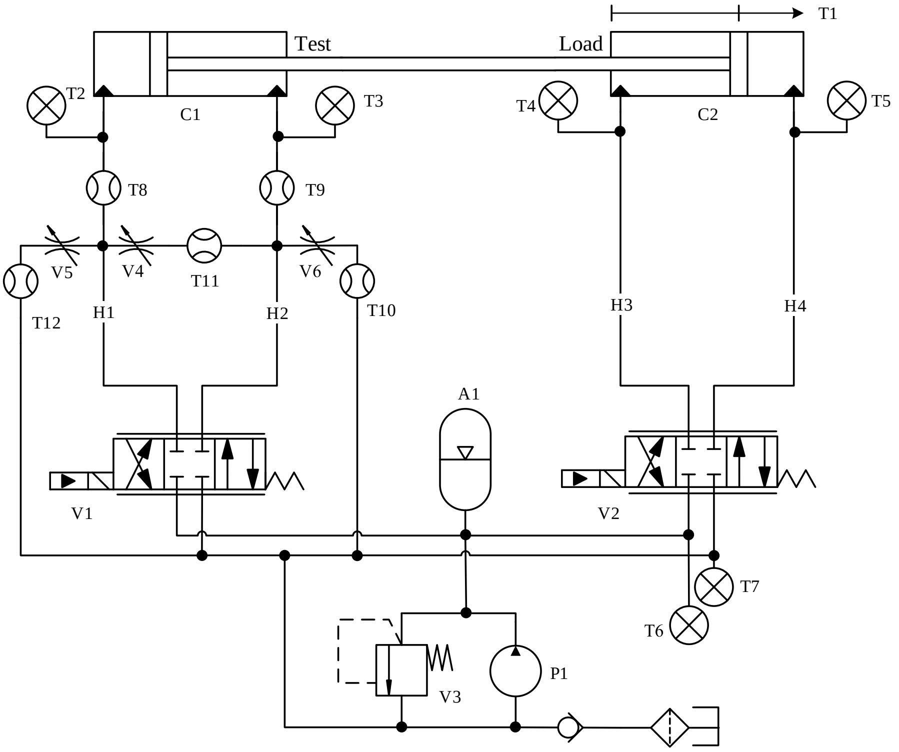


Figure 11.2: Hydraulic diagram showing the interconnection of components and transducers in the test rig.

Symbol	Component	Type	Comment
A1	Accumulator	Bosch	4 l.
H1 & H2	Hoses	SEMPERIT DIN EN 856 4SN	1.6m & 1.6 m. $d = 1/2"$.
H3 & H4	Hoses	SAIAG DIN EN 852 2SN	1.4m & 1.0 m. $d = 3/8"$.
V1 & V2	Servo valves	MOOG D633-313	Input:±10VDC. Output actual spool position 4-20 mA. Supply 24VDC.
V3	Pressure relief valve	Unspecified	Pressure limit of 300 bar
V4-V6	Bleed valves	FT257/2-38	Manually adjusted opening areas
C1	Hydraulic test-cylinder	LJM NH30-S-D-80/40xL-500-S	Maximum pressure 250 bar.
C2	Hydraulic load-cylinder	Hydra Tech 8054202-40/25x400 Regal CC	Maximum pressure 250 bar.
T1	Magnetostrictive, Absolute, Non-contact Linear-Position Sensor	Temposonics R-Series Model RP A41	Repeatability ±0.001% full stroke. Velocity deviation <0.5%. Supply 24VDC. Output 4-20 mA.
T2-T7	Pressure transducers	Danfoss MBS 33 060G2199	250 bar. Output 4-20mA. Supply 10-30VDC.
T2,T3,T6 & T7	Pressure transducers	Danfoss MBS 33-060G2199	250 bar. Output 4-20mA. Supply 10-30VDC.
T4& T5	Pressure transducers	Danfoss MBS 32-3615-1AB06	400 bar. Output 0-10VDC. Supply 15-30VDC.
T8 & T9	Flow sensors	Parker SCQ-060-0-02	±60 l/min (±3VDC). Supply 18-30VDC.
T10-T12	Flow sensors	Parker K-SCVF-015-10-07	±15 l/min (20mA). Supply 18-30VDC.
P1	Pump station	Serman & Tipsmark	Supply pressure of 120 bar.

Table 11.1: Applied components and transducers on the test rig, data sheets are appended.

Two opposing hydraulic actuators are applied in the test rig, one producing a load while the other tracks a position reference. The actuators are controlled independently by servo valves, which receives control signals from a real-time PC through LabVIEW. The experiments of interest require measurements of chamber, supply & tank pressures, piston & valve positions along with the velocity of the piston. The control and acquisition procedure is explained in chap. 10.

Since it is desired to emulate internal and external leakage, the test rig has been equipped with bleed-valves as can be seen in fig. 11.2.

11.2 Estimation of Friction Parameters

In the following section, the tests carried out for estimation of the friction characteristic are presented.

Purpose

The purpose of this test is to calculate an estimate of the actual friction parameters in the actuators on the test rig. The actuators' rods are mounted on a slider as seen in fig. 11.1. To simplify the test sequence it is decided to design a friction model which include the combined system friction.

Methods

The test rig with the forces of relevance for friction parameter estimation is shown in fig. 11.3.

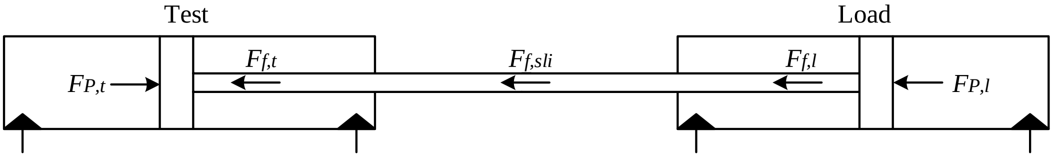


Figure 11.3: Free-body-diagram of the forces acting on the moving parts.

The equation of motion with regard to the notation shown in fig. 11.3 is given by (11.1).

$$\ddot{x}_P m = F_{P,t} - F_{P,l} - \underbrace{(F_{f,l} + F_{f,t} + F_{f,sli})}_{F_f} \tag{11.1}$$

The notations t, l & sli describe test, load and slider respectively.

The actuators are physically coupled (equal velocity) and when they are operated at constant velocity the acceleration force is zero. The combined friction in the system can therefore be expressed by reformulating (11.1) to (11.2).

$$F_f = (p_{A,t}A_{A,t} - p_{B,t}A_{B,t}) - (p_{A,l}A_{A,l} - p_{B,l}A_{B,l}) \tag{11.2}$$

The expression for the modelled friction is given by (11.3) and by adjusting these parameters the friction can be approximated. This friction model is a rough estimate since the real friction may be both pressure-, temperature-, position- & velocity dependent.

$$F_f(\dot{x}_P) = \begin{cases} \tanh(k_{tan}\dot{x}_P)\left(F_{c,p} + (F_{s,p} - F_{c,p})e^{-\frac{|\dot{x}_P|}{c_{s,p}}}\right) + b_{c,p}\dot{x}_P & \text{for } \dot{x}_P \geq 0 \\ \tanh(k_{tan}\dot{x}_P)\left(F_{c,n} + (F_{s,n} - F_{c,n})e^{-\frac{|\dot{x}_P|}{c_{s,n}}}\right) + b_{c,n}\dot{x}_P & \text{for } \dot{x}_P < 0 \end{cases} \tag{11.3}$$

The expression allows for a friction model which depends on the moving direction to account for possible asymmetrical construction properties.

Parameter Sensitivity

It is observed from (11.2) that deviations on the pressure measurements may lead to inaccurate friction parameter estimation. Therefore, a measurement uncertainty of 0.75 bar is applied to depict the span in which the measured friction is expected to lie. This illustrates the sensitivity of the friction. The limits are defined by manipulating the pressure measurements to a worst case as shown in tab. 11.2. Further uncertainties may be introduced by inaccurate transducer calibration. However, the severity of this is unknown and cannot be included in the parameter sensitivity.

Uncertainty band	max [bar]	min [bar]
$p_{A,t}$	+0.75	-0.75
$p_{B,t}$	-0.75	+0.75
$p_{A,l}$	-0.75	+0.75
$p_{A,l}$	+0.75	-0.75

Table 11.2: Description of the applied manipulations of the pressures to obtain the worst case deviations of the measured friction force.

Experimental Procedure

Estimation of the friction parameters for the friction model requires constant velocity on the piston of the cylinder. By doing this at different velocities, the friction model can be designed by adjusting the parameters in (11.3). To achieve this, a constant input to the valve is applied and the velocity measured. During this the chamber pressures are also logged.

Each velocity is tested five times in order to validate the consistency of the friction estimate. Furthermore, if the data points show a significant coherence the estimated friction is not sensitive to measurement noise.

All tests are performed after a warm up procedure has been done. This consists of full valve opening until the piston is near endstop. This is repeated until the oil temperature is ≈55°C. Furthermore, this decreases unwanted air bubbles in the fluid which limits the sources of error.

An example of a test sequence can be found in fig. 11.4.

The velocity shown in the figure is based on flow measurement and known actuator dimensions. This results in less noise than the alternative of deriving the position measurement. The velocity magnitudes were compared for both methods and there was sufficient coherence between the two, which is why the approach with less noise is used. It is observed from the sequence that the velocity is not completely constant, which gives rise to a source of error in the friction estimation. Therefore the slope of the position is calculated, and this is used as the average velocity during each test. Furthermore, the average pressure forces during this period of approximate constant velocity is used when estimating the friction.

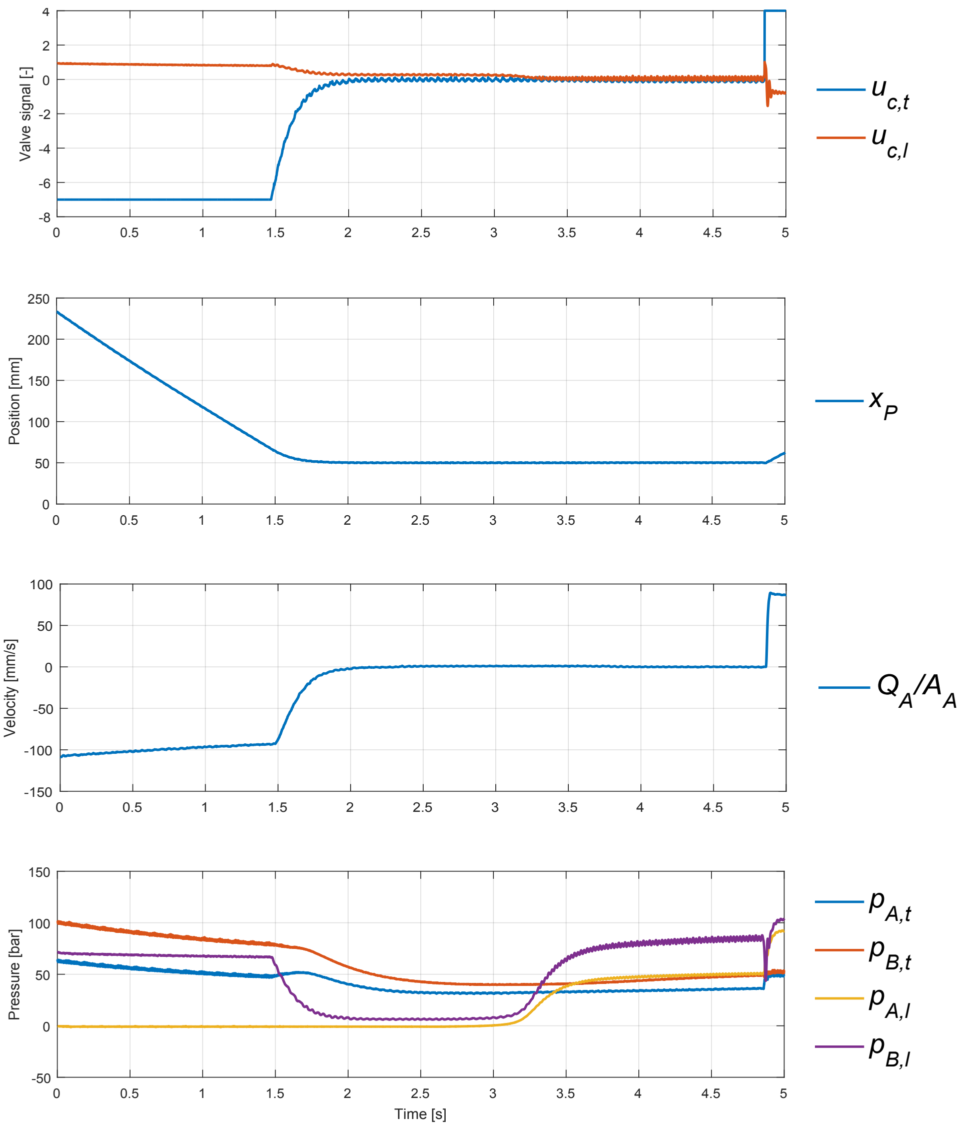


Figure 11.4: Results of the test for determination of friction.

The followed approach is considered sufficient for the purpose of this project, but a future aspect may be to implement velocity control to ensure constant velocity and thereby a more accurate friction estimation may be obtained.

Results

The measured friction force compared with the modelled is shown in fig. 11.5.

The results reveal an asymmetrical friction force but with the expected overall characteristics. The uncertainty span shows that a insufficient pressure calibration or measurement uncertainty will give rise to significant offset.

The estimated parameters describing the friction are shown in tab. 11.3.

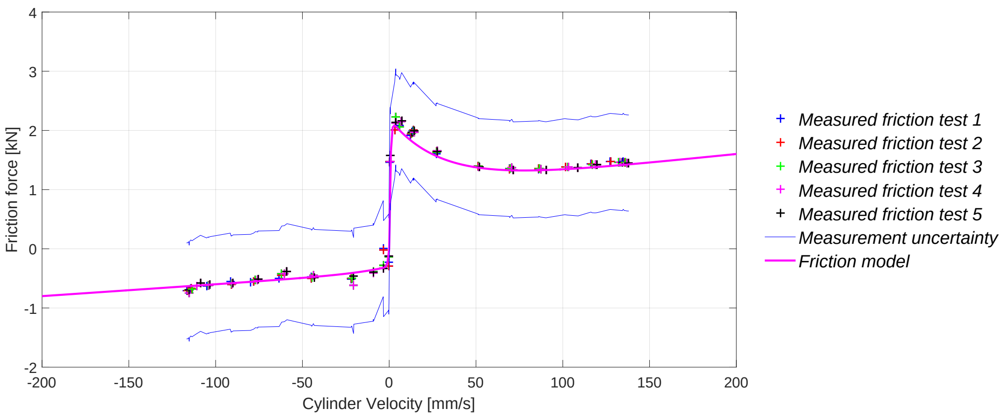


Figure 11.5: Simulated friction force compared with estimated friction from test rig.

$\hat{F}_{c,p}$[N]	$\hat{F}_{s,p}$[N]	$\hat{b}_{c,p}$ [Ns/m]	$\hat{c}_{s,p}$ [m/s]	$\hat{F}_{c,n}$[N]	$\hat{F}_{s,n}$[N]	$\hat{b}_{c,n}$ [Ns/m]	$\hat{c}_{s,n}$ [m/s]
1e3	2.2e3	3e3	30e-3	0.4e3	0.3e3	2e3	20e-3

Table 11.3: Results of the estimated friction parameters from (11.3).

Part Conclusion

An estimate of the relevant friction parameters have been found, however there does exist a large span of uncertainty on the pressure measurements. Furthermore, an offset on the calibration of each sensor will cause discrepancies between the estimated and actual friction. This calibration was performed by fitting the measured supply pressure on the set-up to the supply pressure measurement from the supply pump interface board. Hereafter each chamber from each cylinder is connected to the supply line, and the pressures in the chambers are calibrated in the same manner as the supply pressure sensor. Since the pressures have to be calibrated by another sensor, and the procedure is done manually, the calibration may be corrupted. Therefore the results does have several sources of error.

11.3 Model Verification Tests

The tests performed to verify the derived model are described in the following.

Purpose

The primary purpose of this test is to investigate if the model is sufficient as estimator for model-based FDD. This is done by validating the flows, chamber pressures, piston position and valve signal. These are chosen since they are applied in the FDD scheme and therefore need to be verified. However, the flows are solely included to verify that the measured flow is corresponding to the simulated.

Methods & Experimental Procedures

Verification of the derived model is done by applying valve input steps of various magnitudes and periods. The valve displacement, flows, pressures and piston position are measured. The simulated system is compared with the measurements and if the two are coherent, the system model is verified. This test is carried out three times and if all are similar, only one of the three is presented. This ensures that the test results are consistent.

Initially the servo valves offsets will be determined by looking at the displacement output while zero control signal is applied. Thereby the neutral position of the spring can be estimated under the assumption that the current output from the valve transducer is sufficiently accurate. This verification aids in the overall system verification.

The valve flow is investigated to validate that the valves have been modelled sufficiently. If this is the case, comparison of simulated and measured piston position is expected to show a sufficient fit, since the velocity and flow are directly correlated when not considering compression of fluid. If this is not the case, either the position measurement, flow measurements or actuator manufacturing tolerances are insufficient.

The chamber pressures in the test cylinder are verified by using the measured load actuator pressures in the model to ensure that the external load is simulated as measured. The simulated pressures are adjusted by tuning the soft parameters of the oil stiffness.

Results

The results for determination of the valve offsets are shown in fig. 11.6.

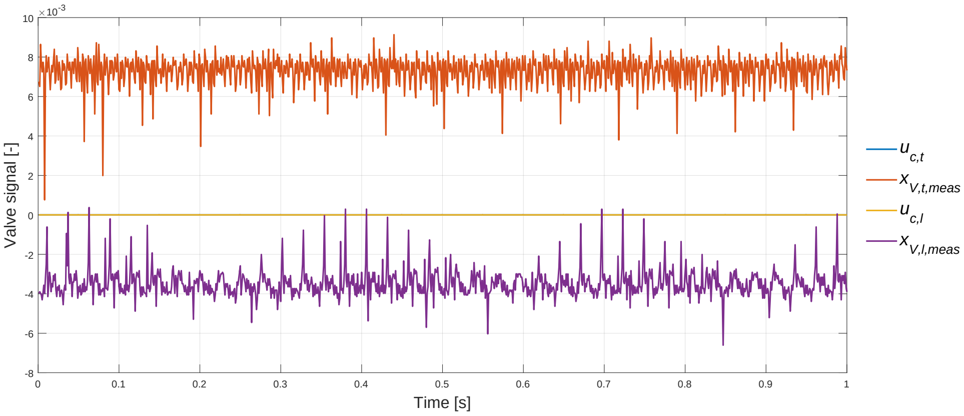


Figure 11.6: Test results to determine the neutral position of the MOOG servo valves.

From the figure it is concluded that the normalised neutral position for the two MOOG servo valves are 7.5e-3 and -3.5e-3.

The main results of both simulation and experimental test are shown in 11.7. As previously mentioned three tests have been conducted. These showed similar results which is why two of them are omitted.

The flows are simulated by applying the information from the manufacturer and as observed there is a sufficient coherence. It should be noted that there is an uncertainty on the measurement of $\approx \pm 0.3$ l/min.

The position reveals some discrepancies. Therefore the measured A-chamber flow divided by the piston area informed by the manufacturer is plotted along side, which shows that the model is accurate and measurement is inaccurate. This leaves a number of possible conclusions, either an incorrect piston area is used, the flow is calibrated insufficiently or the position

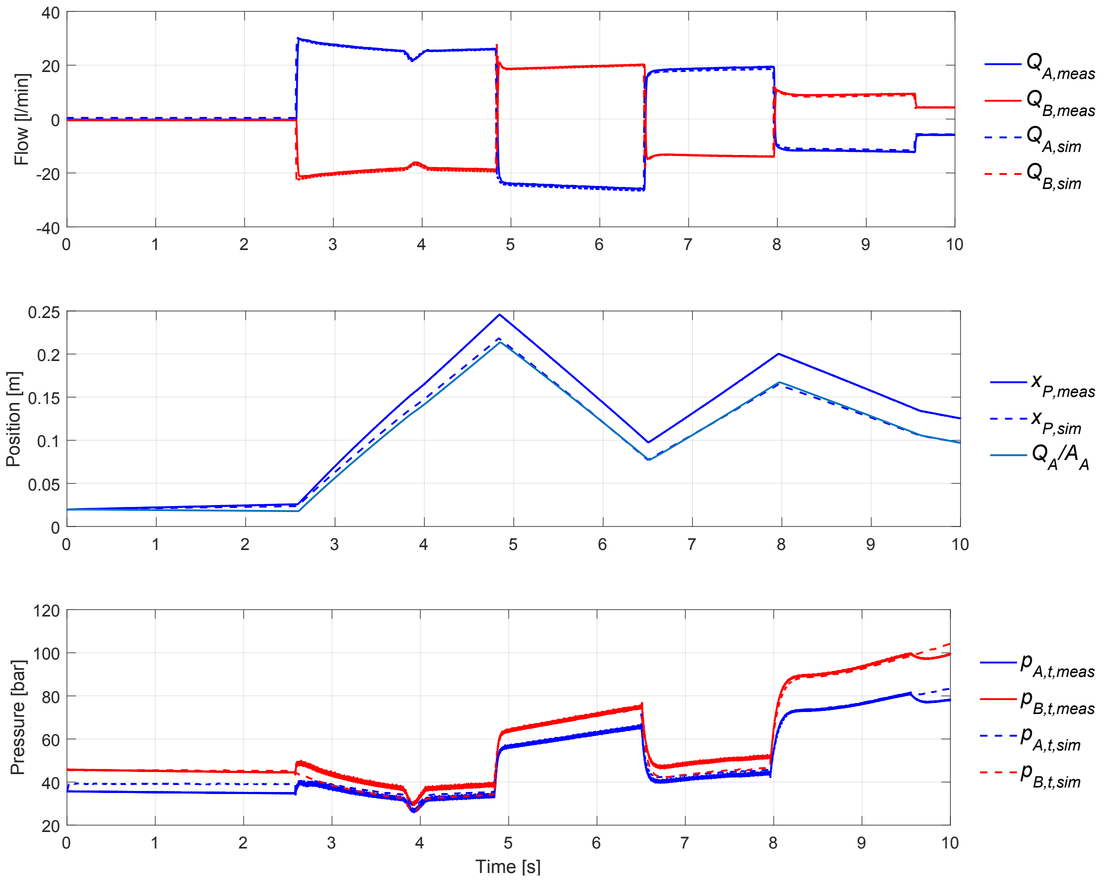


Figure 11.7: Test results and simulated counterparts for verification of the overall model dynamics.

measurement is incorrect.
The valve flow is calculated and measured with the exact specification from the manufacturer and the same is the case for the actuator dimensions. The position measurement has been checked for no- and full- actuator stroke resulting in 0 and 400 mm with a linear characteristics. The actual positions were verified with tape measurements. The actuator and valve specifications are considered to be highly accurate, and will therefore not adjusted.

The simulated movement in the positive direction is slower than anticipated from measurements. The slope in negative direction is fitting sufficiently. This tendency is similar in the chamber pressures, which are insufficient in positive direction but fits in negative direction. The amount of uncertainties caused by pressure measurements (calibration and uncertainty), friction model (simple approximation), actuator dimensions, the simulated load compared with the actual load, and flow measurement (calibration and uncertainty) makes it difficult to argue why the pressures does not fit consistently.

Conclusively it is uncertain if the mathematical model is sufficient to use for model-based FDD. If the FTDANN is able to learn the normal residual between model and measurements it is theoretically possible that it will be able to diagnose the faults accurately although the model does not represent the measurements accurately.
It is therefore proposed to proceed with both the state estimator and to train a NARX ANN with the measured data. These will be compared trough tests with the faults of interest.

Part Conclusion

A coherence between simulated and measured flows have been confirmed. The simulated and measured position showed different slopes while moving in positive direction, but accurate slope in negative direction.
The simulated and measured chamber pressures did reveal coherence in negative direction, but in positive direction the magnitudes did not fit. The issue was considered to have several sources of error, which is why no solution could be proposed. This was also the case for the simulated piston position.

The valve offsets along with the adjustments made on the fluid stiffness is shown in tab. 11.4. The bulk modulus and air in the fluid have been manually tuned to estimate the pressure build-up from the measurements.

Soft Parameters	$\hat{\beta}_{oil}$ [bar]	$\hat{\epsilon}_{A,0}$ [%]	$\hat{u}_{c,t,offset}$ [-]	$\hat{u}_{c,l,offset}$ [-]
Preliminary	6.5e3	0.38	0	0
Adjusted	13e3	1	7.5e-3	-3.5e-3

Table 11.4: The adjustments of the soft parameters applied in the model. Performed to fit the measurements.

CHAPTER 12

Fault Detection and Diagnosis Tests

Contents

In the following chapter evaluation tests of the developed FDD scheme on the physical test rig are described. The FDD scheme of interest has been presented and explained in chap. 8. A recap on the FDD scheme utilised is shown in fig. 12.1.

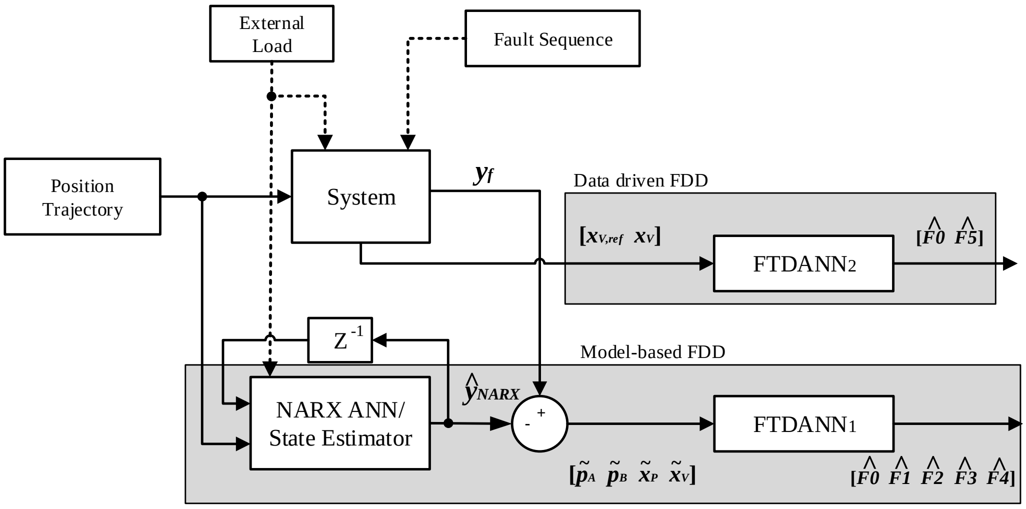


Figure 12.1: The FDD scheme with estimated values denoted with a hat, i.e. $\hat{y}$ and the difference between estimate and actual values are denoted with a tilde, i.e. $\tilde{p}_A$.

Fig. 8.2 shows that the generation of a residual is either originating by using a classical state estimator or by a NARX ANN. This has been chosen since the verification of the model was problematic at some situation as explained in 11.3.

12.1 Purpose

The purpose of the tests are to verify the developed FDD schemes capability to estimate an actual leakage fault, stuck valve or pressure transducer failure, which are emulated on a physical test set-up.

12.2 Methods

By emulating the faults of interest on the test set-up it is possible to obtain data series that can be used to train and validate the developed FDD scheme. The approach will be similar to the one described in sec. 8.3.
It was initially proposed to apply the ANNs trained from modelled data, with the purpose of having an efficient approach for FDD development. However, the model verification revealed that there were some discrepancies between measured and simulated data, which is why the data has been collected experimentally, and the ANNs retrained.

The input and output regressors applied for training can be described as shown in (12.1)-(12.6), all variables are measured values from the test set-up, with exception of $\{x_{P,ref}\}$ which is the input reference. The measurable values have all been filtered with a 6^{th}. order Butterworth filter, with a cut-off frequency of 50 Hz, the filter design is explained in app. F.

$$\mathbf{U}_{\text{NARX}} = \begin{bmatrix} \{x_{P,ref}\}_1 & \{x_{P,ref}\}_2 & \cdots & \{x_{P,ref}\}_m \end{bmatrix} \tag{12.1}$$

$$\mathbf{Y}_{\text{NARX}} = \begin{bmatrix} \{p_A\}_1 & \{p_A\}_2 & \cdots & \{p_A\}_m \\ \{p_B\}_1 & \{p_B\}_2 & \cdots & \{p_B\}_m \\ \{x_P\}_1 & \{x_P\}_2 & \cdots & \{x_P\}_m \\ \{x_V\}_1 & \{x_V\}_2 & \cdots & \{x_V\}_m \end{bmatrix} \tag{12.2}$$

$$\mathbf{U}_{\text{FTDANN}_1} = \begin{bmatrix} \{\tilde{p}_A\}_{1,1} & \{\tilde{p}_A\}_{2,1} & \cdots & \{\tilde{p}_A\}_{m,1} & \{\tilde{p}_A\}_{1,2} & \{\tilde{p}_A\}_{2,2} & \cdots & \{\tilde{p}_A\}_{m,\varrho_1} \\ \{\tilde{p}_B\}_{1,1} & \{\tilde{p}_B\}_{2,1} & \cdots & \{\tilde{p}_B\}_{m,1} & \{\tilde{p}_B\}_{1,2} & \{\tilde{p}_B\}_{2,2} & \cdots & \{\tilde{p}_B\}_{m,\varrho_1} \\ \{\tilde{x}_P\}_{1,1} & \{\tilde{x}_P\}_{2,1} & \cdots & \{\tilde{x}_P\}_{m,1} & \{\tilde{x}_P\}_{1,2} & \{\tilde{x}_P\}_{2,2} & \cdots & \{\tilde{x}_P\}_{m,\varrho_1} \\ \{\tilde{x}_V\}_{1,1} & \{\tilde{x}_V\}_{2,1} & \cdots & \{\tilde{x}_V\}_{m,1} & \{\tilde{x}_V\}_{1,2} & \{\tilde{x}_V\}_{2,2} & \cdots & \{\tilde{x}_V\}_{m,\varrho_1} \end{bmatrix} \tag{12.3}$$

$$\mathbf{Y}_{\text{FTDANN}_1} = \begin{bmatrix} \{F1\}_{1,1} & \{F1\}_{2,1} & \cdots & \{F1\}_{m,1} & \{F1\}_{1,2} & \{F1\}_{2,2} & \cdots & \{F1\}_{m,\varrho_1} \\ \{F2\}_{1,1} & \{F2\}_{2,1} & \cdots & \{F2\}_{m,1} & \{F2\}_{1,2} & \{F2\}_{2,2} & \cdots & \{F2\}_{m,\varrho_1} \\ \{F3\}_{1,1} & \{F3\}_{2,1} & \cdots & \{F3\}_{m,1} & \{F3\}_{1,2} & \{F3\}_{2,2} & \cdots & \{F3\}_{m,\varrho_1} \\ \{F4\}_{1,1} & \{F4\}_{2,1} & \cdots & \{F4\}_{m,1} & \{F4\}_{1,2} & \{F4\}_{2,2} & \cdots & \{F4\}_{m,\varrho_1} \end{bmatrix} \tag{12.4}$$

$$\mathbf{U}_{\text{FTDANN}_2} = \begin{bmatrix} \{x_V\}_{1,1} & \{x_V\}_{2,1} & \cdots & \{x_V\}_{m,1} & \{x_V\}_{1,2} & \{x_V\}_{2,2} & \cdots & \{x_V\}_{m,\varrho_2} \\ \{x_{V,ref}\}_{1,1} & \{x_{V,ref}\}_{2,1} & \cdots & \{x_{V,ref}\}_{m,1} & \{x_{V,ref}\}_{1,2} & \{x_{V,ref}\}_{2,2} & \cdots & \{x_{V,ref}\}_{m,\varrho_2} \end{bmatrix} \tag{12.5}$$

$$\mathbf{Y}_{\text{FTDANN}_2} = \begin{bmatrix} \{F5\}_{1,1} & \{F5\}_{2,1} & \cdots & \{F5\}_{m,1} & \{F5\}_{1,2} & \{F5\}_{2,2} & \cdots & \{F5\}_{m,\varrho_2} \end{bmatrix} \tag{12.6}$$

where

ϱ_1	is the number of fault modes in FTDANN$_1$, 5.	[-]
ϱ_2	is the number of fault modes in FTDANN$_2$, 2.	[-]
m	is the number of test trajectories, 1.	[-]
NARX	is an index to indicate data applied for NARX network.	[-]
FTDANN$_1$	is an index to indicate data applied to diagnose the faults $F1$-$F4$.	[-]
FTDANN$_2$	is an index to indicate data applied to diagnose the fault $F5$.	[-]
$F1$	is the internal leakage flow.	[l/min]
$F2$	is the external A leakage flow.	[l/min]
$F3$	is the external B leakage flow.	[l/min]
$F4$	is the non-dimensional pressure failure.	[-]
$F5$	is the non-dimensional stuck valve failure.	[-]

The training method utilised in for the ANNs, is the same as presented in sec. 7.3.3. Furthermore, the design used, is the same as used for the ANNs presented in sec. 8.2.2. The given designs and training periods, for the ANNs used in this section can be seen in tab. 12.1.

Network	d_u	d_y	n	Training period
NARX	9	2	4	2 h 30 min
FTDANN$_1$ (NARX)	6	-	6	42 min
FTDANN$_1$ (Classical state estimator)	6	-	6	3h 41 min
FTDANN$_2$	12	-	5	9 min

Table 12.1: ANN design variables, based on the results from sec. 8.2.2

12.3 Experimental Procedures

Each test is initiated by a warm-up procedure as described earlier in sec. 11.2. This is to ensure temperature of ≈55°C of the oil, to avoid changing fluid properties as a function of temperature resulting in inconsistent results.
Subsequently all sensors are calibrated to minimise bias errors on the measurements.

A sinusoidal reference of 1.5 rad/s with 0.1 m amplitude is applied for all tests and a constant load of 2 kN is referenced to the load cylinder. The position reference frequency is therefore decreased by a factor of two and the amplitude increased by a factor of five from what was applied during the theoretical tests. This results in a longer period of time before the direction is changed, which was necessary in order to obtain accurate leakage measurements, since the applied sensors need a certain amount of flow to give consistent outputs. This issue is especially associated with the external leakage sensors, which are of the type gearwheel volume counter, where the gearwheel needs to be spinning at the same rate as the flow in order to give an accurate output.

By opening internal leakage valve V4 connecting chamber A and B in the test cylinder to a given opening area, the internal leakage fault can be emulated. 100 seconds will be logged for each test sequence. The valve configuration can be seen in fig. 11.2.
The same procedure is followed with external leakage valve V5 & V6 and furthermore, a test with all the afore mentioned valves closed is performed to have data for the system without a fault.

The pressure transducer failure is emulated by manipulating the measured pressure output and the stuck valve is tested by manipulating the reference signal within the control software.

The proposed test sequence allows for a similar data structure as presented in sec. 8.2.

A single continuous sequence containing the faults of interest is proposed to evaluate the FDD scheme's performance. The fault outputs are similar to previous faults and the generalisation capability will therefore be visualised.
Ultimately the described test will represent an actual diagnosis situation where the system changes from a healthy- to a faulty- state. If this can be diagnosed the scheme is considered sufficient for its purpose.

12.4 Results

The main results of both fault measurements and estimations are shown in fig. 12.2 representing the FDD scheme with a classical state estimator. The faults are introduced one at a time and applied for a 100 second period.
An EMA of ten seconds has been applied for all data series to decrease oscillations, thereby enhancing the visibility of each fault estimation. It should be noted that the measured leakages without EMA applied oscillates from 0.3-1.2 l/min.

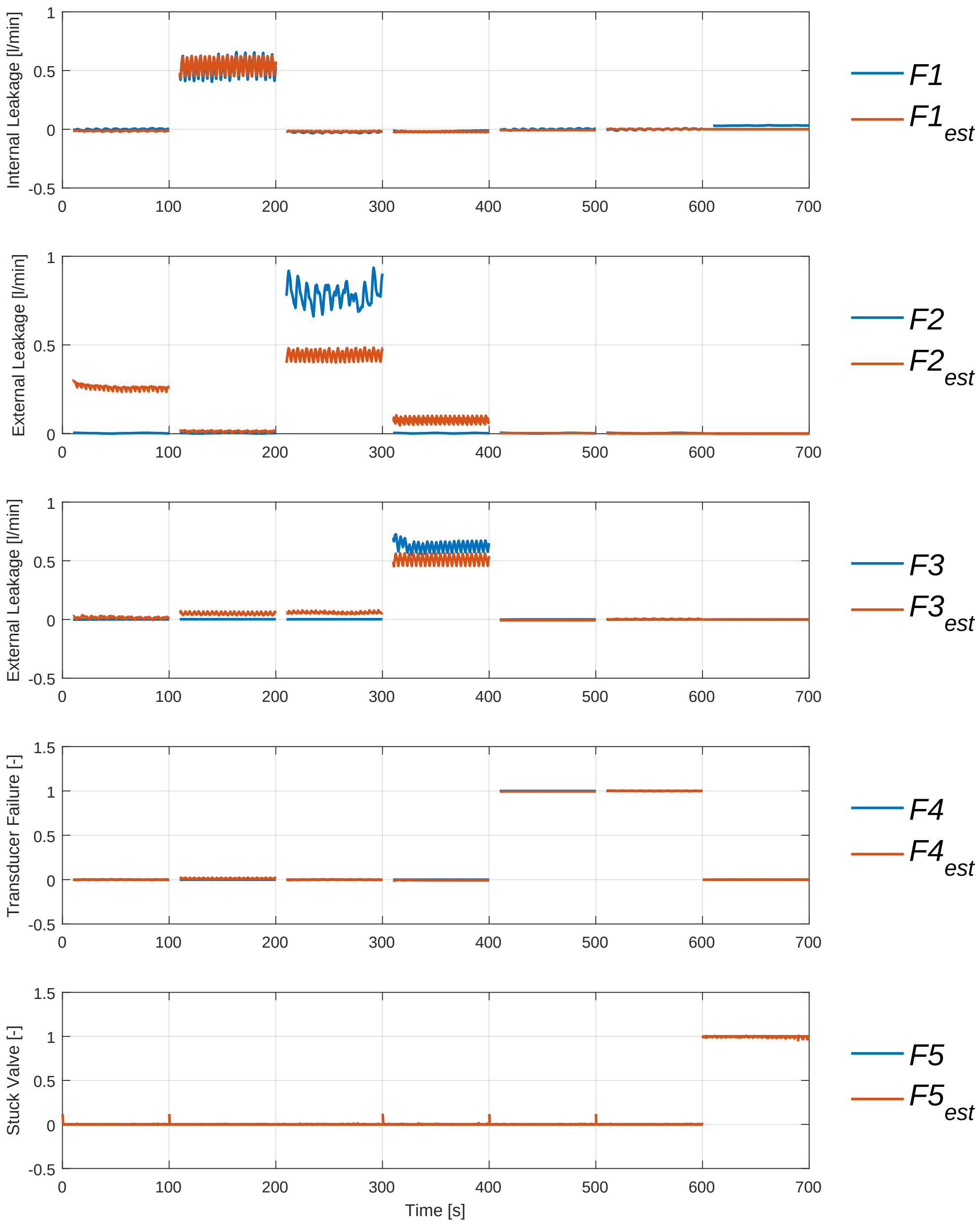


Figure 12.2: Experimental results of the FDD scheme with classical state estimator compared with the measured faults. Every 100 second represents a given fault in ascending order from $F1$-$F5$ with $F4$ being represented twice to emulate 1 & 250 bar pressure transducer output.

The results from fig. 12.2 shows consistent fault estimation in all situations except $F2$ & $F3$. This is a decrease of performance compared with the theoretical tests. The main sources of error are considered to be the detected issues with the modelling of pressures in sec. 11.3 and the possible bias from uncertainty and calibration of the sensors that may be present in the measurements.
Therefore it is suggested for future work to investigate the validity of the measurements, which may be used to improve the model and ultimately the FDD scheme.

The results when the NARX ANN is utilised as estimator, and a FTDANN as diagnose scheme are shown in fig. 12.3.

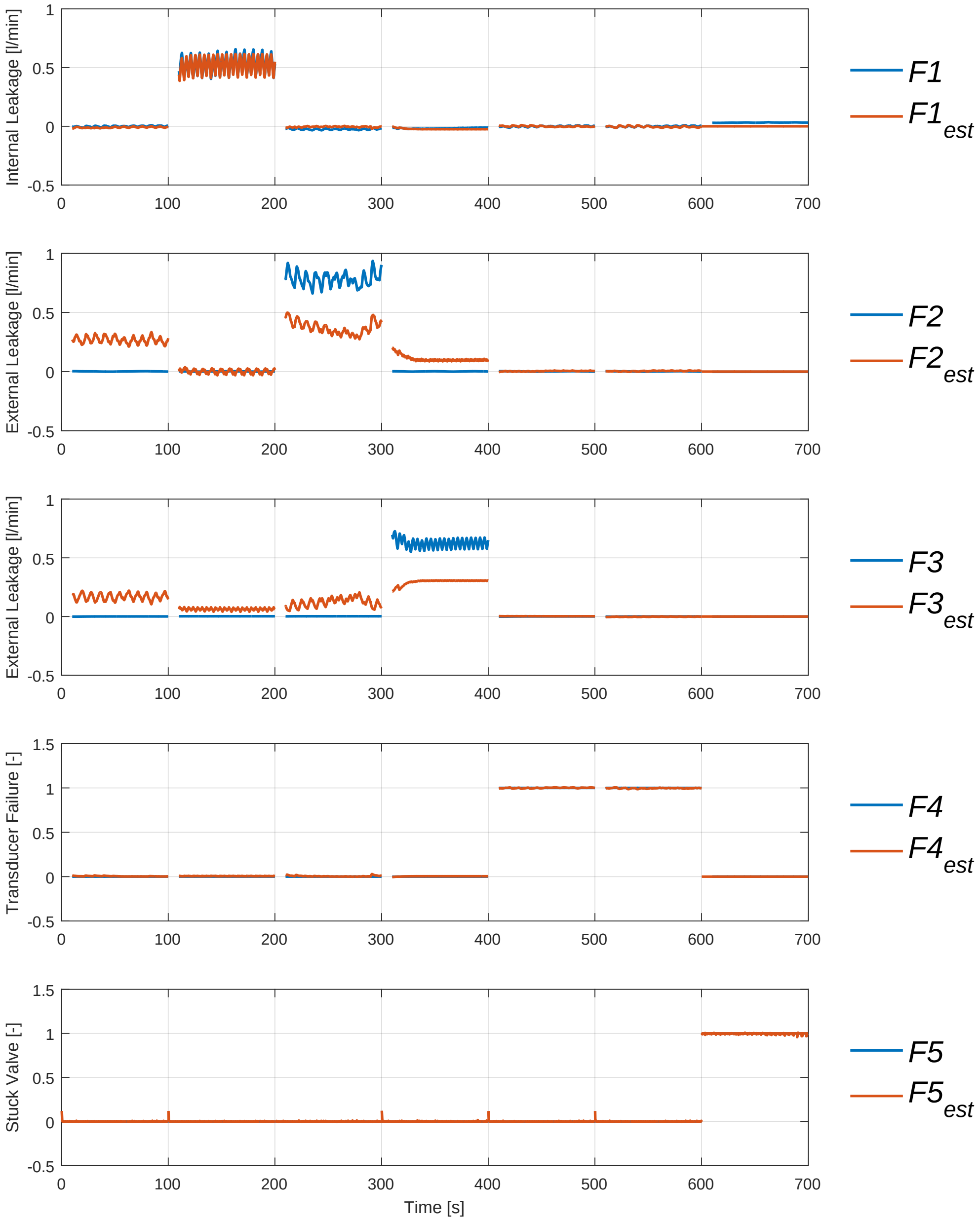


Figure 12.3: Experimental results of the FDD scheme with NARX ANN compared with the measured faults. Every 100 second represents a given fault in ascending order from $F0$-$F5$ with $F4$ being represented twice to emulate 1 & 250 bar pressure transducer output.

The results reveal similar tendencies as the case with a classical estimator shown in fig. 12.2. To visualise the difference, the respective RMS values of both FDD schemes are shown in fig. 12.4. These are compared with the theoretical performance of the model-based FDD scheme previously described in sec. 8.3 where a sinusoidal reference was applied.

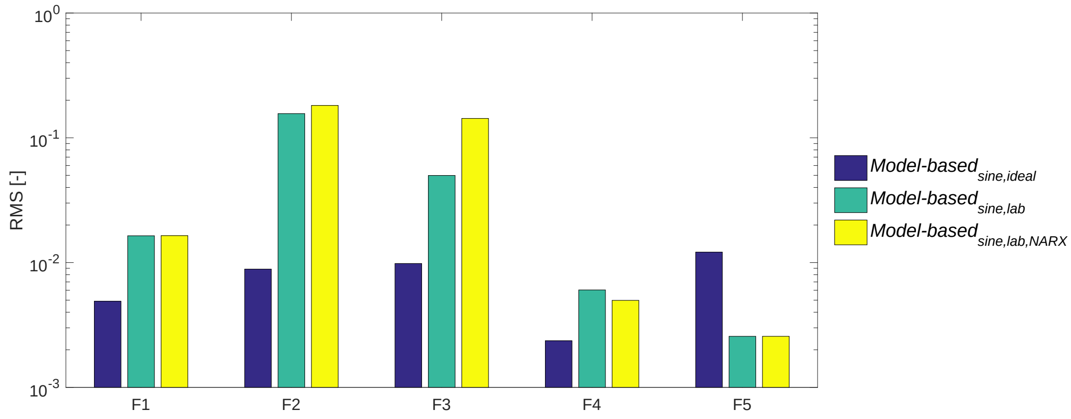


Figure 12.4: RMS values of estimated- and actual faults from the experimental tests compared with the theoretical RMS values from results shown previously in fig. 8.7.

From fig. 12.4 it can be seen that the FDD scheme with a classical estimator shows slightly better performance than the NARX ANN estimator. Furthermore, the theoretical performance is superior during the leakage faults. This is considered to be a consequence of the uncertainty connected with measurements and the models ability to predict the system behaviour.

12.4.1 Evaluation of ANN Generalisation

The generalisation of the ANN is evaluated by the FDD scheme's ability to recognise similar faults as previously seen, however, introduced at an arbitrary time instant during one time sequence.

The leakage magnitudes are slightly different than during training but in a somewhat similar range. This will reveal if the FDD scheme is sufficient to be applied on the test set-up.

The results from the test are shown in fig. 12.5.

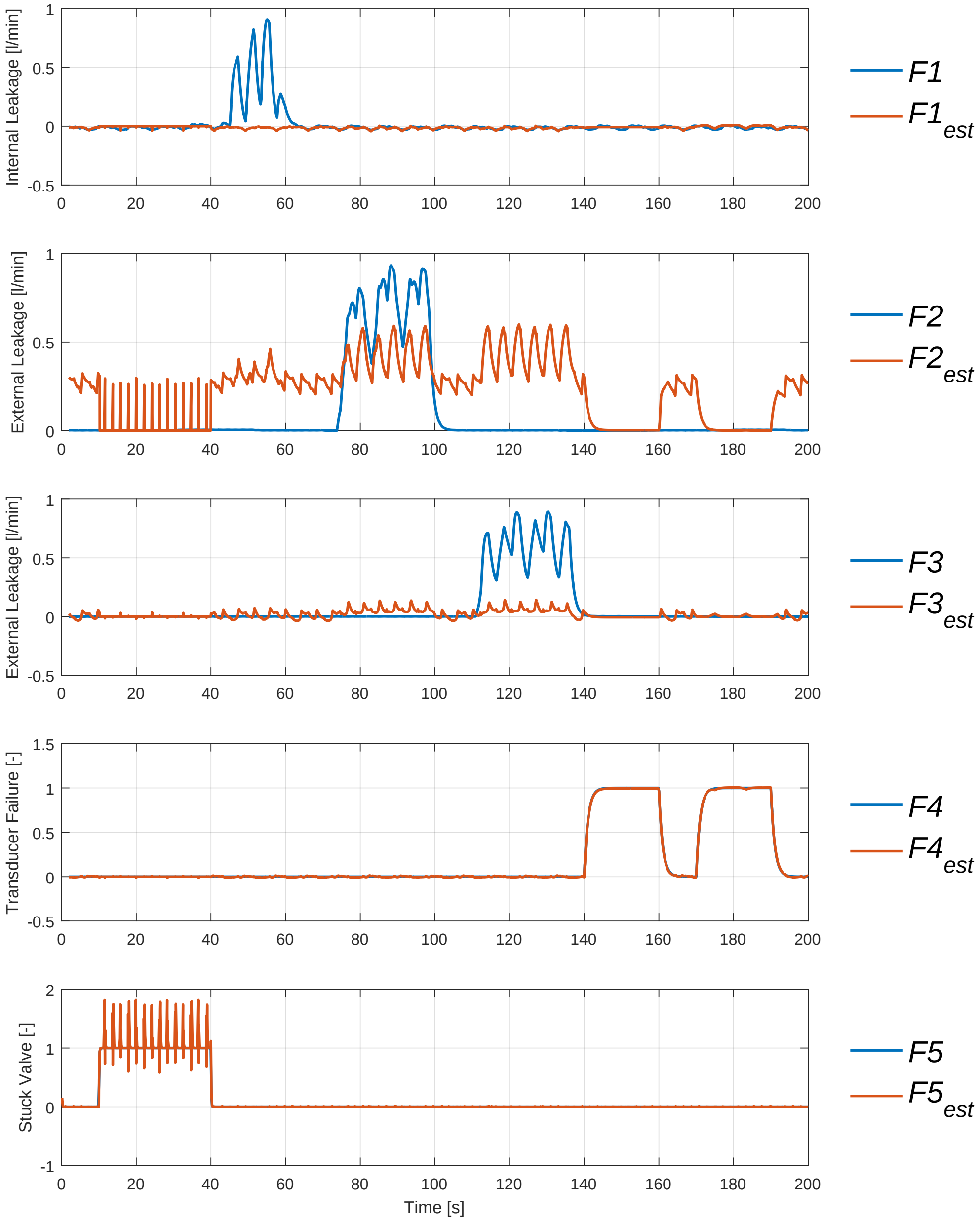


Figure 12.5: Experimental results of the FDD scheme with classical state estimator compared with the measured faults. The leakage magnitudes and time of occurrence are different than the one applied in the training set.

It is observed from fig. 12.5 that the leakage faults are not diagnosed accurately. Therefore, the residuals during these faults are investigated which revealed that only the pressure residuals had a sight noticeable change. These pressure residuals are shown in fig. 12.6.

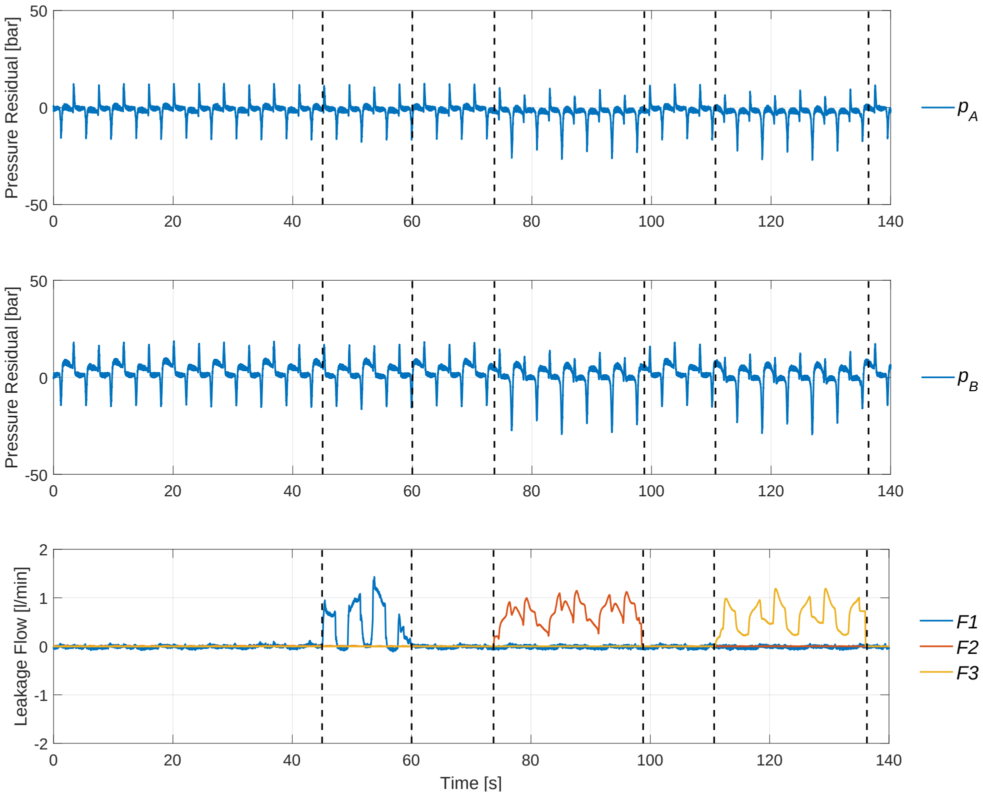


Figure 12.6: Pressure residuals during $F1 - F3$ where the introduction and removal of each fault are marked with a black vertical dotted line.

The graphs illustrates that $F1$ is not noticed in the pressures, but $F2$ & $F3$ causes the pressure residuals to have larger peaks than normal while maintaining the overall tendency.
The main conclusion is that the change in residuals in general is insignificant during $F1$-$F3$. The insufficient state estimator performance and measurement uncertainties compromises the capabilities of the FDD scheme when applied on the physical test set-up. Therefore any future improvements of the FDD scheme is proposed addressed towards these issues, otherwise the ANN FDD approach may not remedy from the performance issues.

12.5 Part Conclusion

The FDD scheme was able to learn seven different patterns representing a series of symptoms connected to different faults. Since the diagnosis is performed by an ANN, which is a black box approach, it is not possible to obtain insight of exactly which phenomena caused the faults to trigger within the ANN. Therefore a test with new data, which previously was not included in the training set was conducted. This resulted in severe decrease of performance during the leakage faults, in the model-based FDD scheme. Hence the generalisation of the ANN could not be verified. This conclusion supports the argument that the training has resulted in convergence between inputs and outputs, but with insufficient capability to recognise the faults when introduced randomly.
Furthermore, when additional data sets from leakage faults were introduced in the training process, the FDD schemes showed decreased performance. This tendency supports the earlier mentioned conclusion, that the measured data might be corrupted.

CHAPTER 13

CONCLUSION

The main objective of this thesis was to investigate how an automated fault detection and diagnosis (FDD) scheme could be designed for the pitching system of a wind turbine. This problem was analysed and relevant literature was reviewed. This lead to the conclusion, that there could be an unrealised potential by applying artificial neural networks (ANN) to design such a FDD scheme. The final problem statement was:

> *"Utilising standard artificial neural networks, how may faults be detected and diagnosed within a hydraulic servo system?*
> *The faults considered are: internal/external leakages, stuck servo valve and pressure transducer failure."*

A solution strategy was formulated to answer the question in a systematic manner and the main part conclusions hereof are summarised in this chapter.

A mathematical dynamic model of the hydraulic servo system was developed and this was initially used to analyse the leakage faults. Based on the analysis of the leakages it was concluded that abnormalities may occur in the measurable variables: chambers pressures, piston position, piston velocity and valve spool displacement. The magnitudes of the residuals were in a promising range compared with the defined level of measurement uncertainty. However, the lowest severities of faults were considered to be in the periphery of what were expected measurable and it was therefore not considered feasible to attempt diagnosing faults below this limit.

Different FDD schemes based on ANNs were proposed for preliminary testing with inspiration from the literature, where especially the novel aspect from [W. J. Crowther, 1998] using a physical understanding of the system to determine the inputs, was adapted in this thesis. The specific combination of the symptoms and faults have not been tested in any of the reviewed literature. Furthermore, application of NARX ANN to generate residuals used for diagnosis of a hydraulic servo system has not previously been tested.
The preliminary tests showed that a model-based scheme possessed greatest potential for the leakage faults in the hydraulic servo system, and this was therefore chosen for further investigation. Furthermore, it was concluded that the most efficient diagnosis was obtained when a given number of previous system states were included along with the current state of

the system. This approach is also consensus in the reviewed literature regarding fluid power FDD with application of ANNs.

The model-based scheme was expanded to include the transducer failure and stuck servo valve and an optimisation was proposed to minimise the risk of either under- or overfitting of the ANNs. These optimum designs were developed to approximate the ANN complexity required for the specific problem. The optimised designs did yield superior performance compared with the preliminary designs.
The FDD scheme was trained to recognise three different operating patterns. The scheme was trained to recognise the faults of interest by analysing the system states.

The evaluation was done in a strategic manner, so that only one parameter was changed for each test, making it possible to limit possible sources of error. It was concluded that the FDD scheme was sufficient with respect to generalisation when a mathematical model was applied to generate residuals. However, the scheme did show a significant decrease of performance during stationary operation of the piston for all the considered faults.
Measurement noise was simulated and the results did not show any significant deterioration. Furthermore, velocity measurements were omitted from the tests which yielded a delayed but accurate diagnosis.
Analysis of the NARX ANN state estimator yielded insufficient performance when subjected to the three different operating patterns. Therefore, the NARX ANN estimator with an external load applied was only tested with a sinusoidal position reference. This tests revealed that external leakage flows were detectable, however, the origin of the leakage could not be determined. The remaining faults were detected with sufficient certainty.
Due to the issues connected with the NARX ANN estimator, it was proposed to evaluate the performance of a pure data driven scheme in a similar manner as for the model-based scheme. This did not yield the desired performance on diagnosis of the leakage flows, however, sufficient for the remaining faults. It was proposed to add an extra hidden layer in the ANN. The additional layer did enhance the diagnosis performance at the cost of increased training period. However, the theoretical performance was not superior compared with the model-based scheme with a classical state estimator.

Software for supervisory control and data acquisition has been developed and implemented to the hardware on the test set-up. This serves the main purpose of doing data acquisition from the transducers and control of the servo valves. Furthermore, safety and condition monitoring considerations have been presented, and implemented, which will ensure that the system will run as intended and that the necessary data will be logged.
A method for communicating between Simulink and LabVIEW was proposed, such that the fault detection and diagnosis scheme could be executed in the Simulink environment, while receiving real system measurements from the LabVIEW environment. The communication method have not been tested on the set-up.

Based on measurements from the experimental results the model could not be verified in the entire test sequence. Therefore, the derived state estimator was concluded to be a rough estimate of the measured states.
It was experimentally tested if the diagnosis ANN could distinguish between modelling errors and residuals caused by actual faults. The faults of interest were emulated on the test set-up and the data was used to train the FDD scheme, which ultimately showed a decrease in

performance compared to the theoretical studies. In the experimental results, the FDD scheme was able to learn seven different patterns representing the different faults.
The diagnosis is performed by an ANN, which is a black box approach and it was therefore not possible to obtain insight in which phenomena are causing a given fault to trigger. However, an analysis of the residuals given to the ANN was done, which showed a correlation between the significance of residual and the ability to diagnose a given fault.

Lastly, the experimental tests did not yield equal performance compared with the theoretical tests. This indicates that the FDD scheme may remedy if the measurements and operation conditions on the practical set-up can be verified with the model.
In short the most promising FDD approach utilising standard ANNs in regard to the presented faults for the hydraulic servo system, was a model-based scheme with a classical state estimator utilising FTDANNs.

Future Aspects

Further work is proposed towards a more accurate state estimator with the use of either NARX ANN or a classical state estimator. This is expected to enhance the model-based FDD scheme. Furthermore, the certainty of the measurements from the test set-up is desired improved in order for a mathematical model to be accurately verified.

Alternatively the model may be excluded, by the addition of hidden layers in the data driven diagnosis. Due to the simulated improvements when adding a hidden layer. Therefore, further attention is proposed towards investigating application of several hidden layers. Furthermore, it was found through simulation and a literature review of [X. Glorot, 2010], that a gradient based training algorithm is not ideal for deep layer ANNs. It is lastly suggested to investigate other training algorithms.

BIBLIOGRAPHY

A. El-Betar, M. M. Abdelhamed, 2006. A. El-Assal R. Adbelsatar A. El-Betar, M. M. Abdelhamed. *Fault Diagnosis of a Hydraulic Actuator Circuit Using An Artificial Neural Network*. pages 1–21, 2006.

A. S. Willsky, 1974. H. L. Jones A. S. Willsky. *A generalized likelihood ratio approach to state estimation in linear systems subject to abrupt changes*. 1974.

Arora, 2012. Jasbir S. Arora. *Optimum Design*. Academic Press, 3^{rd} edition, 2012. ISBN 978-0-12-381375-6.

B. Chen, P. C. Matthews, 2013. P. J. Tavner B. Chen, P. C. Matthews. *Wind turbine pitch faults prognosis using a-priori knowledge-based ANFIS*. pages 1–14, 2013.

Beard, 1973. R. V. Beard. *Failure Accomodation in Linear Systems through Self-Reorganization*. 1973.

C. S. Byington, M. Watson, 2004. D. Edwards C. S. Byington, M. Watson. *Data-driven neural network methodology to remaining life predictions for aircraft actuator components*. pages 1–9, 2004.

C.L.Phillips og J.M.Parr, 2011. C.L.Phillips og J.M.Parr. *Feedback control systems*. Pentice Hall, 5^{th} edition, 2011.

Cybenko, 1989. G. Cybenko. *Approximation by Superpositions of a Sigmoidal Function*. pages 1–6, 1989.

Diaconescu, 2008. E. Diaconescu. *The use of NARX Neural Networks to predict Chaotic Time Series*. pages 1–10, 2008.

Dimopoulos, 2011. Hercules G. Dimopoulos. *Analog Electronic Filters: Theory, Design and Synthesis*. Springer Science & Business Media, 2011. ISBN 9789400721906.

E. Sales-Setien, I. Penarrocha, 2015. D. Dolz R. Sanchis E. Sales-Setien, I. Penarrocha. *Fault Detection in the Blade and Pitch System of a Wind Turbine with H2 PI Observers*. pages 1–10, 2015.

Ehsan Sobhani-Tehrani, 2009. Khashayar Khorasani Ehsan Sobhani-Tehrani. *Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach*. Springer, 2009. ISBN 978-0-387-92906-4.

G. J. Preston, D. N. Shields, 1992. S. Daley G. J. Preston, D. N. Shields. *Application of a Robust Nonlinear Fault Detction Observer to a Hydraulic System*. pages 1–6, 1992.

Gertler, 1998. Janos Gertler. *Fault Detection and Diagnosis in Engineering Systems (Electrical Engineering and Electronics)*. CRC Press, 1998. ISBN 0-8247-9427-3.

H. Khan, C. Abou, 2005. N. Sepehri H. Khan, C. Abou. *Nonlinear observer-based fault detection technique for electro-hydraulic servo-positioning systems*. pages 1–23, 2005.

H. Khan, S. Abou, 2002. N. Sepehri H. Khan, S. Abou. *Fault Detection in Electro-Hydraulic Positioning Systems Using Sequential Test of Wald*. pages 1–6, 2002.

H. Liu, D. Liu, 2014. C. Lu X. Wang H. Liu, D. Liu. *Fault Diagnosis of Hydraulic Servo System Using The Unscented Kalman Filter*. pages 1–13, 2014.

H. Rahn, 2015. M. Groenkjaer H. Rahn. *Control and Experimental Evaluation of Speed-variable Switched Differential Pump Concept*, Aalborg University, 2015.

H. Su, T. J. McAvoy, 1992. P. Werbos H. Su, T. J. McAvoy. *Long-Term Predictions of Chemical Processes Using Recurrent Neural Netwroks: A Parallel Training Approach*. pages 1–15, 1992.

H. Wang, 1996. S. Daley H. Wang. *Actuator Fault Diagnosis: An Adaptive Observer-Based Technique*. pages 1–6, 1996.

Hecht-Nielsen, 1988. R. Hecht-Nielsen. *Neurocomputing: Picking the Human Brain*. pages 1–6, 1988.

Hornik, 1991. K. Hornik. *Approximation Capabilities of Muitilayer Feedforward Networks*. pages 1–6, 1991.

Hosovsky, 2011. A. Hosovsky. *Genetic Optimization of Neural Network Structure for Modeling of Biomass Fired Boiler Emissions*. pages 1–6, 2011.

Industries, 2016. Hydratech Industries. *Research and Development Department*. `http://www.hydratech-industries.com/En-US/RnD/SitePages/Home.aspx`, 2016.

Instruments, 2006. National Instruments. *UDP Communication in LabVIEW*. `http://www.ni.com/tutorial/4950/en/`, 2006.

Isermann, 2006. Rolf Isermann. *Fault-Diagnosis Systems An Introduction from Fault Detection to Fault Tolerance*. Springer, 2006. ISBN 978-3-540-24112-6.

J. Anzurez-Marin, 2009. O. Cuevas-Silva J. Anzurez-Marin. *The Fault Diagnosis Problem: Residual Generators Design Using Neural Networks in a Two-tank Interconnected System*. pages 1–6, 2009.

J. Du, 2010. S. Wang J. Du. *Hiberarchy Clustering Fault Diagnosis of Hydraulic Pump*. pages 1–7, 2010.

J. Liniger, M. Soltani, 2016. H. C. Pedersen J. Carroll N. Sepehri J. Liniger, M. Soltani. *Reliability based design of fluid power pitch systems for wind turbines*. pages 1–12, 2016.

K. Cao, J. Zhang, 2007. L. Hu K. Cao, J. Zhang. *Fault Diagnosis of Electro-hydraulic Position Servo Closed-loop System Based on Support Vector Regression*. pages 1–6, 2007.

L. An, 2003. N. Sepehri L. An. *Hydraulic Actuator Circuit Fault Detection Using Extended Kalman Filter*. pages 1–6, 2003.

L. An, 2014a. N. Sepehri L. An. *Hydraulic Actuator Leakage Fault Detection Using Extended Kalman Filter.* pages 1–14, 2014.

L. An, 2006. N. Sepehri L. An. *Hydraulic Actuator Lekage Quantification Scheme Using Extended Kalman Filter and Sequential Method.* pages 1–6, 2006.

L. An, 2014b. N. Sepehri L. An. *Leakage Fault Detection in Hydraulic Actuators Subject to Unknown External Loading.* pages 1–12, 2014.

L. An, 2004. N. Sepehri L. An. *Leakage Fault Identification in a Hydraulic Positioning System Using Extended Kalman Filter.* pages 1–6, 2004.

L. Guo, 2004. H. Wang L. Guo. *Fault detection and diagnosis for general stochastic systems using B-spline expansions and nonlinear observers.* pages 1–6, 2004.

L. Hongmei, W. Shaoping, 2006a. O. Pingchao L. Hongmei, W. Shaoping. *Fault Diagnosis Based on Improved Elman Neural Network for a Hydraulic Servo System.* pages 1–6, 2006.

L. Hongmei, W. Shaoping, 2006b. O. Pingchao L. Hongmei, W. Shaoping. *Fault Diagnosis in a Hydraulic Position Servo System Using RBF Neural Network.* pages 1–7, 2006.

L. Li, 2013. T. Thurner L. Li. *Accurate Modeling and Identification of Servo-Hydraulic Cylinder Systems in Multi-Axial Test Applications.* pages 1–9, 2013.

L. Ning, G. Huaping, 2011. J. Wei T. Cheng L. Ning, G. Huaping. *Discussion the application of parameter measurement on equipment hydraulic system fault diagnosis.* pages 1–4, 2011.

M. H. Beale, M. T. Hagan, 2016. H. B. Demuth M. H. Beale, M. T. Hagan. *Neural Network Toolbox User Guide.* `http://se.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf`, 2016.

M. Karpenko, N. Sepehri, 2002. D. Scuse M. Karpenko, N. Sepehri. *Diagnosis of Process Valve Actuator Faults Using a Multilayer Neural Network.* pages 1–11, 2002.

M. T. Hagan, 1994. M. B. Menhaj M. T. Hagan. *Training feedforward networks with the Marquardt algorithm.* pages 1–5, 1994.

MathWorks, 2015a. MathWorks. *Elman neural network.* `http://se.mathworks.com/help/nnet/ref/elmannet.html`, 2015.

MathWorks, 2016a. MathWorks. *Feedforward neural network.* `http://se.mathworks.com/help/nnet/ref/feedforwardnet.html`, 2016.

MathWorks, 2016b. MathWorks. *Design Time Series NARX Feedback Neural Networks.* `http://se.mathworks.com/help/nnet/ug/design-time-series-narx-feedback-neural-networks.html`, 2016.

MathWorks, 2015b. MathWorks. *Nonlinear autoregressive neural network with external input.* `http://se.mathworks.com/help/nnet/ref/narxnet.html`, 2015.

MathWorks, 2015c. MathWorks. *Improve Neural Network Generalization and Avoid Overfitting.* `http://se.mathworks.com/help/nnet/ug/improve-neural-network-generalization-and-avoid-overfitting.html`, 2015.

MathWorks, 2015d. MathWorks. *Neural Network Toolbox Release Notes.* `http://se.mathworks.com/help/nnet/release-notes.html`, 2015.

Mohieddine Jelali, 2003. Andreas Kroll Mohieddine Jelali. *Hydraulic Servo-systems.* Springer, 2003. ISBN 978-1-4471-1123-8.

MOOG, 2009. MOOG. *Servovalves For Electrohydraulic position, velocity, pressure or force control systems D633,* MOOG, 2009.

P. Li, 2001. V. Kadirkamanathan P. Li. *Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems.* pages 1–7, 2001.

P. S. Crowther, 2005. R. J. Cox P. S. Crowther. *A Method for Optimal Division of Data Sets for Use in Neural Networks.* Springer Berlin Heidelberg, 2005. ISBN 978-3-540-31997-9.

Pedersen, 2016. *Henrik Clemmensen Pedersen.* Private communication, 2016.

R. H. Chen, 1999. J. L. Speyer R. H. Chen. *Optimal Stochastic Fault Detection Filter.* pages 1–6, 1999.

R. Isermann, 2010. M. Munchhof R. Isermann. *Identification of Dynamic Systems: An Introduction with Applications (Advanced Textbooks in Control and Signal Processing).* Springer, 2010. ISBN 978-3-540-78878-2.

R. J. Patton, 1996. J. Chen R. J. Patton. *Optimal Filering and Robust fault diagnosis of stochastic systems with unknown disturbances.* pages 1–6, 1996.

R. J. Patton, J. Chen, 1994. T. M. Siew R. J. Patton, J. Chen. *FAULT DIAGNOSIS IN NONLINEAR DYNAMIC SYSTEMS VIA NEURAL NETWORKS.* pages 1–6, 1994.

R. Song, 2002. N. Sepehri R. Song. *Fault Detection and Isolation in Fluid Power Systems Using a Parametric Estimation Method.* pages 1–6, 2002.

Rosenblatt, 1958. F. Rosenblatt. *The perceptron: A probabilistic model for information storage & organisation in the brain.* pages 386–408, 1958.

S. Andersson, A. Soderberg, 2006. S. Bjorklund S. Andersson, A. Soderberg. *Friction models for sliding dry, boundary and mixed lubricated contacts.* pages 1–8, 2006.

S. He, 2000. N. Sepehri S. He. *Online Modeling and Prediction of a Hydraulic Force-acting System Using Neural Networks.* pages 1–6, 2000.

S. Rajasekaran, 2012. G.A. Vijayalakshmi Pai S. Rajasekaran. *Neural Networks Fuzzy Logic, and Genetic Algorithms.* Phi Learning Private Limited, 2012. ISBN 978-81-203-2186-1.

T. Kobayashi, 2008. D. L. Simon T. Kobayashi. *Aircraft Engine On-Line Diagnostics Through Dual-Channel Sensor Measurements: Development of a Baseline System.* pages 1–17, 2008.

T. Kobayashi, 2003. D. L. Simon T. Kobayashi. *Application of a Bank of Kalman Filters for Aircraft Engine Fault Diagnostics.* pages 1–15, 2003.

T. Kobayashi, 2006. D. L. Simon T. Kobayashi. *Hybrid Kalman Filter A New Approach for Aircraft Engine In-Flight Diagnostics.* pages 1–26, 2006.

T. T. Le, J. Watton, 1997. D. T. Pham T. T. Le, J. Watton. *An artificial neural network based approach to fault diagnosis and classification of fluid power systems.* pages 1–11, 1997.

T.O.Andersen, 2003. T.O.Andersen. *Fluid Power Systems - Modelling and Analysis*, Aalborg University Institute of Energy Technology, 2003.

T.O.Andersen, 2007. M.R.Hansen T.O.Andersen. *Fluid Power Circuits - System Design and Analysis*, Aalborg University, 2007.

V. A. Skormin, 1995. J. Apone V. A. Skormin. *ON-LINE DIAGNOSTICS OF A VARIABLE DISPLACEMENT PUMP OF A FLIGHT ACTUATION SYSTEM.* pages 1–8, 1995.

V. A. Skormin, J. Apone, 1992. J. J. Dunphy V. A. Skormin, J. Apone. *On-Line Diagnostics of a Self-Contatined Flight Actuator.* pages 1–11, 1992.

W. J. Crowther, K. A. Edge, 1998. C. R. Burrows R. M. Atkinson D. J. Woolons W. J. Crowther, K. A. Edge. *Fault Diagnosis of a Hydraulic Actuator Circuit Using Neural Networks - an Output Vector Space Classification Approach.* pages 1–12, 1998.

W. Mcculloch, 1943. W. Pitts W. Mcculloch. *A logical calculus of the ideas immanent in nervous activity.* pages 115–133, 1943.

W. Vianna, 2014. J. Malere W. Vianna. *Aircraft Hydraulic System Leakage Detection and Servicing Recommendations Method.* pages 1–5, 2014.

Watton, 2007. John Watton. *Modelling, Monitoring and Diagnostic Techniques for Fluid Power Systems.* Springer, 2007. ISBN 978-1-84628-373-4.

Wilkinson, 2011. Michael Wilkinson. *Empirical analys is of wind turbine reliability.* `http://www.ewea.org/annual2011/fileadmin/ewec2011_files/images/conference/Side_Events/Empirical_analysis_of_wind_turbine_reliability.pdf`, 2011.

X. Glorot, 2010. Y. Bengi X. Glorot. *Understanding the difficulty of training deep feedforward neural networks.* pages 1–8, 2010.

X. Wu, Y. Li, 2012. F. Li Z. Yang W. Teng X. Wu, Y. Li. *Adaptive Estimation-Based Leakage Detection for a Wind Turbine Hydraulic Pitching System.* pages 1–8, 2012.

Y. Vidal, C. Tutiven, 2015. J. Rodellar L. Acho Y. Vidal, C. Tutiven. *Fault Diagnosis and Fault-Tolerant Control of Wind Turbines via a Disctete Time Controller with a Disturbance Compensator.* pages 1–17, 2015.

Yi-Hui, 2007. L. Yi-Hui. *Evolutionary Neural Network Modeling for Forecasting the Field Failure Data of Repairable Systems.* pages 1–7, 2007.

Z. Yinshuo, X. Jun, 2013. L. Lei Z. Yinshuo, X. Jun. *Fault Diagnosis of Hydraulic System Based on Improved BP Neural Network Technology.* pages 1–4, 2013.

Part V

Appendices

Appendix

A

Linear Model

In the following a linear model describing a double-acting hydraulic actuator will be derived. All linearised variables are denoted with a 0 in the subscript, e.g. p_{A0} is a constant linearisation pressure for chamber A. The derived nonlinear model is the baseline, although some rewritings of the equations are done to simplify the derivation of the linear model. The equations are shown in (A.1)-(A.7)

$$\dot{p}_A = Q_A - A_A \dot{x}_P \underbrace{\frac{\beta(p_{A0})}{A_A x_p + V_{A0}}}_{c_{hA}} \tag{A.1}$$

$$\dot{p}_B = Q_B + \alpha A_A \dot{x}_P \underbrace{\frac{\beta(p_{B0})}{V_{B0} - \alpha A_A x_p}}_{c_{hB}} \tag{A.2}$$

$$Q_{S\to A} = k_V x_V \sqrt{p_S - p_A} \quad , \quad x_V \geq 0 \tag{A.3}$$

$$Q_{A\to T} = k_V x_V \sqrt{p_A - p_T} \quad , \quad x_V \leq 0 \tag{A.4}$$

$$Q_{S\to B} = k_V x_V \sqrt{p_S - p_B} \quad , \quad x_V \leq 0 \tag{A.5}$$

$$Q_{B\to T} = k_V x_V \sqrt{p_B - p_T} \quad , \quad x_V \geq 0 \tag{A.6}$$

$$k_V = k_A C_D \sqrt{\frac{2}{\rho_{oil}}}$$

$$F_L = p_A A_A - \alpha p_B A_A = p_L A_A \tag{A.7}$$

$$\alpha = \frac{A_B}{A_A}$$

The equations are linearised by applying the presented assumptions:

(i.1) The servo valves are constructed with ideal zero-lap spools
(i.2) The servo valves spool have perfect symmetrical opening areas
(i.3) Possible leakage flows in servo valves are disregarded
(i.4) Constant supply- & tank pressure
(i.5) Bulk modulus is constant
(i.6) The dynamics of the servo valves are disregarded due to their rapid response compared with the rest of the system
(i.7) The actuator has no internal leakage
(i.8) Steady state flow conditions

Reduction of the System Order and Linearisation

For simplification, a reduced order model is derived in the following. This is done by applying **(i.2)**, **(i.3)** plus **(i.8)** and only considering positive spool position the flows can be used to describe the chamber pressures.

$$\alpha Q_A = -Q_B \tag{A.8}$$

$$\Updownarrow \tag{A.9}$$

$$\alpha k_V x_V \sqrt{p_S - p_A} = -k_V x_V \sqrt{p_B - p_T} \tag{A.10}$$

$$\Updownarrow \tag{A.11}$$

$$\alpha \sqrt{p_S - p_A} = -\sqrt{p_B - p_T} \tag{A.12}$$

$$\Updownarrow \tag{A.13}$$

$$\alpha^2 (p_S - p_A) = (p_B - p_T) \tag{A.14}$$

$$\tag{A.15}$$

The chamber pressures can be isolated to:

$$p_A = \frac{\alpha^2 p_S + p_T - p_B}{\alpha^2} \tag{A.16}$$

$$p_B = \alpha^2 p_S + p_T - p_A \alpha^2 \tag{A.17}$$

The load pressure in (A.7) can be described as (A.18) and be used to isolate A and B pressures depending on the load pressure. The reason for this rewriting will become clear when linearising the flow.

$$p_L = p_A - \alpha p_B = \frac{\alpha^2 p_S + p_T - p_B}{\alpha^2} - \alpha^2 p_S + p_T - p_A \alpha^2 \tag{A.18}$$

$$\Updownarrow \tag{A.19}$$

$$p_A = \frac{p_L + \alpha^3 p_S + \alpha p_T}{1 + \alpha^3} \text{ for } x_v \geq 0 \tag{A.20}$$

$$p_B = \frac{\alpha^2 p_S + p_T - \alpha^2 p_L}{1 + \alpha^3} \text{ for } x_v \geq 0 \tag{A.21}$$

Similar approach is taken for the negative spool position resulting in (A.22) and (A.23).

$$p_A = \frac{p_L + \alpha^3 p_S + \alpha p_T}{1 + \alpha^3} \text{ for } x_v \leq 0 \tag{A.22}$$

$$p_B = \frac{p_S + \alpha^2 p_T - \alpha^2 p_L}{1 + \alpha^3} \text{ for } x_v \leq 0 \tag{A.23}$$

By Taylor approximation the flows can be linearised as shown in (A.24) and (A.26) with the shown linearisation coefficients.

$$\Delta Q_A = k_{q,A} \Delta x_V + k_{qp,A} \Delta p_A \tag{A.24}$$

$$k_{q,A} = \left. \frac{\partial Q_A}{\partial x_v} \right|_{\overline{p}_0} = \begin{cases} k_V \sqrt{p_S - p_{A0}} & \text{for } x_v \geq 0 \\ k_V \sqrt{p_{A0} - p_T} & \text{for } x_v \leq 0 \end{cases} ; \; k_{qp,A} = \left. \frac{\partial Q_A}{\partial p_A} \right|_{\overline{p}_0} = \begin{cases} \frac{-k_V x_{V0}}{2\sqrt{p_S - p_{A0}}} & \text{for } x_v \geq 0 \\ \frac{k_V x_{V0}}{2\sqrt{p_{A0} - p_T}} & \text{for } x_v \leq 0 \end{cases} \tag{A.25}$$

$$\Delta Q_B = k_{q,B} \Delta x_V + k_{qp,B} \Delta p_B \tag{A.26}$$

$$k_{q,B} = \left. \frac{\partial Q_B}{\partial x_v} \right|_{\overline{p}_0} = \begin{cases} -k_V \sqrt{p_{B0} - p_T} & \text{for } x_v \geq 0 \\ -k_V \sqrt{p_S - p_{B0}} & \text{for } x_v \leq 0 \end{cases} ; \; k_{qp,B} = \left. \frac{\partial Q_B}{\partial p_A} \right|_{\overline{p}_0} = \begin{cases} \frac{-k_V x_{V0}}{2\sqrt{p_{B0} - p_T}} & \text{for } x_v \geq 0 \\ \frac{k_V x_{V0}}{2\sqrt{p_S - p_{B0}}} & \text{for } x_v \leq 0 \end{cases} \tag{A.27}$$

The governing equations have now been reduced and linearised. This allows for a definition of the transfer function.

Deriving the Transfer Function

The pressure changes (Δp_L) can be obtained directly from (A.22) and (A.23) because of **(i.4)**. Note that this is independent of sign on valve reference.

$$\Delta p_A = \frac{1}{1+\alpha^3}\Delta p_L \tag{A.28}$$

$$\Delta p_B = -\frac{\alpha^2}{1+\alpha^3}\Delta p_L \tag{A.29}$$

Now the derivative of change in load pressure can be obtained by utilising (A.1),(A.2), (A.24), (A.26), (A.28) and (A.29).

$$\Delta \dot{p}_L = \Delta \dot{p}_A - \alpha \Delta \dot{p}_B = c_{hA}(\Delta Q_A - A_A \dot{x}_P) - \alpha c_{hB}(\Delta Q_B + \alpha A_A \dot{x}_P) \tag{A.30}$$

$$= c_{hA}(k_{q,A}\Delta x_V + k_{qp,A}\Delta p_A - A_A \Delta \dot{x}_P) - \alpha c_{hB}(k_{q,B} x_V + k_{qp,B} p_B + \alpha A_A \Delta \dot{x}_P) \tag{A.31}$$

$$= c_{hA}\left(k_{q,A}\Delta x_V + k_{qp,A}\frac{1}{1+\alpha^3}\Delta p_L - A_A\Delta \dot{x}_P\right) - \alpha c_{hB}\left(k_{q,B}\Delta x_V - k_{qp,B}\frac{\alpha^2}{1+\alpha^3}\Delta p_L + \alpha A_A \Delta \dot{x}_P\right) \tag{A.32}$$

$$= \underbrace{(c_{hA}k_{q,A} - \alpha c_{hB}k_{q,B})}_{k_Q}\Delta x_V + \underbrace{\left(c_{hA}k_{qp,A}\frac{1}{1+\alpha^3} - c_{hB}k_{qp,B}\frac{\alpha^2}{1+\alpha^3}\right)}_{k_{QP}}\Delta p_L - \underbrace{(c_{hA}A_A + c_{hB}\alpha A_A)}_{A_P}\Delta \dot{x}_P \tag{A.33}$$

$$= k_Q \Delta x_V + k_{QP}\Delta p_L - A_P \Delta \dot{x}_P \tag{A.34}$$

The acceleration of the piston can be described by (A.35).

$$\ddot{x}_P = \frac{p_L A_A - B\dot{x}_P}{m} \tag{A.35}$$

The governing equations (A.34) and (A.35) can be combined to the final TF (A.36) and (A.37) with valve position as input and either piston position or force as output, the position output block diagram of this is shown in fig. A.1.

$$\frac{x_P(s)}{x_V(s)} = \frac{A_A k_Q}{ms^2 + Bs - k_{QP}ms - k_{QP}B + A_A A_P}\frac{1}{s} \tag{A.36}$$

$$\frac{F_P(s)}{x_V(s)} = \frac{A_A k_Q\,(ms+B)}{ms^2 + Bs - k_{QP}ms - k_{QP}B + A_A A_P} \tag{A.37}$$

The eigenfrequencies of the test and load cylinder is given by (A.38) and this as a function of cylinder positions are shown in fig. A.2 & A.3.

$$\omega_{n,cyl} = \sqrt{\frac{A_A^2}{m}\left(c_{hA} + \alpha^2 c_{hB}\right)} \tag{A.38}$$

This rounds the derivation of a linear model, and this can be applied to design a simple position and force controllers for the HSS.

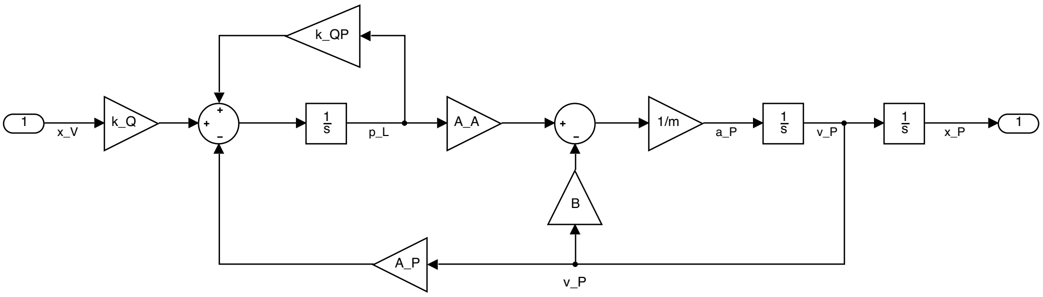


Figure A.1: Block diagram of the linearised HSS by means of (A.34) and (A.35).

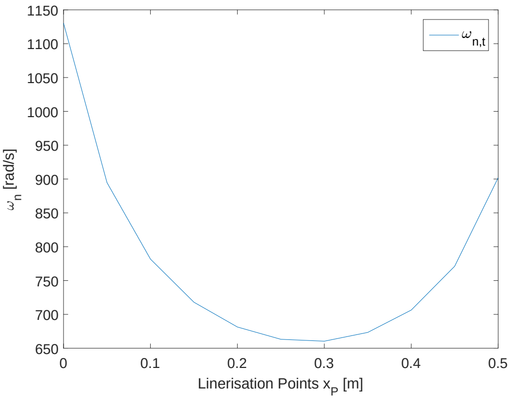


Figure A.2: Eigenfrequency of test cylinder.

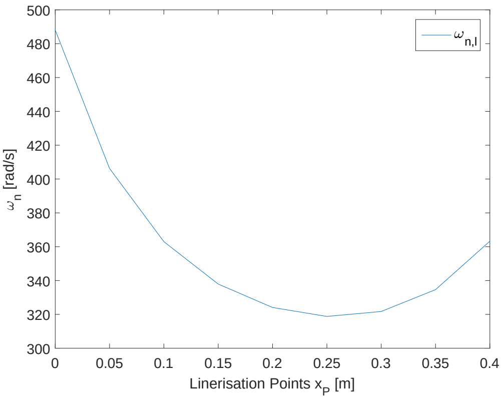


Figure A.3: Eigenfrequency of load cylinder.

A.1 Verification of Linear Model

Verification of the linear model is done by comparing the force step response of the nonlinear model with the force step response of the linear model. The linearisation points for both positive and negative direction can be seen in (A.39).

$$\boldsymbol{x}_{+0,t} = \begin{bmatrix} x_{P0,t} \\ \dot{x}_{P0,t} \\ x_{V0,t} \\ p_{A0,t} \\ p_{B0,t} \end{bmatrix} = \begin{bmatrix} 0.15 & \text{m} \\ 0.03 & \text{m/s} \\ 0.034 & - \\ 36 & \text{bar} \\ 48 & \text{bar} \end{bmatrix} \qquad \boldsymbol{x}_{-0,t} = \begin{bmatrix} 0.15 & \text{m} \\ -0.03 & \text{m/s} \\ -0.1693 & - \\ 64 & \text{bar} \\ 85 & \text{bar} \end{bmatrix} \tag{A.39}$$

The positions are chosen to represent the point with the lowest eigenfrequency. The velocity is chosen to represent the expected average operating condition and the pressures and valve opening are calculated based on the velocity. In this manner zero acceleration and actual required flow rate is applied.

A similar linearisation point is chosen for the load cylinder, however with a position that correspond to (A.40).

$$x_{P0,l} = 0.400 - x_{P0,t} \tag{A.40}$$

This gives the points shown in (A.41).

$$\boldsymbol{x}_{+0,l} = \begin{bmatrix} x_{P0,l} \\ \dot{x}_{P0,l} \\ x_{V0,l} \\ p_{A0,l} \\ p_{B0,l} \end{bmatrix} = \begin{bmatrix} 0.25 & \mathrm{m} \\ 0.03 & \mathrm{m/s} \\ 0.1462 & - \\ 23 & \mathrm{bar} \\ 37 & \mathrm{bar} \end{bmatrix} \qquad \boldsymbol{x}_{-0,l} = \begin{bmatrix} 0.25 & \mathrm{m} \\ -0.03 & \mathrm{m/s} \\ -0.0438 & - \\ 59 & \mathrm{bar} \\ 98 & \mathrm{bar} \end{bmatrix} \tag{A.41}$$

Therefore the force step response is compared by applying the linearised valve opening at $t = 0$. The simulated linear and nonlinear force is shown in fig. A.4.

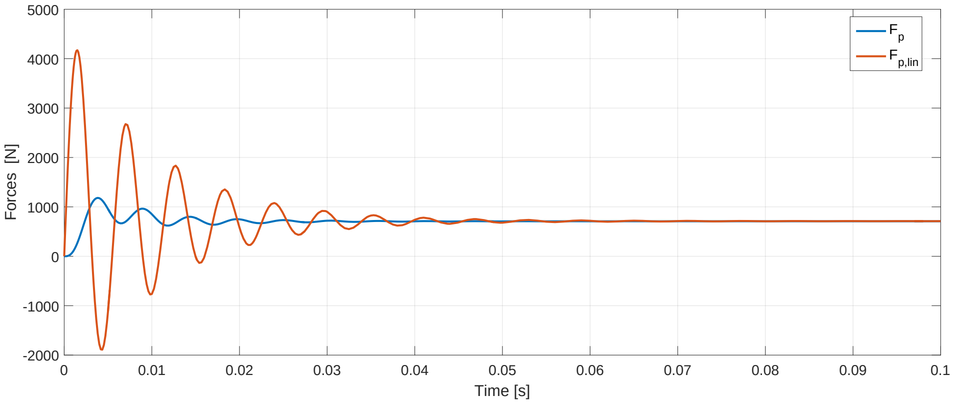


Figure A.4: The step response of the transfer function describing the piston force when a 10% valve signal is given as input. The linear response is without the valve dynamics.

In fig. A.4 it can be observed, that the forces does not match in magnitude or frequency. This is caused by the missing valve dynamics, which influences the system when the step is applied. This indicates, that the valve dynamics cannot be disregarded in the linear model. The response when the 2. order valve dynamics is included can be seen in fig. A.5.

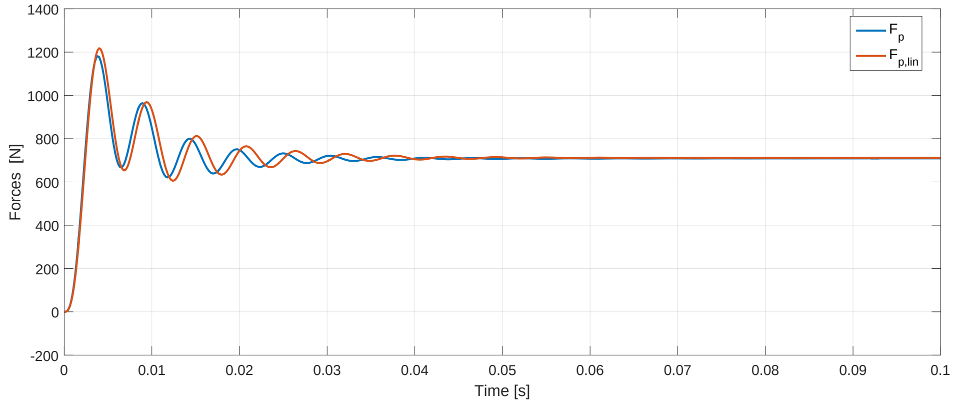


Figure A.5: The step response of the transfer function describing the piston force when a 10% valve signal is given as input. The valve dynamics is included in the response.

In fig. A.5 the valve dynamics has been included in the linear model, and the responses have an acceptable fit. This confirms that the linear force TF can be used to design controllers when the system is near the linearisation point.

The position TF is verified without the free-integrator, i.e by solely considering the velocity responses. The velocity response is shown in fig. A.6.

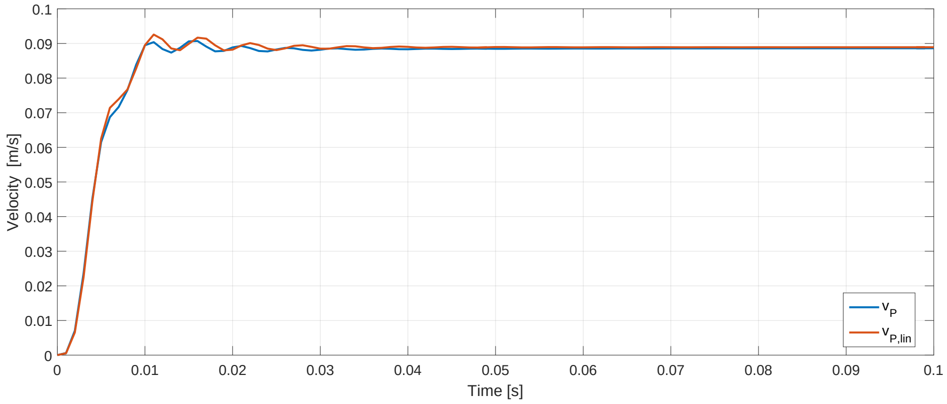


Figure A.6: The step response of the transfer function describing the piston velocity when a 10% valve signal is given as input. The valve dynamics is included in the response.

The velocity in fig. A.6 shows that the linear model agrees with the nonlinear at the linearisation point.

The above analysis shows that the linear model is valid at the verification point. However, the linear model does not agree with the nonlinear when moving away from the linearisation point.

APPENDIX B

FAILURE MECHANISM ANALYSIS

In tab. B the components with corresponding failure modes and causes are presented.

Component	Failure modes	Root causes	Phenomena	Characteristics
Servo valve	Blocked nozzle	Oil pollution (air/water/acid/debris)	Increased zero offset, decreased frequency response, unstable system	Abrupt, obvious fault
	Spool stuck open/closed	Oil pollution, spool valve deformation, solenoid malfunction or spring malfunction	Constant output flow, lower system pressure	Abrupt, obvious fault
	Leakage	Edge wear or radial main spool wear	Increased zero offset, decreased system gain, lower pressure, greater noise	Slow, unobvious fault
	Gain deviation	Elastic element with shock or deformation	Deviated gain	Abrupt, obvious fault features
Sensor	Variable output gain	Adjustment circuit bridge gradual	Time-varying output gain	Slow, gradually obvious fault
	Constant output gain	Adjustment circuit bridge failure	Varied output gain with constant coefficient	Abrupt, obvious fault
	Constant output deviation	Elastic element with shock or deformation	Constantly deviated output	Abrupt, obvious fault
Actuator	Drift	Elastomer with internal force imbalance	Graded-offset output	Slow, unobvious incipient fault
	Leakage	Moving parts wear or sealing failure	Slower velocity, lower system pressure	Slow, unobvious fault
	Stuck	Radial unbalance or oil pollution	Loss of control of the actuator	Abrupt, obvious fault
Pump	Leak of inlet pressure	Supercharging device breakdown, liquid level of oil tank is too low, oil viscosity, inlet pipe	Intermittent violent vibration, abnormal noise, fluctuation of outlet pressure	Slow, gradually obvious fault
	Wear of port plate	Oil pollution, insufficient lubrication	Volumetric efficiency decrease, leakage oil increase, abnormal vibration	Slow, unobvious fault
	Fault of roller bearing	Roller fatigue wear, components of bearing wear and strain	Slight abnormal vibration	Slow, unobvious fault
	Off-center of swashplate	Assembling error, serious wear	slight abnormal vibration	Slow, unobvious gradual fault
	Increscent clearance of piston/shoe	Wear of matching surface between sliding shoe and piston, clearance increment	slight vibration, performance no change	Slow, unobvious increscent fault

Table B.1: Table suggested by [H. Liu, 2014], axial piston pump analysis added from [J. Du, 2010]

APPENDIX C

Activation Functions and Genetic Algorithm

C.1 Activation Functions

The activation function within the artificial neuron, acts like a transfer function, where it translates the input signal to a output signal of the artificial neuron. The most commonly used activation functions are shown in fig C.1-7.1.

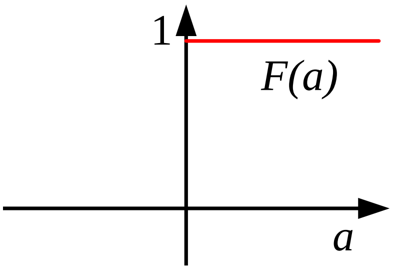


Figure C.1: Unipolar step.

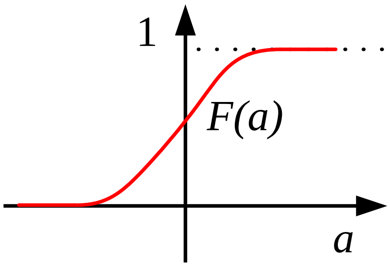


Figure C.2: Unipolar sigmoid.

In fig. C.1 the output is set at one of two possible output levels, depending on if the input signal is above or below a given threshold. This gives the mathematical function in (C.1).

$$F(a) = \begin{cases} 0 & if \quad 0 > a \\ 1 & if \quad a \leq 0 \end{cases} \tag{C.1}$$

In fig. C.2 the output function is defined by the logistical sigmoid function, where the output values are ranging between 0 and 1. The function can be described by (C.2).

$$F(a) = \frac{1}{1+e^{-2a}} \tag{C.2}$$

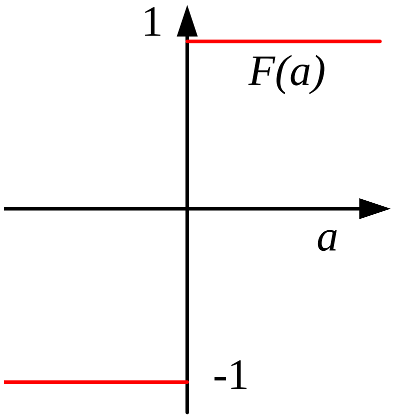


Figure C.3: Bipolar step.

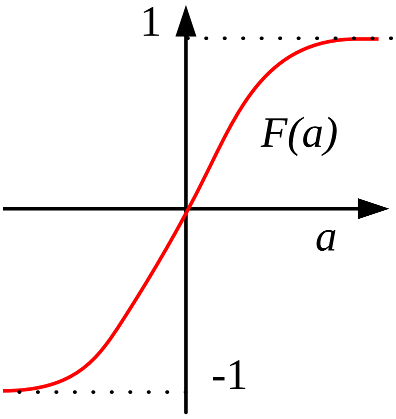


Figure C.4: Bipolar sigmoid.

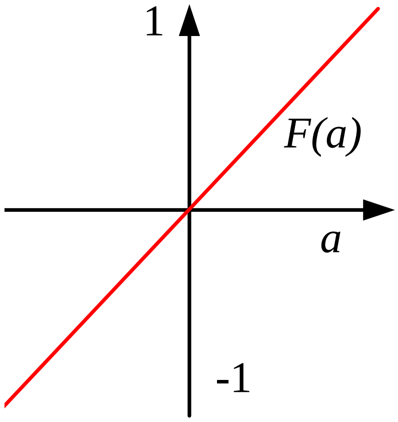


Figure C.5: Identity activation function

Similar to the unipolar function, a bipolar set of the activation functions can be utilised. These do also operate with negative values and the bipolar step function can be described by (C.3).

$$F(a) = \begin{cases} -1 & if \quad 0 > a \\ 1 & if \quad a \leq 0 \end{cases} \tag{C.3}$$

The bipolar sigmoid function is given by (C.4).

$$F(a) = \frac{1 - e^{-2a}}{1 + e^{-2a}} \tag{C.4}$$

Lastly, the identity function is shown in (C.5).

$$F(a) = a \tag{C.5}$$

C.2 Optimisation of ANN Structures

To optimise the number of neurons, input- and output delays, a generic optimisation algorithm (GA) has been utilised.
The input vector denoted as $\boldsymbol{x}$ will consist of the number of hidden neurons, number of input delays and number of output delays, as shown in (C.6).

$$\boldsymbol{x} = \begin{bmatrix} d_u & d_y & n \end{bmatrix}^T \tag{C.6}$$

Note that all inputs can only take integer values, this information will also be provided to the GA.

The GA algorithm can be explained in three main steps

Step 1

GA creates a random initial population.

Step 2

GA creates a sequence of new populations. Every time the algorithm takes a step, GA takes the individuals in the current population and uses these, to generate the next population. In order to generate a new population, GA takes the following sub-steps.

1. Rates all the different members of the current population, and rates them based on their fitness value.
2. All fitness scores are normalised.
3. Members with the highest fitness values, are chosen as parents.
4. Some of the members in the current population which have a lower fitness value, are passed to the next population.
5. GA produces children from the chosen parents. The children can be made by mutating a single parent, or by combining two parents composition.
6. The new generated children becomes the next generation.

Step 3

GA repeats from **Step 2**, or stops if a stopping criteria is met.

The stopping criteria includes a check of current generation number with the maximum allowed, while continuously checking if the number of allowed stall generations has been exceeded. A stall generation is defined as a generation where the average relative change of fitness function does not exceed the function tolerance.
The GA option parameters are shown in tab. C.2.

Max generations	Population size	Stall generations	Function tolerance
50	5	5	1e-3

Table C.1: Setting for the GA to define stopping.

APPENDIX D

Controller Tracking Results

The tracking performance of the controllers designed in chap. 9 are shown in fig. D.1, D.2, D.3 & D.4.

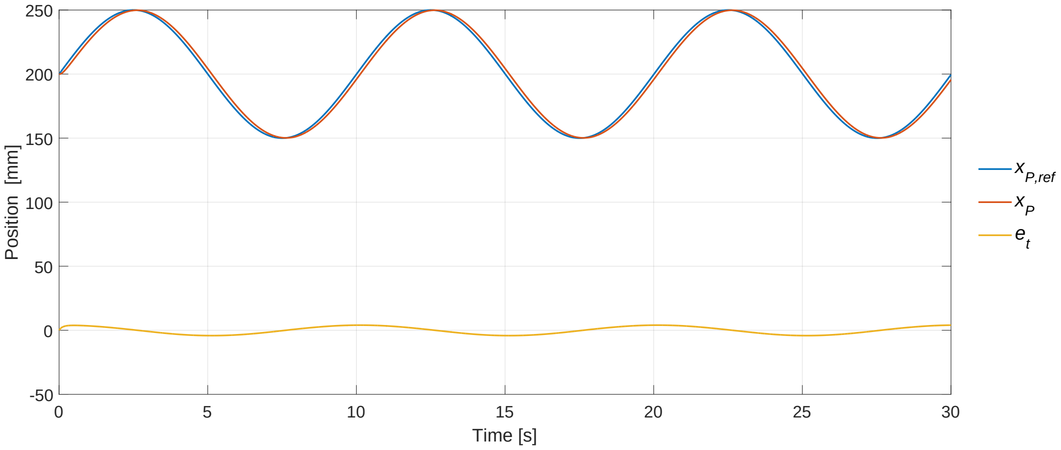


Figure D.1: Simulated position tracking when $G_{C,t}$ is applied.

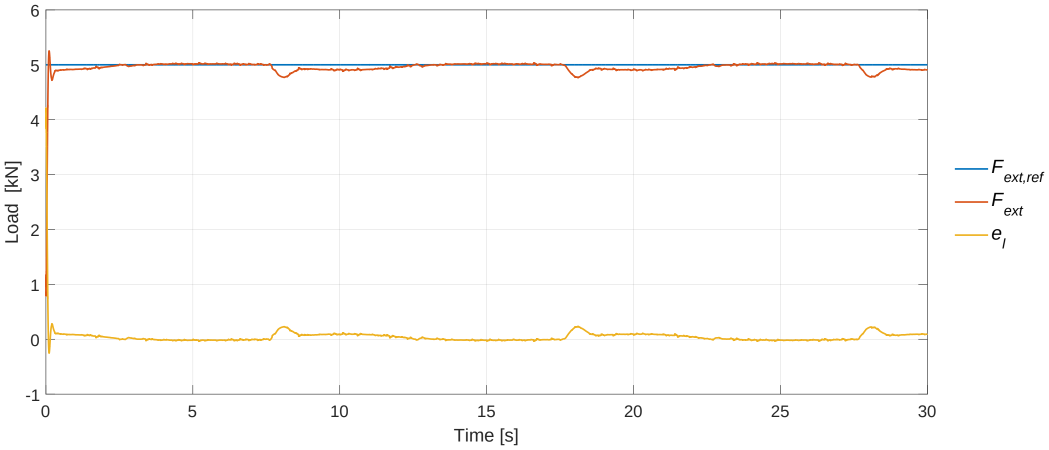


Figure D.2: Simulated force tracking of 5 kN load when both $G_{C,l}$ plus velocity feedforward are applied.

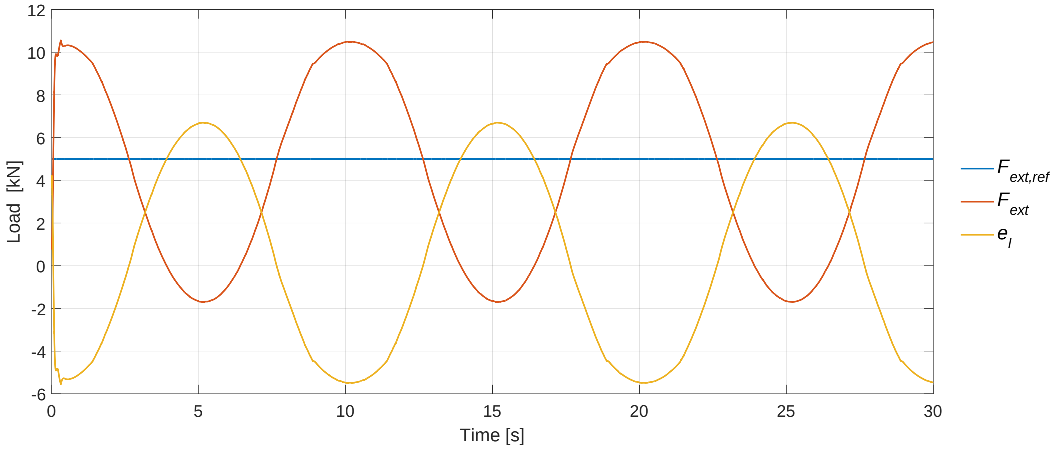


Figure D.3: Simulated force tracking of 5 kN load when $G_{C,l}$ without velocity feedforward is applied.

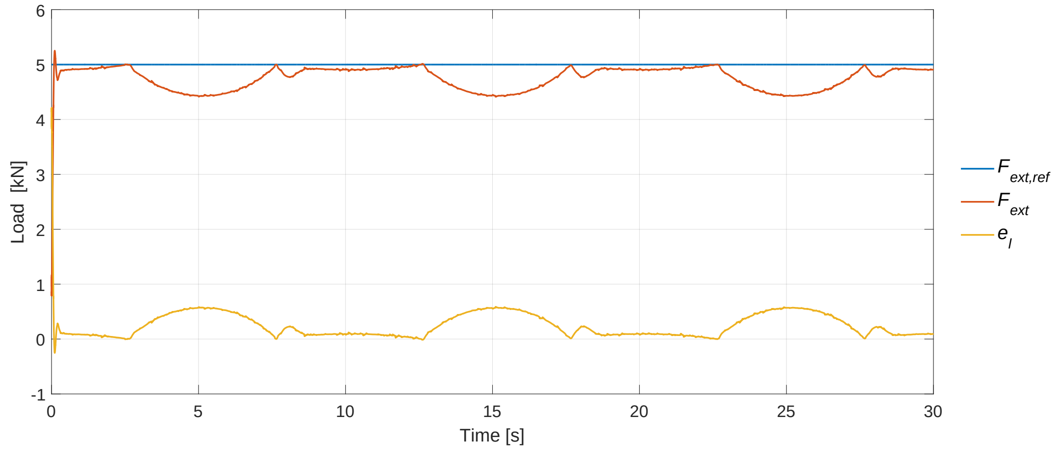


Figure D.4: Simulated force tracking of 5 kN load when both $G_{C,l}$ plus velocity feedforward for positive direction are applied.

Appendix E

Controller Discretisation

The following chapter describes the controller discretisation used for implementation of the controllers in the developed LabVIEW software. Since the LabVIEW software operates in the discrete time domain all controllers needs to be converted from the continuous s-domain to the discrete z-domain.

Tustin's Approximation

In Tustin's approximation, it is assumed that the Laplace transfer function is approximately the same as the transfer function in the z-domain.

This emulation method usually gives good results, if the sampling frequency (1 kHz) is atleast 20 times bigger than the compensated closed loop system bandwidth (the bandwidth for test cylinder is 28 Hz, for the load cylinder it is 63 Hz). From the assumption that the transfer functions in both the s- and z-domain are approximately the same, the maximum possible difference between the z- and s-domain frequency, can be evaluated from (E.1), [C.L.Phillips og J.M.Parr, 2011].

$$\begin{aligned} \omega_w &= \frac{20\omega_b}{\pi}\tan\left(\frac{\pi}{20}\right) \\ \omega_w &= 1.008\omega_b \end{aligned} \tag{E.1}$$

From (E.1) it can be seen, that the difference between the z-domain and s-domain frequency is less than 0.8%, hence Tustin's approximation is considered feasible, this results in (E.2).

$$s \approx \frac{2}{T}\frac{z-1}{z+1} \tag{E.2}$$

Where T is the sampling time of the Real Time Loop.

Proportional and Lag Controllers

For simplicity reasons, the controller discretisation is split up into two parts, where each part is converted separately. The two controllers utilised are as following, a position controller:

$$G_{\mathrm{P}} = k_p \tag{E.3}$$

and a force controller:

$$G_{\text{lag}} = \frac{a_1 s + a_0}{b_1 s + 1} \tag{E.4}$$

Proportional

$$p(s) = k_p e(s) \tag{E.5}$$

Since the proportional part is not dependent on s it can be written directly as

$$\begin{aligned} e(s) &= E(z) = e(k) \\ p(k) &= k_p e(k) \end{aligned} \tag{E.6}$$

Lag

The lag part for the controller is described by the continuous function in (E.7).

$$G_{\text{lag}}(s) = \frac{a_1 s + a_0}{b_1 s + 1} e(s) \tag{E.7}$$

Here (E.7) is converted into the discrete z-domain again by using Tustin's approximation. This is shown in(E.8).

$$\begin{aligned} M_{\text{lag}}(z) &= \frac{a_1 \frac{2}{T}\frac{z-1}{z+1} + a_0}{b_1 \frac{2}{T}\frac{z-1}{z+1} + 1} E(z) \\ M_{\text{lag}}(z) &= \frac{a_0 T - 2a_1 + a_0 T z + 2a_1 z}{T - 2b_1 + Tz + 2b_1 z} E(z) \\ (T - 2b_1 + Tz + 2b_1 z) M_{\text{lag}}(z) &= (a_0 T - 2a_1 + a_0 T z + 2a_1 z) E(z) \\ \left((T - 2b_1) z^{-1} + T + 2b_1\right) M_{\text{lag}}(z) &= \left((a_0 T - 2a_1) z^{-1} + a_0 T + 2a_1\right) E(z) \end{aligned} \tag{E.8}$$

The difference equation of the lag controller can then be written as shown in E.9.

$$m_{\text{lag}}(k) = \frac{2a_1 + a_0 T}{2b_1 + T} e(k) - \frac{2a_1 - a_0 T}{2b_1 + a_0 T} e(k-1) + \frac{2b_1 - T}{2b_1 + T} m(k-1) \tag{E.9}$$

The final P and lag controller may then be implemented directly in the software.

Appendix F

Butterworth Filter Design

In the following appendix chapter a design of a n^{th} order Butterworth filter will be explained. The chapter is been made with inspiration from chapter two in [Dimopoulos, 2011]. The n^{th} order Butterworth filter utilises a modified low-pass filter function, where the transfer function is given as in (F.1).

$$H(s) = \frac{1}{\prod_{k=1}^{n} \frac{(s-s_k)}{\omega_c}} \tag{F.1}$$

where s_k is determined by (F.2).

$$s_k = \omega_c e^{\frac{j(2k+n-1)\pi}{2n}} \tag{F.2}$$

$$k = 1, 2, 3..., n \tag{F.3}$$

The poles given by (F.1) will be placed in a circle, equally spaced around the imaginary axis, with radius ω_c, which is illustrated in fig. F.1.

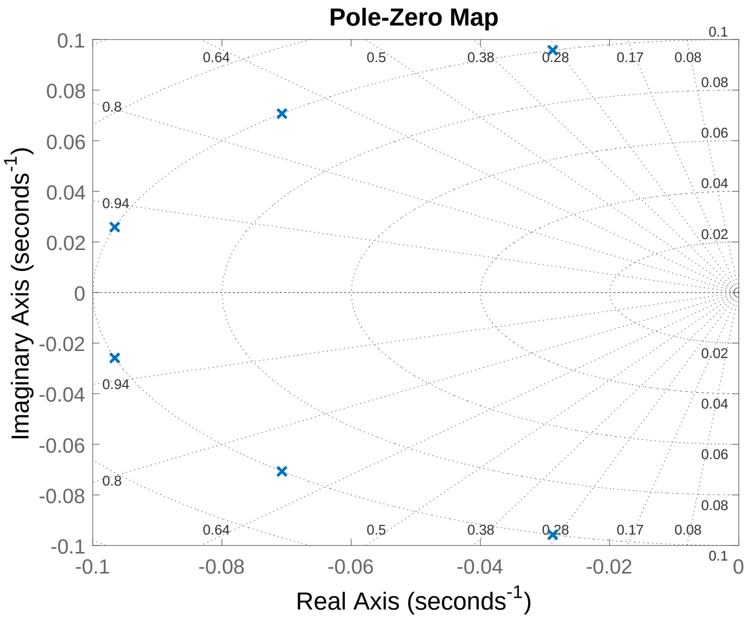


Figure F.1: Pole-location of 6^{th}-order Butterworth filter.

Normalised Butterworth

When the Butterworth filter is implemented in MATLAB, a normalisation procedure is used. This is done by writing the polynomial as a set of coefficients, which then are multiplied with the pole pairs complex conjugates. The polynomials are given in (F.4) and (F.5).

$$B_n(s) = \prod_{k=1}^{\frac{n}{2}} \left[s^2 - 2s\cos\left(\frac{2k+n-1}{2n}\pi\right) + 1 \right] \tag{F.4}$$

$$n = even$$

$$B_n(s) = (s+1) \prod_{k=1}^{\frac{n-1}{2}} \left[s^2 - 2s\cos\left(\frac{2k+n-1}{2n}\pi\right) + 1 \right] \tag{F.5}$$

$$n = odd$$

The normalised Butterworth filter transfer function, is then expressed as (F.6).

$$H(s) = \frac{1}{B_n(\alpha)} \tag{F.6}$$

where

$$\alpha = \frac{s}{\omega_c} \tag{F.7}$$

Where ω_c denotes the normalised cut-off frequency. For the filtering of the measurement data, a 6^{th} order Butterworth filter, cut-off frequency of 100 Hz is utilised. The normalised cut-off frequency at the sampling rate of 1000 Hz is then given as (F.8).

$$\omega_c = \frac{100\text{Hz}}{1000\text{Hz}} \tag{F.8}$$

From (F.4), the polynomial of the transfer function can then be determined by (F.9).

$$B_6(\alpha) = \left(\alpha^2 + 0.5176s + 1\right)\left(\alpha^2 + 1.4142s + 1\right)\left(\alpha^2 + 1.9319s + 1\right) \tag{F.9}$$

Where (F.9) and (F.7) is inserted into (F.6) which yields the final transfer function (F.10).

$$H(s) = \frac{1}{\left(\left(\frac{s}{\omega_c}\right)^2 + 0.5176\left(\frac{s}{\omega_c}\right) + 1\right)\left(\left(\frac{s}{\omega_c}\right)^2 + 1.4142\left(\frac{s}{\omega_c}\right) + 1\right)\left(\left(\frac{s}{\omega_c}\right)^2 + 1.9319\left(\frac{s}{\omega_c}\right) + 1\right)} \tag{F.10}$$

The transfer function results in the normalised frequency response shown in F.2.

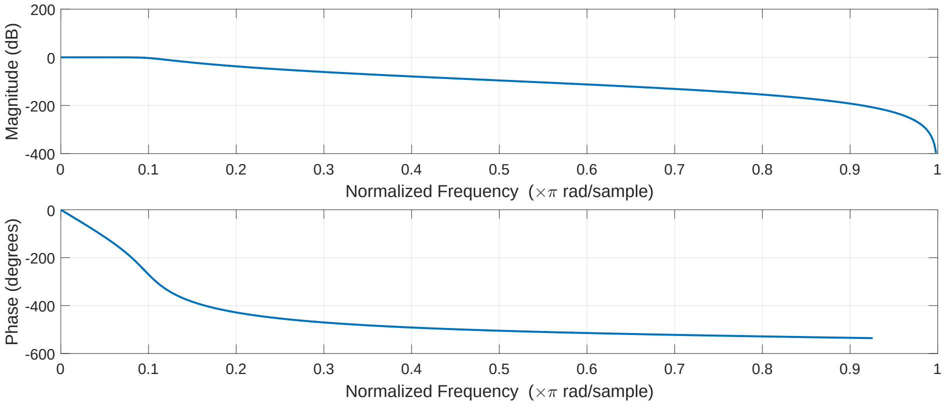


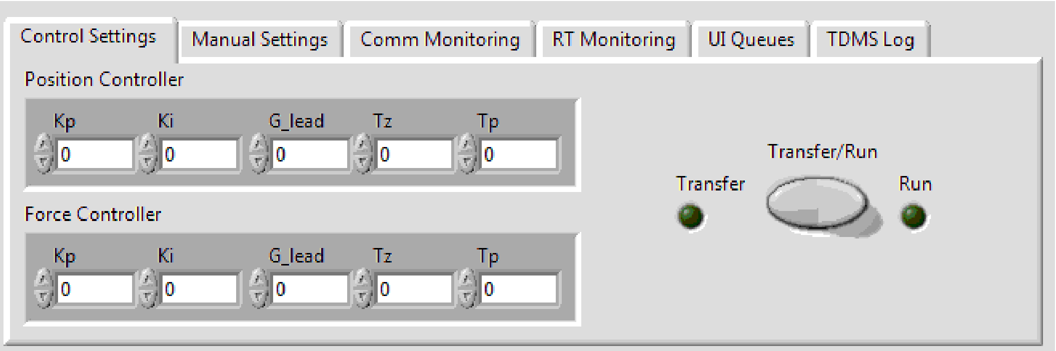
Figure F.2: 6^{th}-order Butterworth filter, with a normalised cut-off frequency of 0.1.

APPENDIX G

User Interface and Implementation

G.1 User Interface

In this section an explanation of the different user interfaces will be given. The first control panel, shown in fig. G.1, is the main user interface (UI), where the different controller gains can be specified, along with a Transfer/Run button. Where the transfer button, places the pistons in the start position of the trajectory, and the Run button launches the trajectory. The default mode is Transfer. Furthermore, during the transfer mode the output voltage limited to 1V out of 10V, so if a bad controller is typed in, the system is still protected.



Position Controller

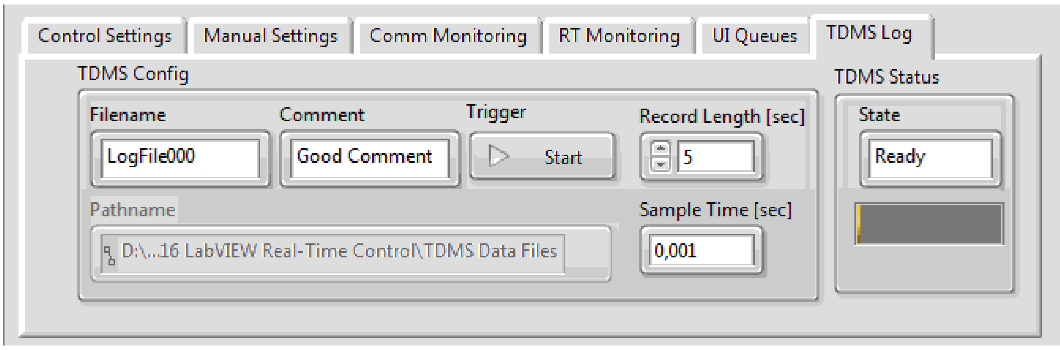
Kp	Ki	G_lead	Tz	Tp
0	0	0	0	0

Force Controller

Kp	Ki	G_lead	Tz	Tp
0	0	0	0	0

Figure G.1: Main user interface controls.

The Technical Data Management Streaming (TDMS) data logger is located in the fan *TDMS Log* and shown in fig. G.2. The user specifies the file name, a comment for the file, and the duration of the log. Lastly, the trigger button initiates the TDMS logger.



TDMS Config

Filename: LogFile000

Comment: Good Comment

Trigger

Record Length [sec]: 5

Pathname: D:\...16 LabVIEW Real-Time Control\TDMS Data Files

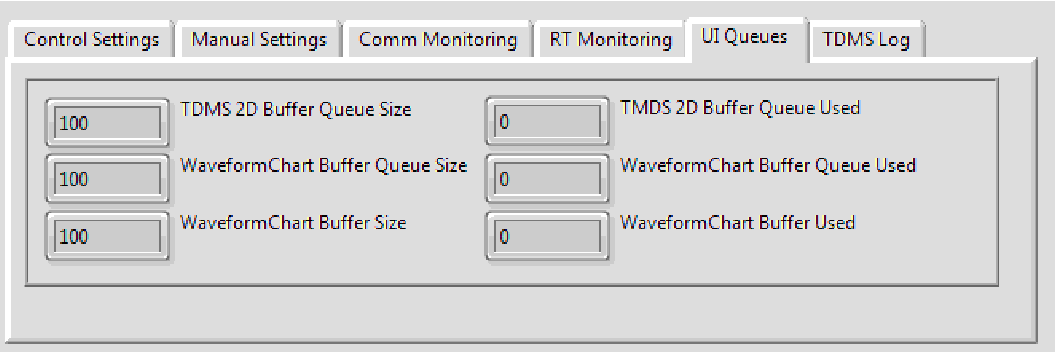
Sample Time [sec]: 0,001

TDMS Status

State: Ready

Figure G.2: Data logger UI.

The *UI Queues* tab shown in fig. G.3, gives the user information on the buffer status. Here the different buffer sizes can be seen and how much is being used at a given time during a run.



100 TDMS 2D Buffer Queue Size

100 WaveformChart Buffer Queue Size

100 WaveformChart Buffer Size

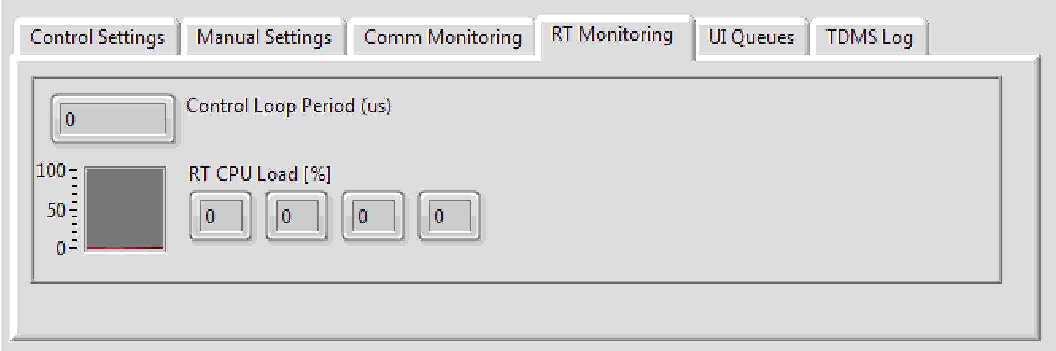
0 TMDS 2D Buffer Queue Used

0 WaveformChart Buffer Queue Used

0 WaveformChart Buffer Used

Figure G.3: Buffer health UI.

The last tab of interest is the *RT Monitoring* shown in fig. G.4. In this tab the Real Time (RT) Machines CPU load can be monitored, along with the time it takes the RT Loop to complete an iteration. If the code in the RT loop contains heavy or bad code, it will give a high CPU load, and maybe also a slower completion time of the RT Loop, hence a finished late error is triggered.

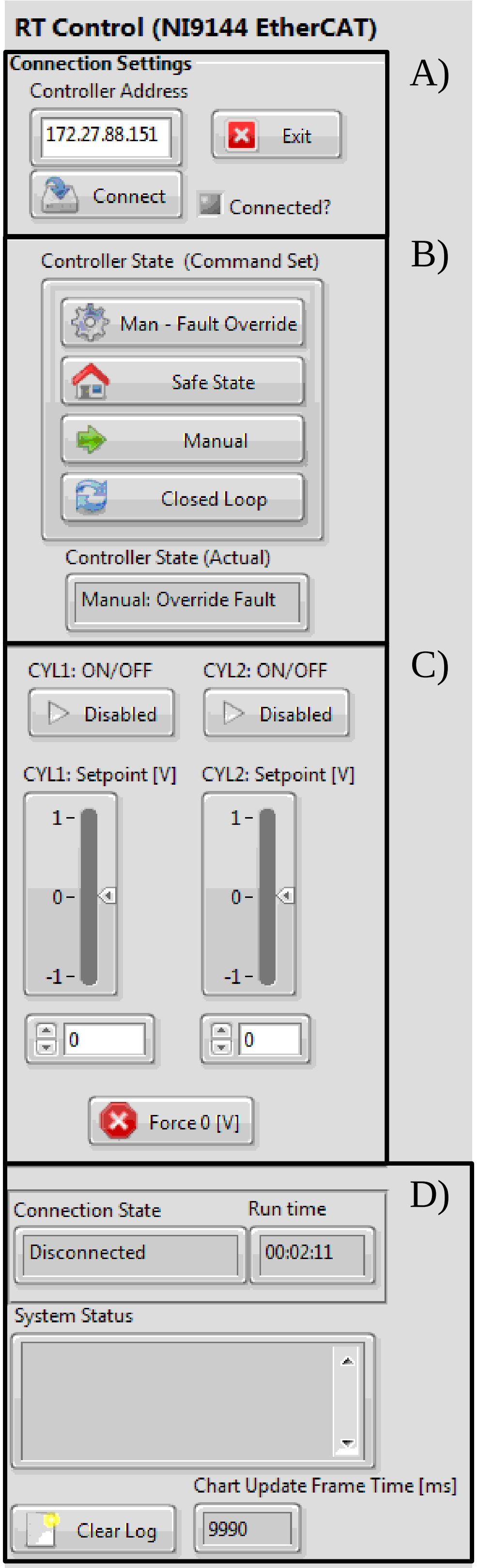


Control Loop Period (us): 0

RT CPU Load [%]: 0 0 0 0

Figure G.4: RT-Machine health and RT Loop time UI.

The main user control panel can be seen in fig. G.5, where the different states of the state machine may be controlled, along with the manual controls for the cylinders.



Controller Address: 172.27.88.151

Controller State (Actual): Manual: Override Fault

CYL1: Setpoint [V]: 0

CYL2: Setpoint [V]: 0

Connection State: Disconnected

Run time: 00:02:11

Chart Update Frame Time [ms]: 9990

Figure G.5: State Machine UI Controls.

A) The first task of the user is to initiate the TCP/IP connection to the RT-Machine. This is done by inputting the IP address of the RT-Machine, and hit connect. Furthermore, the connection can also be terminated by hitting the Exit button.

B) Here the State Machine can be controlled, and the current state is displayed. The default start-up state is the Safe State.

C) If the Manual or Manual Override state have been selected, then the two cylinders may be controlled manual, through the valves, from the controls shown. This is done, by feeding the valves with a direct voltage reference without a controller.

D) Here the connection state between the UI PC and the RT-Machine can be seen, along with a System Status Log, which displays all software errors or warnings that might be triggered during a run.

G.2 Implementation of Controllers

The controllers are implemented in two nested case structures, where the outer case determines whether the system is in a error or no error mode, The inner case determines which case of the state machine is active i.e. closed loop, manual, safe state or manual override, this inner nest case can be seen in fig. G.6.

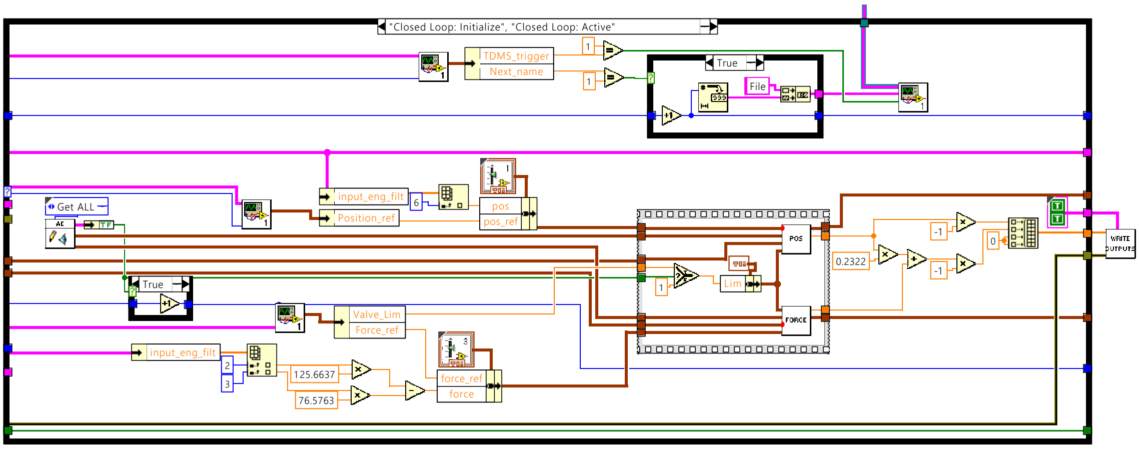


Figure G.6: Implementation of the controllers nested case structures.

In fig. G.7, the Flat Sequence structure is used to ensure that the controller subdiagrams are executed before the output variables are written to the output channels. Further more. it can be seen, how the data is being routed to the correct controller block, which contains the difference equations presented in E.

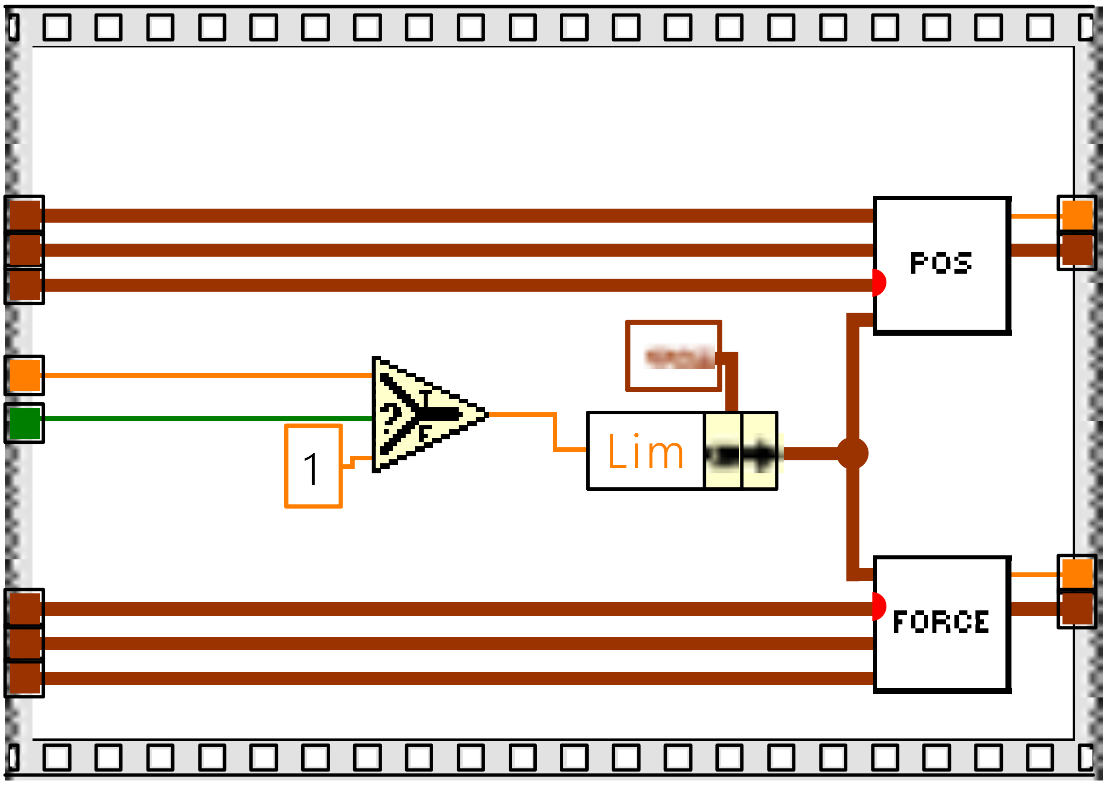
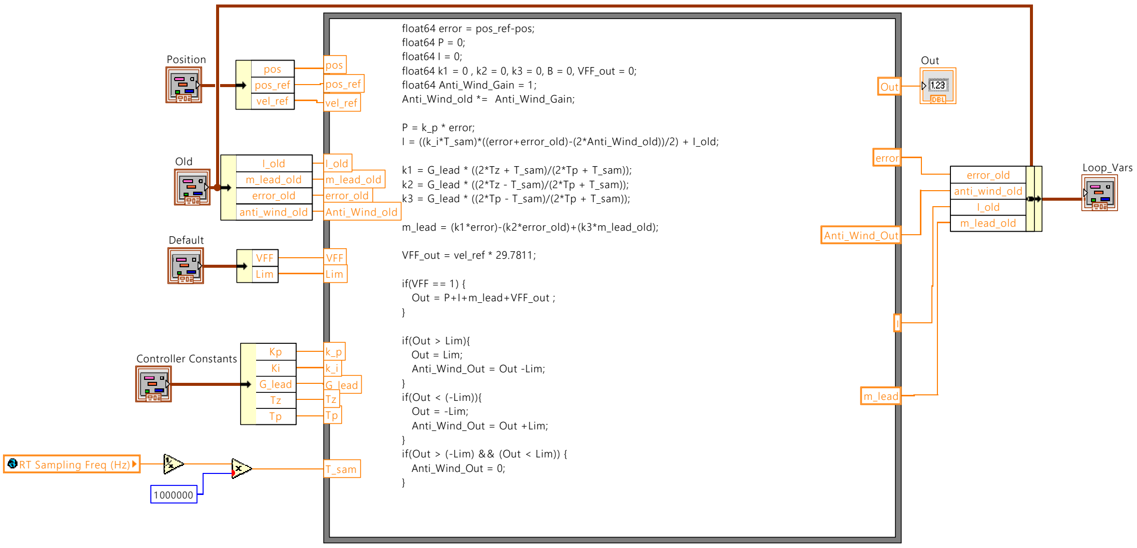


Figure G.7: Controllers and limitation selector, in flat sequence.

The implementation of the controllers difference equations derived in E is done, as shown in fig. G.8, where input data is shown in the left, and output data on the right. It should be noted, that the force controller is implemented like the position controller.



```
float64 error = pos_ref-pos;
float64 P = 0;
float64 I = 0;
float64 k1 = 0 , k2 = 0, k3 = 0, B = 0, VFF_out = 0;
float64 Anti_Wind_Gain = 1;
Anti_Wind_old *=  Anti_Wind_Gain;

P = k_p * error;
I = ((k_i*T_sam)*((error+error_old)-(2*Anti_Wind_old))/2) + I_old;

k1 = G_lead * ((2*Tz + T_sam)/(2*Tp + T_sam));
k2 = G_lead * ((2*Tz - T_sam)/(2*Tp + T_sam));
k3 = G_lead * ((2*Tp - T_sam)/(2*Tp + T_sam));

m_lead = (k1*error)-(k2*error_old)+(k3*m_lead_old);

VFF_out = vel_ref * 29.7811;

if(VFF == 1) {
   Out = P+I+m_lead+VFF_out ;
}

if(Out > Lim){
   Out = Lim;
   Anti_Wind_Out = Out -Lim;
}
if(Out < (-Lim)){
   Out = -Lim;
   Anti_Wind_Out = Out +Lim;
}
if(Out > (-Lim) && (Out < Lim)) {
   Anti_Wind_Out = 0;
}
```

Figure G.8: Implemented position controller on RT-Machine.

G.2.1 Automatic Data Logging for Parameter Identification Tests

To speed up the parameter identification tests presented in sec. 11.2, an automated test sequence have been implemented. The program utilises the controllers described above, where the user feeds in a desired test sequence trajectory, with the addition of a trigger value, to let the program know when to log data. The concept is shown in fig. G.9.

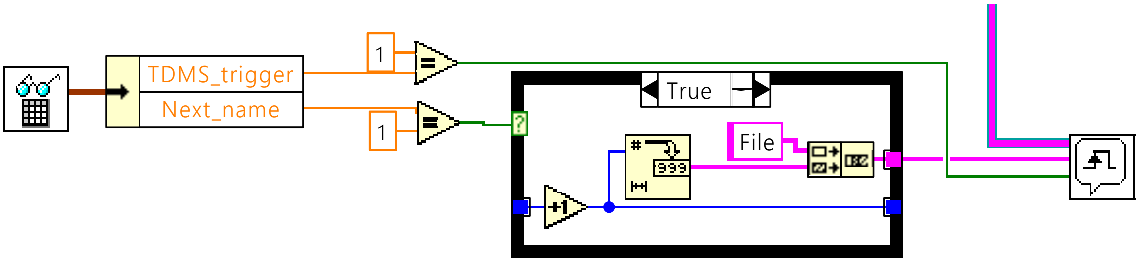


Figure G.9: Implemented of Technical Data Management Structure (TDMS) trigger, for automatic parameter identification.

When the trigger is received a five seconds logging period is activated, when the logging is finished the program iterates the name structure, and prepares the next trajectory run.

APPENDIX H

DATA ACQUISITION PRINTED CIRCUIT BOARD

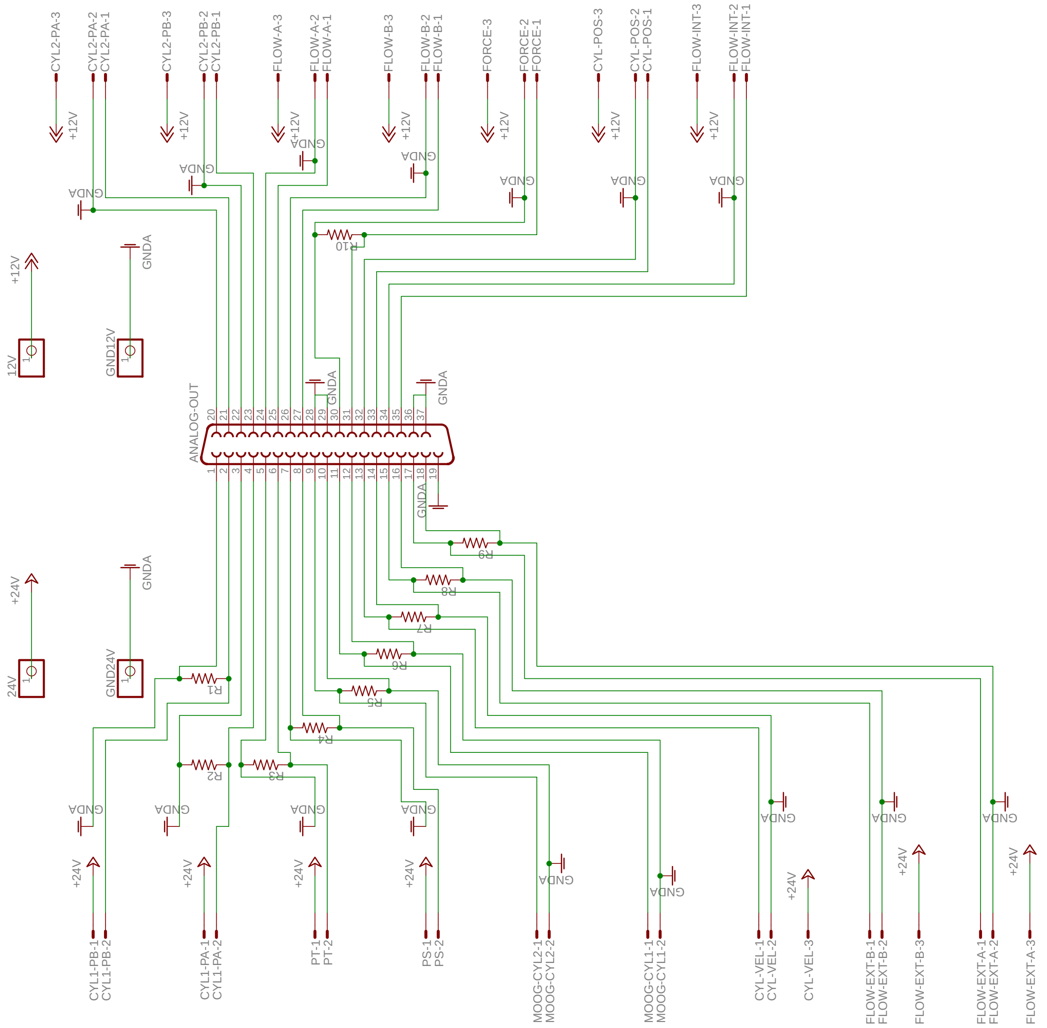


Figure H.1: Diagram of sensor wiring on the test set-up.

Appendix I

Result Graphs & Modelling Parameters

I.1 Evaluation of Residuals in the Extended FDD

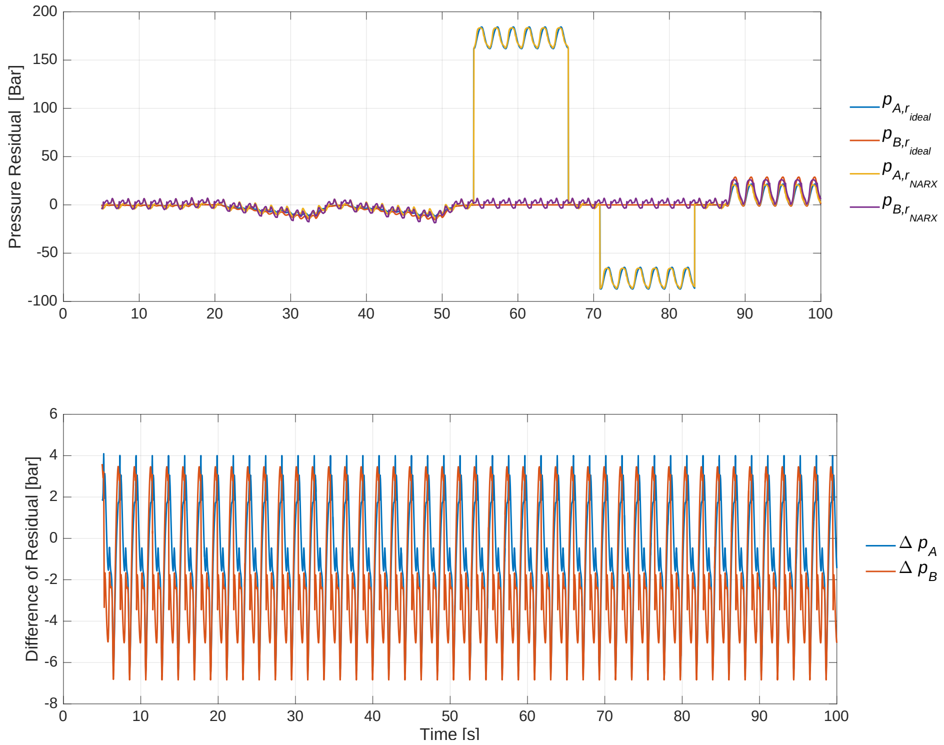


Figure I.1: **Sine wave input**: Simulated pressure residuals and the difference of these ($\Delta p_A = p_{A,r_{ideal}} - p_{A,r_{NARX}}$).

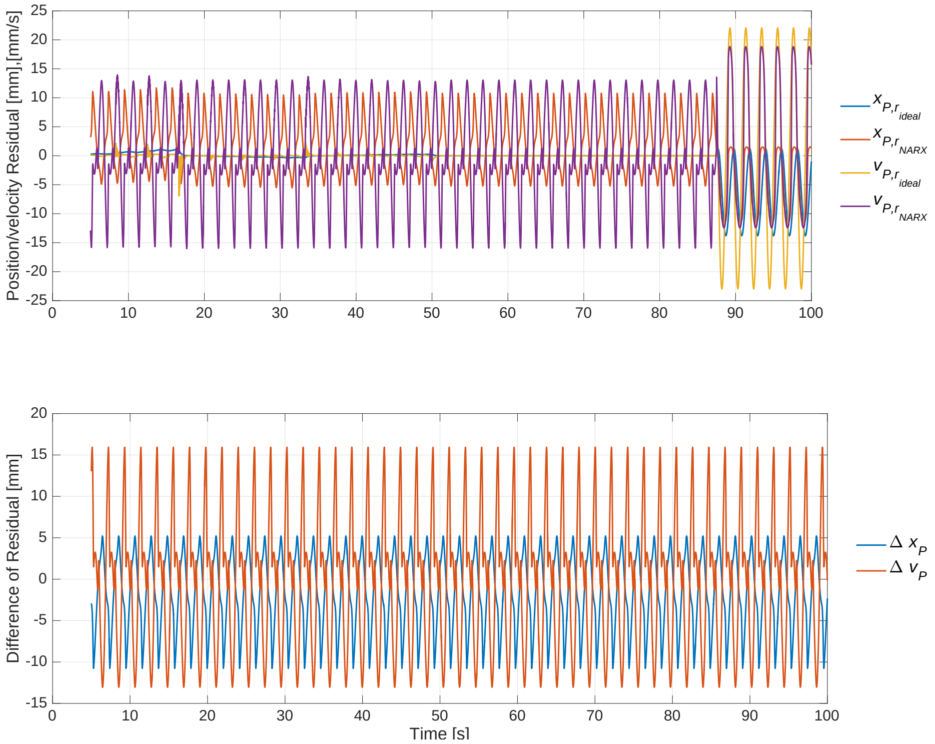


Figure I.2: **Sine wave input**: Simulated position and velocity residuals and the difference of these ($\Delta x_P = x_{P,r_{ideal}} - x_{P,r_{NARX}}$).

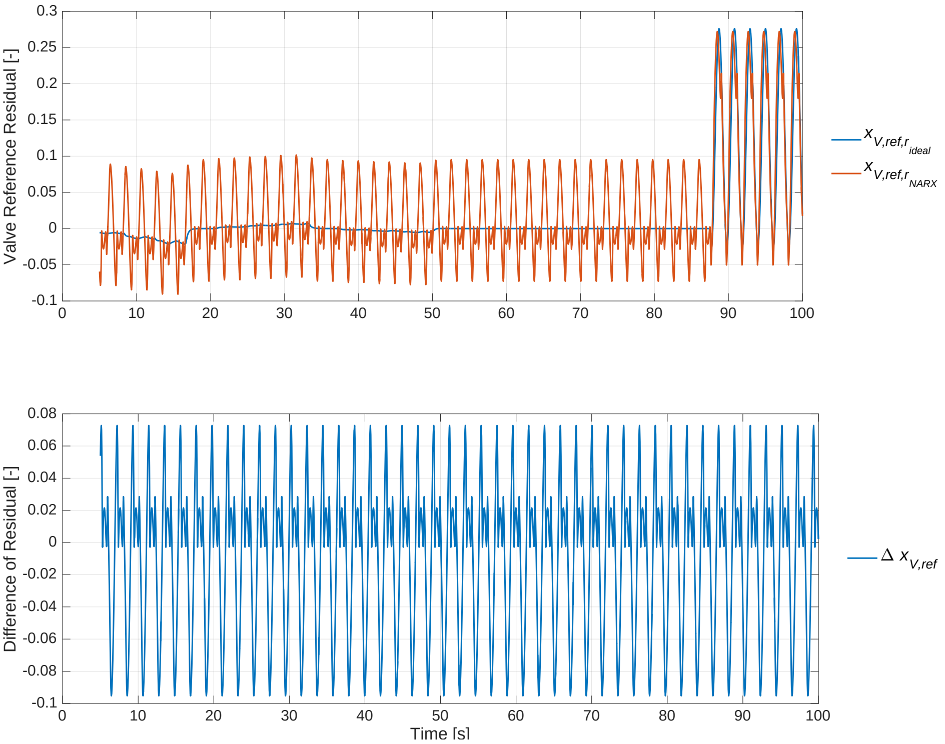


Figure I.3: **Sine wave input**: Simulated control signal residuals and the difference of these ($\Delta x_{V,ref} = x_{V,ref,r_{ideal}} - x_{V,ref,r_{NARX}}$).

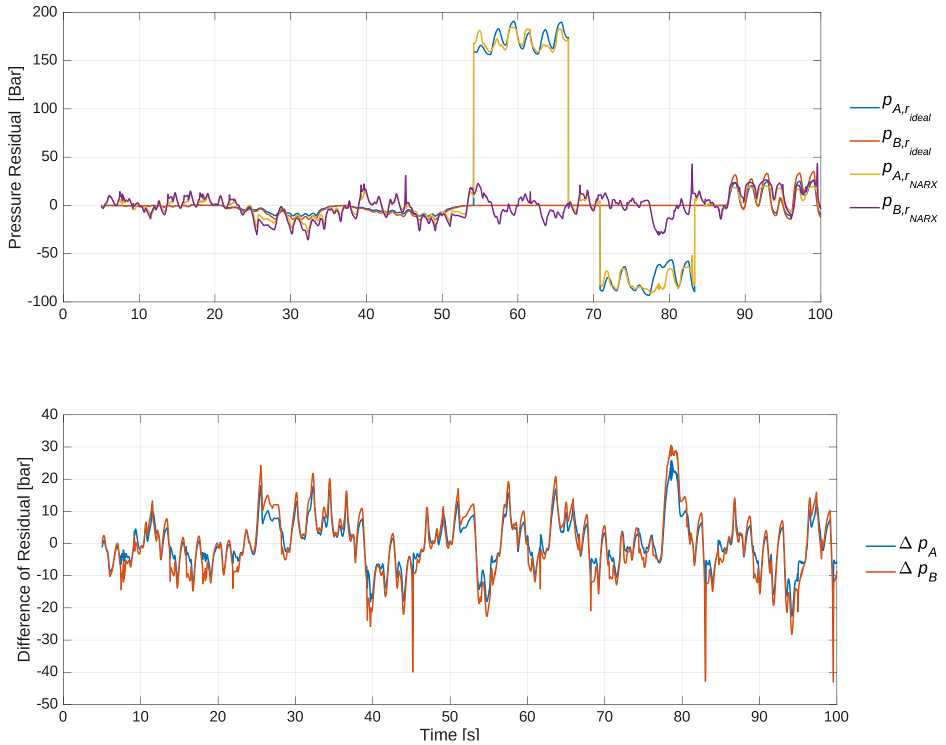


Figure I.4: **Sine wave with Gaussian white noise input**: Simulated pressure residuals and the difference of these ($\Delta p_A = p_{A,r_{ideal}} - p_{A,r_{NARX}}$).

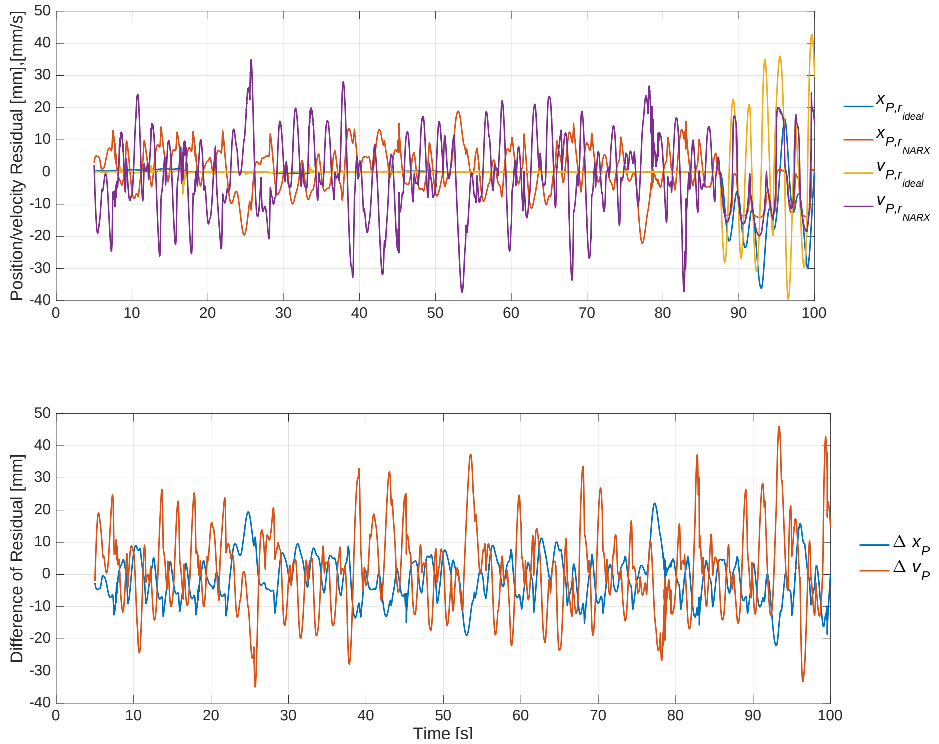


Figure I.5: **Sine wave with Gaussian white noise input**: Simulated position and velocity residuals and the difference of these ($\Delta x_P = x_{P,r_{ideal}} - x_{P,r_{NARX}}$).

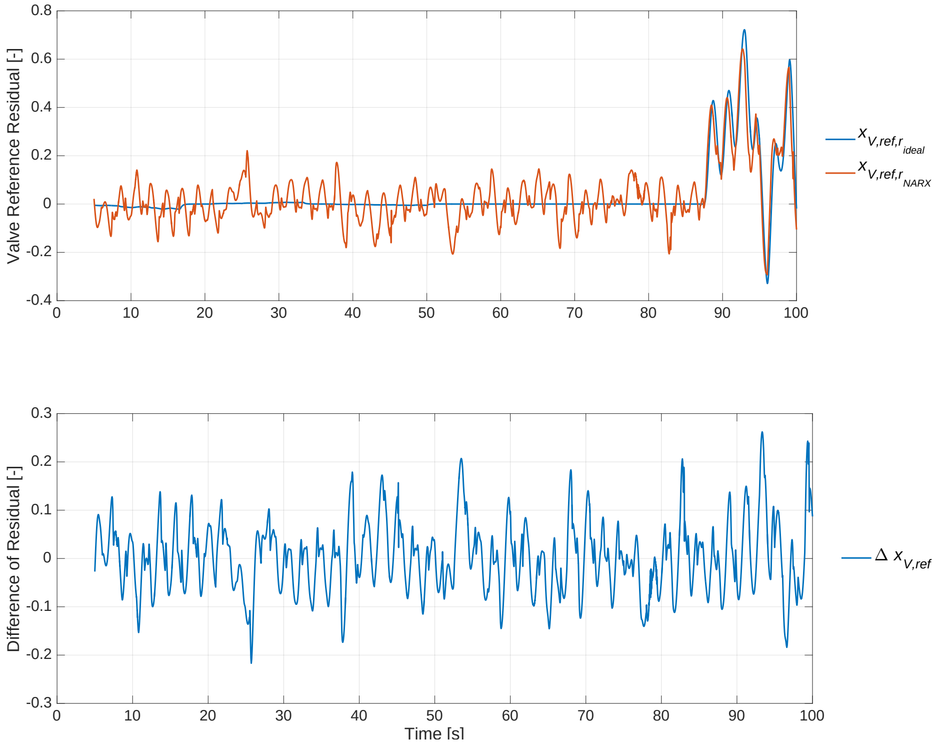


Figure I.6: **Sine wave with Gaussian white noise input**: Simulated control signal residuals and the difference of these ($\Delta x_{V,ref} = x_{V,ref,r_{ideal}} - x_{V,ref,r_{NARX}}$).

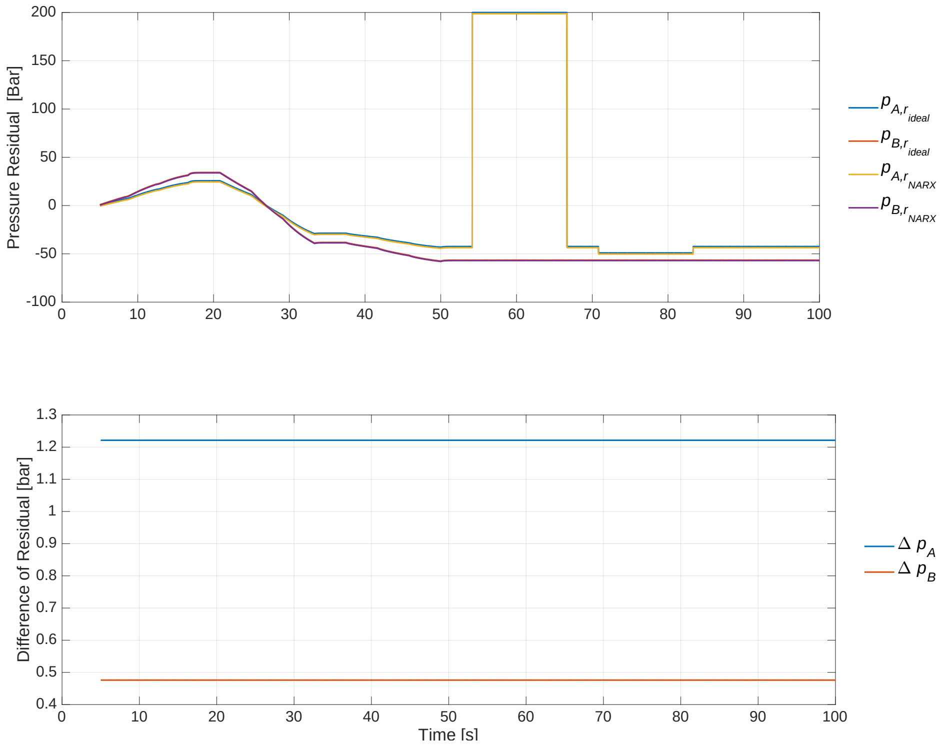


Figure I.7: **Constant input**: Simulated pressure residuals and the difference of these ($\Delta p_A = p_{A,r_{ideal}} - p_{A,r_{NARX}}$).

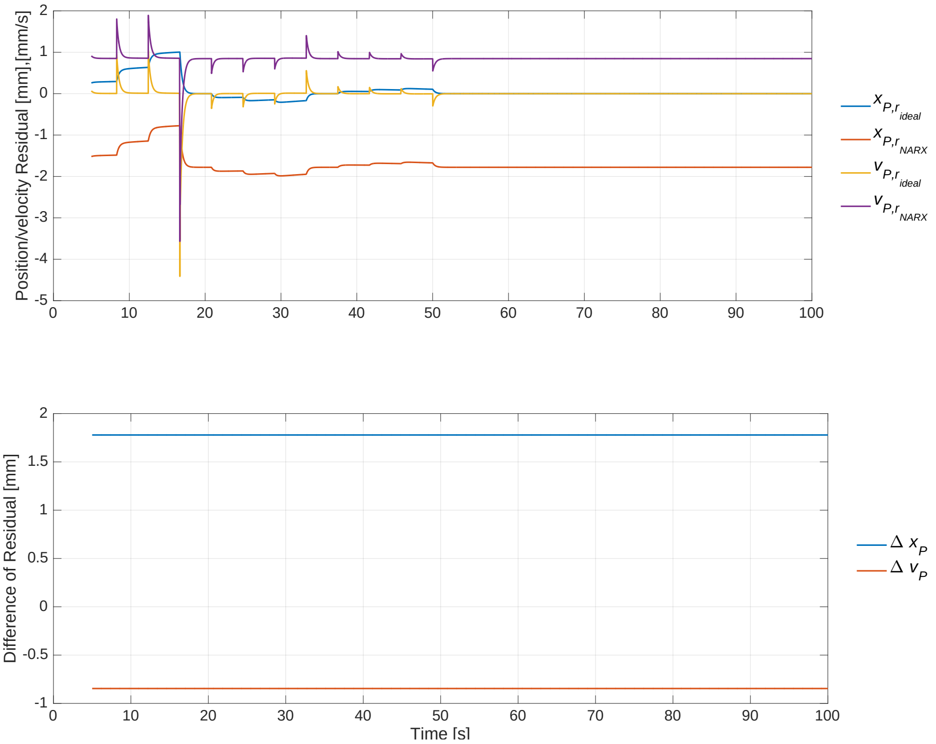


Figure I.8: **Constant input**: Simulated position and velocity residuals and the difference of these ($\Delta x_P = x_{P,r_{ideal}} - x_{P,r_{NARX}}$).

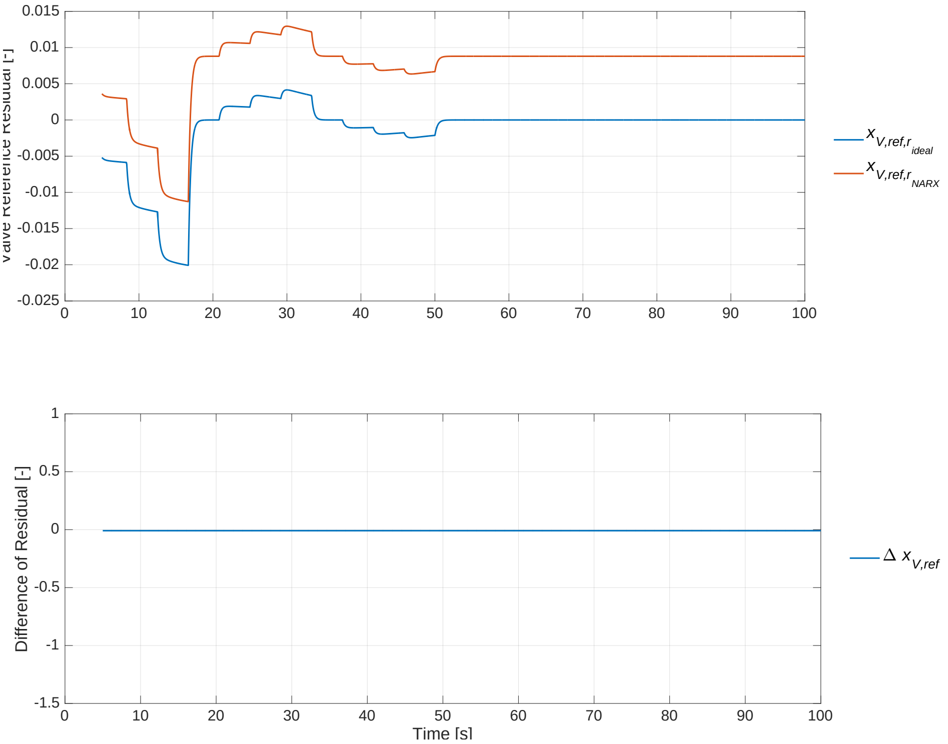


Figure I.9: **Constant input**: Simulated control signal residuals and the difference of these ($\Delta x_{V,ref} = x_{V,ref,r_{ideal}} - x_{V,ref,r_{NARX}}$).

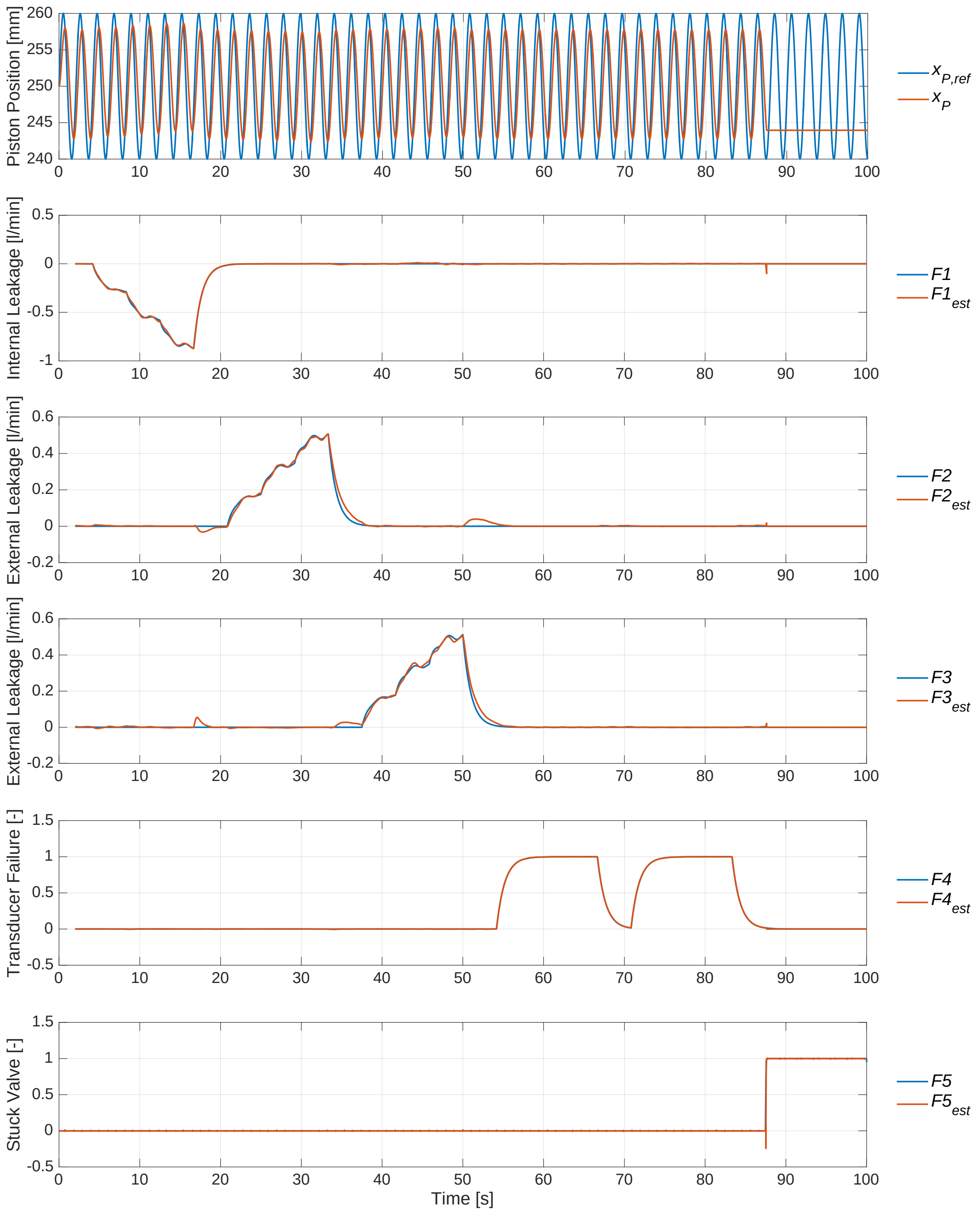


Figure I.10: **Sine wave input**: 2 second EMA of estimated and actual simulated faults when ideal residuals ($\mathbf{r}_{ideal}$) are applied along with noise on the measurements.

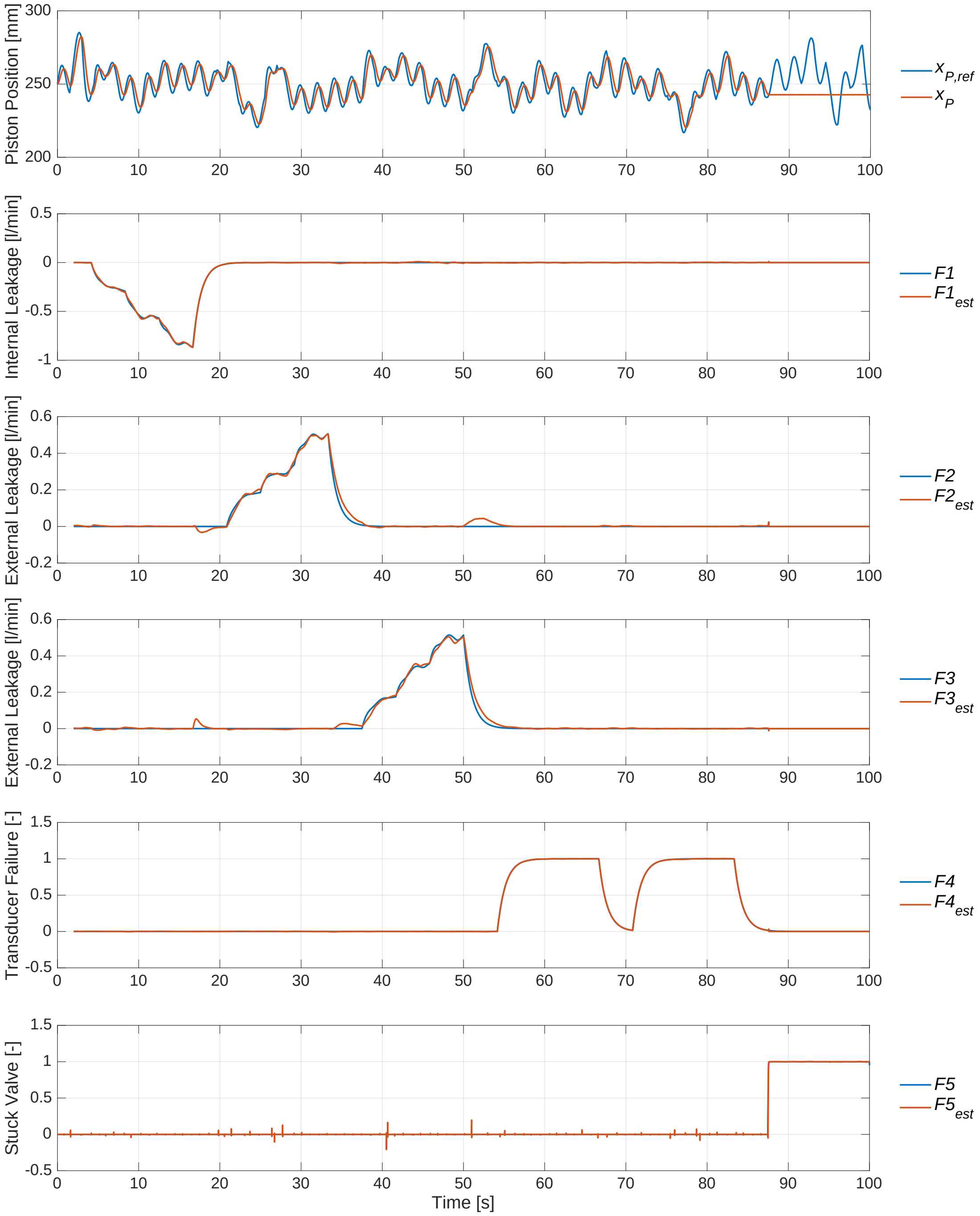


Figure I.11: **Sine wave with Gaussian white noise input**: 2 second EMA of estimated and actual simulated faults when ideal residuals ($\mathbf{r}_{ideal}$) are applied along with noise on the measurements.

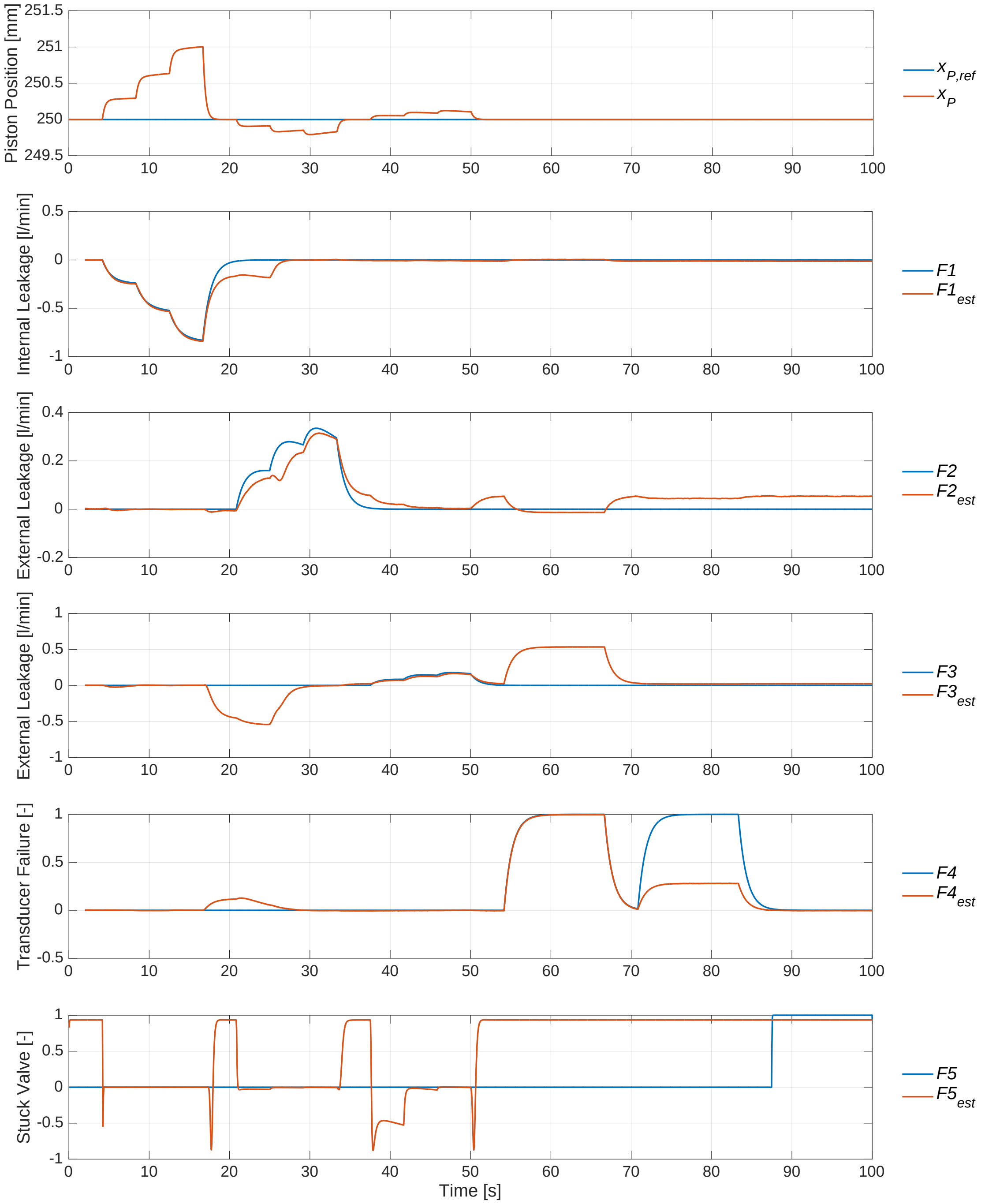


Figure I.12: **Constant input**: 2 second EMA of estimated and actual simulated faults when ideal residuals ($\mathbf{r}_{ideal}$) are applied along with noise on the measurements.

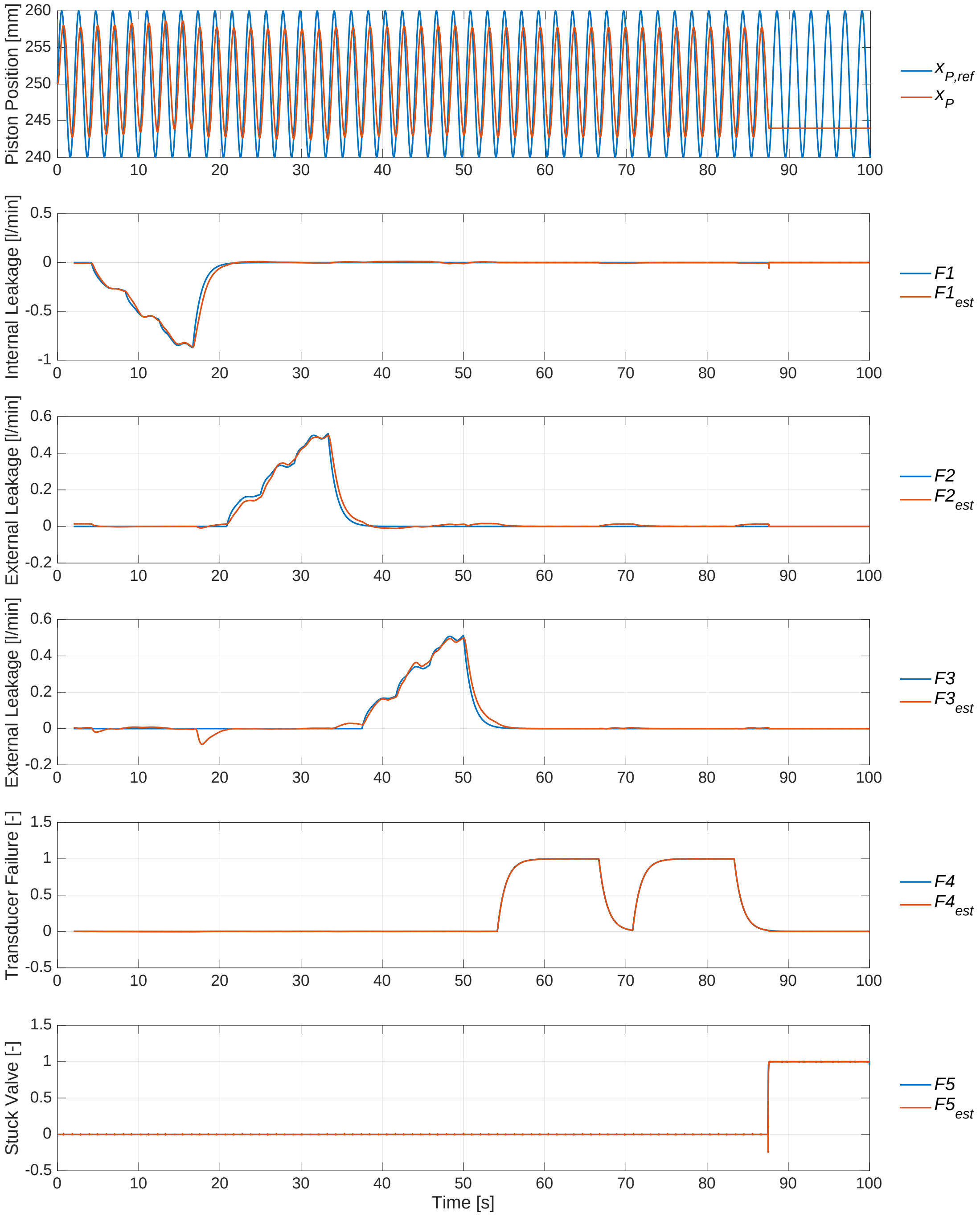


Figure I.13: **Sine wave input**: 2 second EMA of estimated and actual simulated faults when ideal residuals ($\mathbf{r}_{ideal}$) without velocity are applied along with noise on the measurements.

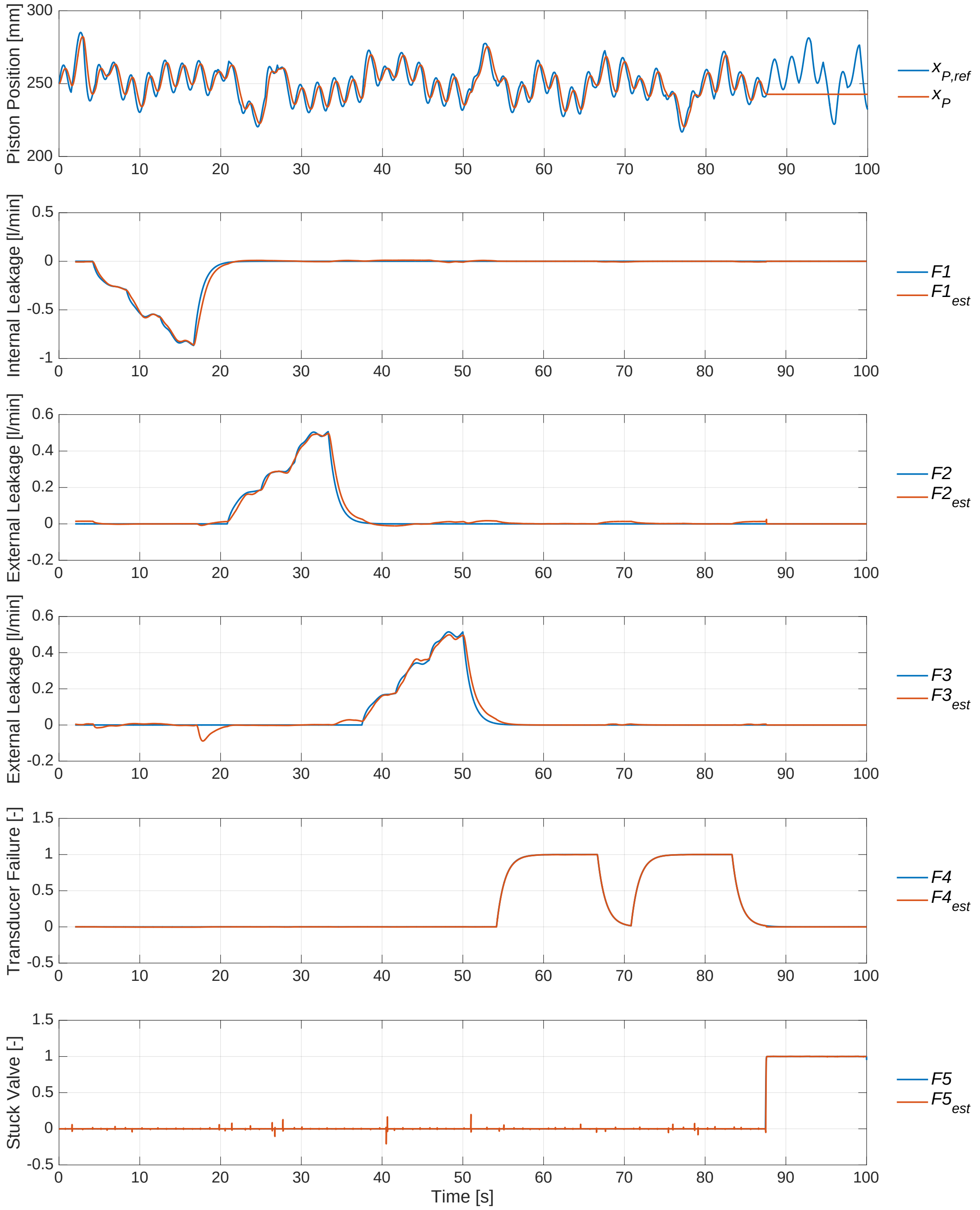


Figure I.14: **Sine wave with Gaussian white noise input**: 2 second EMA of estimated and actual simulated faults when ideal residuals ($\mathbf{r}_{ideal}$) without velocity are applied along with noise on the measurements.

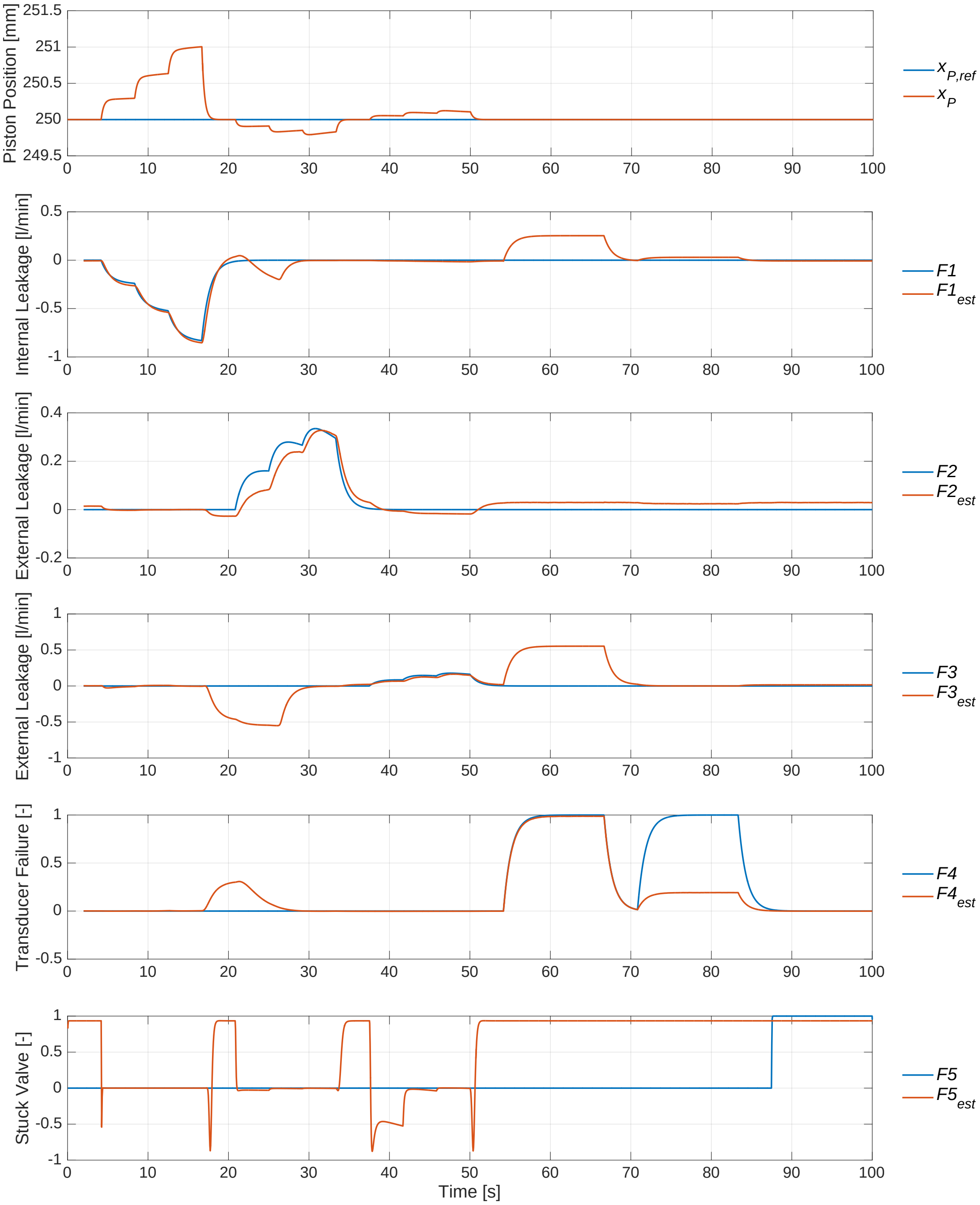


Figure I.15: **Constant input**: 2 second EMA of estimated and actual simulated faults when ideal residuals ($\mathbf{r}_{ideal}$) without velocity are applied along with noise on the measurements.

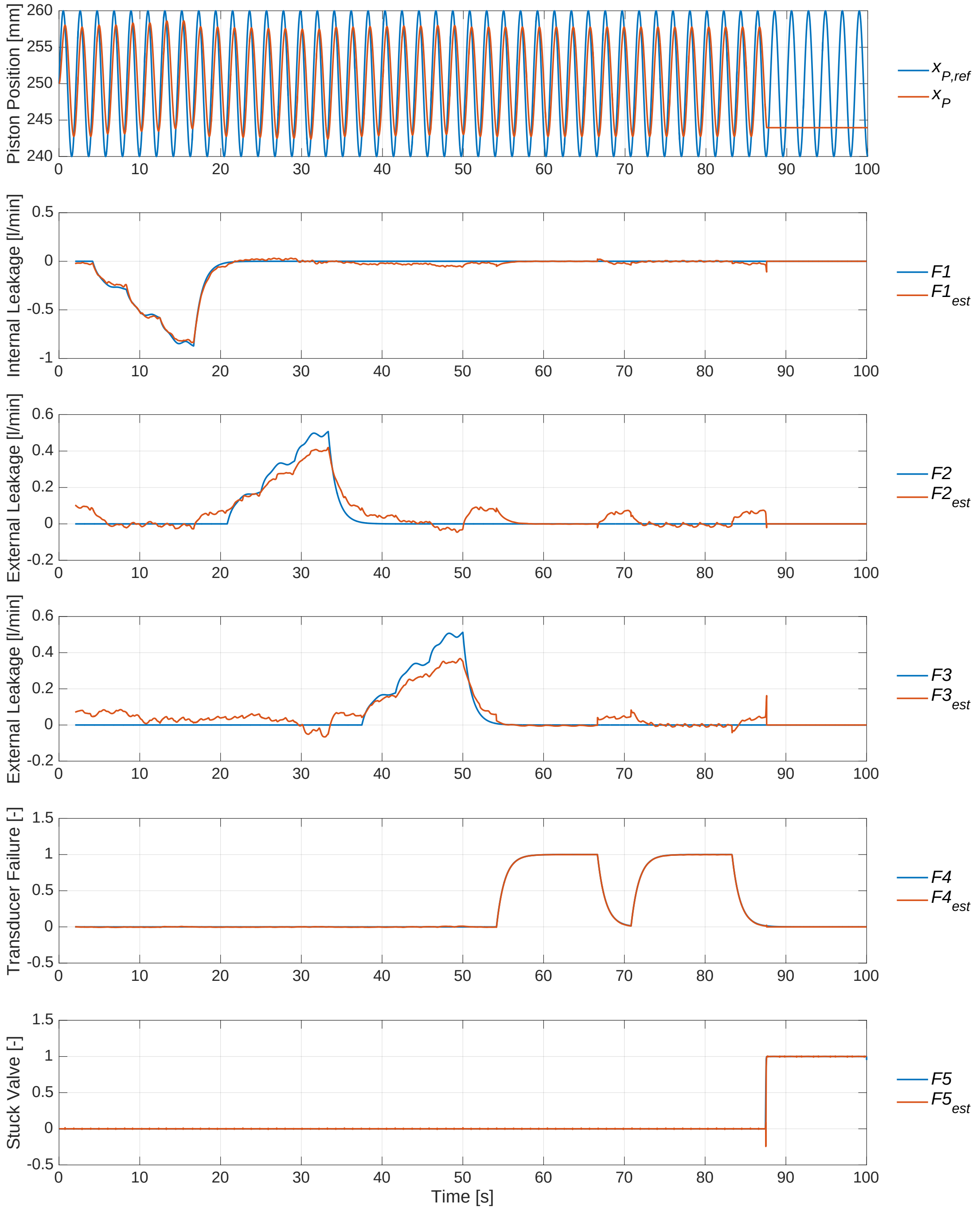


Figure I.16: **Sine wave input**: 2 second EMA of estimated and actual simulated faults for the data-driven scheme.

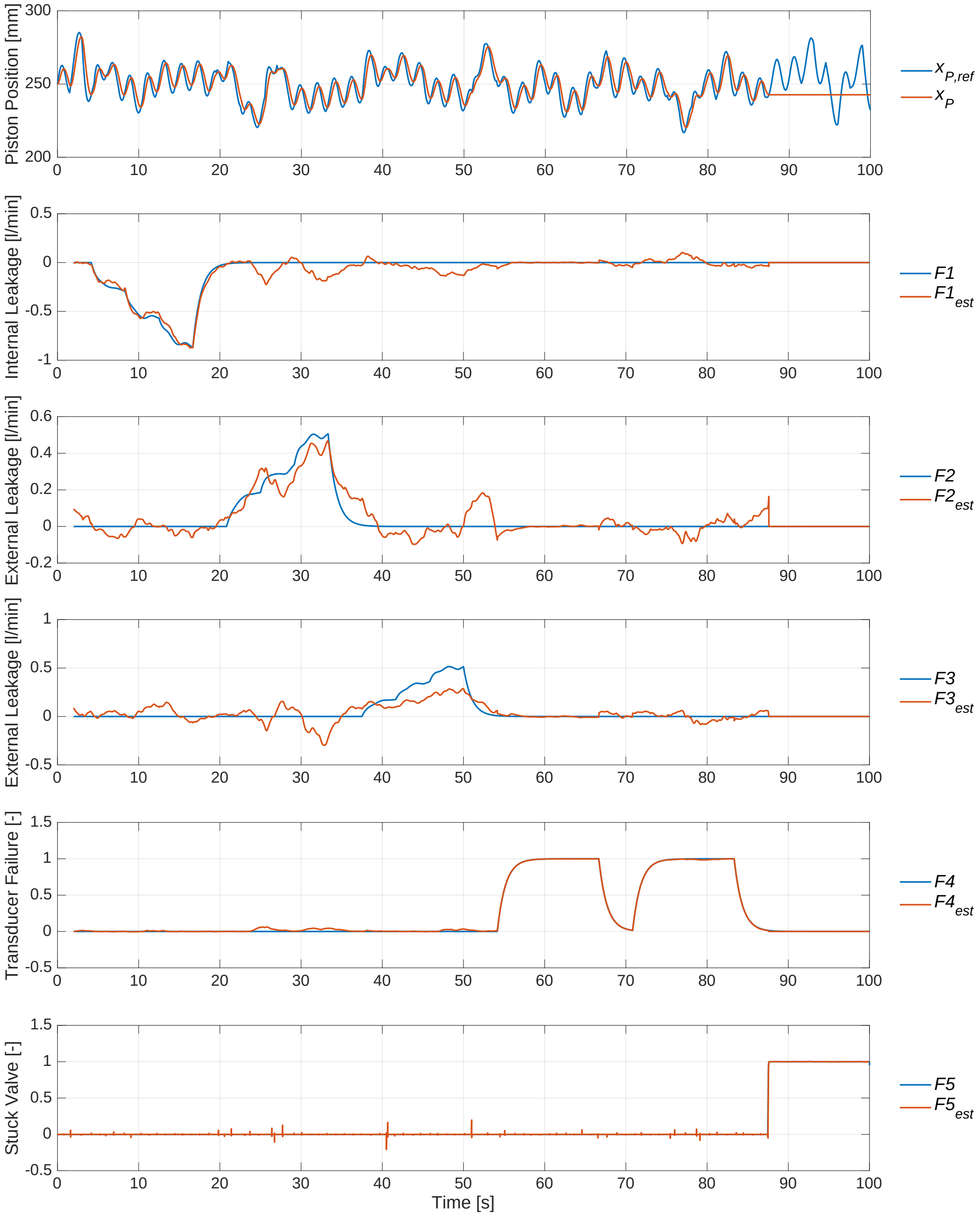


Figure I.17: **Sine wave with Gaussian white noise input**: 2 second EMA of estimated and actual simulated faults for the data-driven scheme.

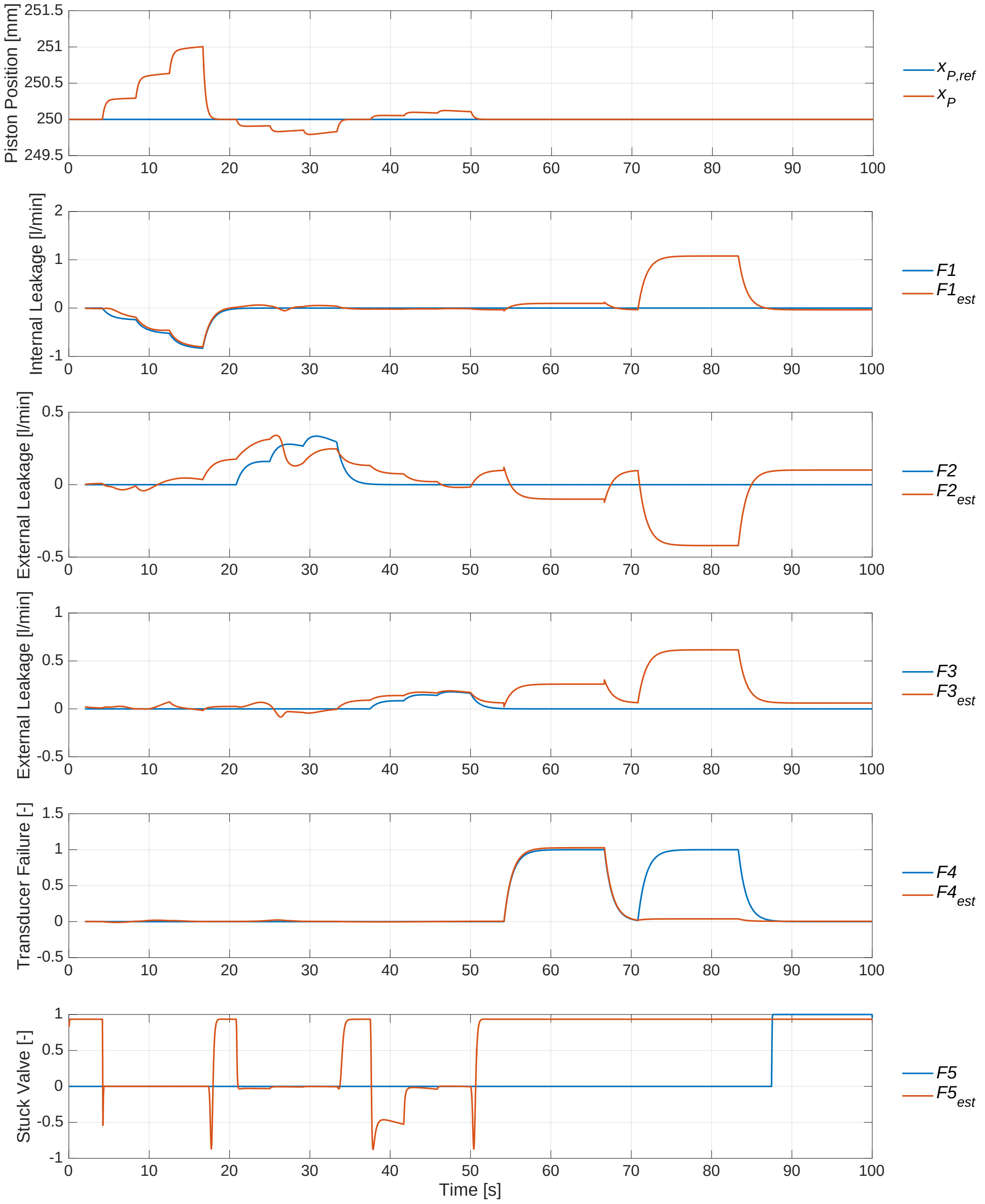


Figure I.18: **Constant input**: 2 second EMA of estimated and actual simulated faults for the data-driven scheme.

I.2 Evaluation of Multilayer Focused Time-Delay Schemes

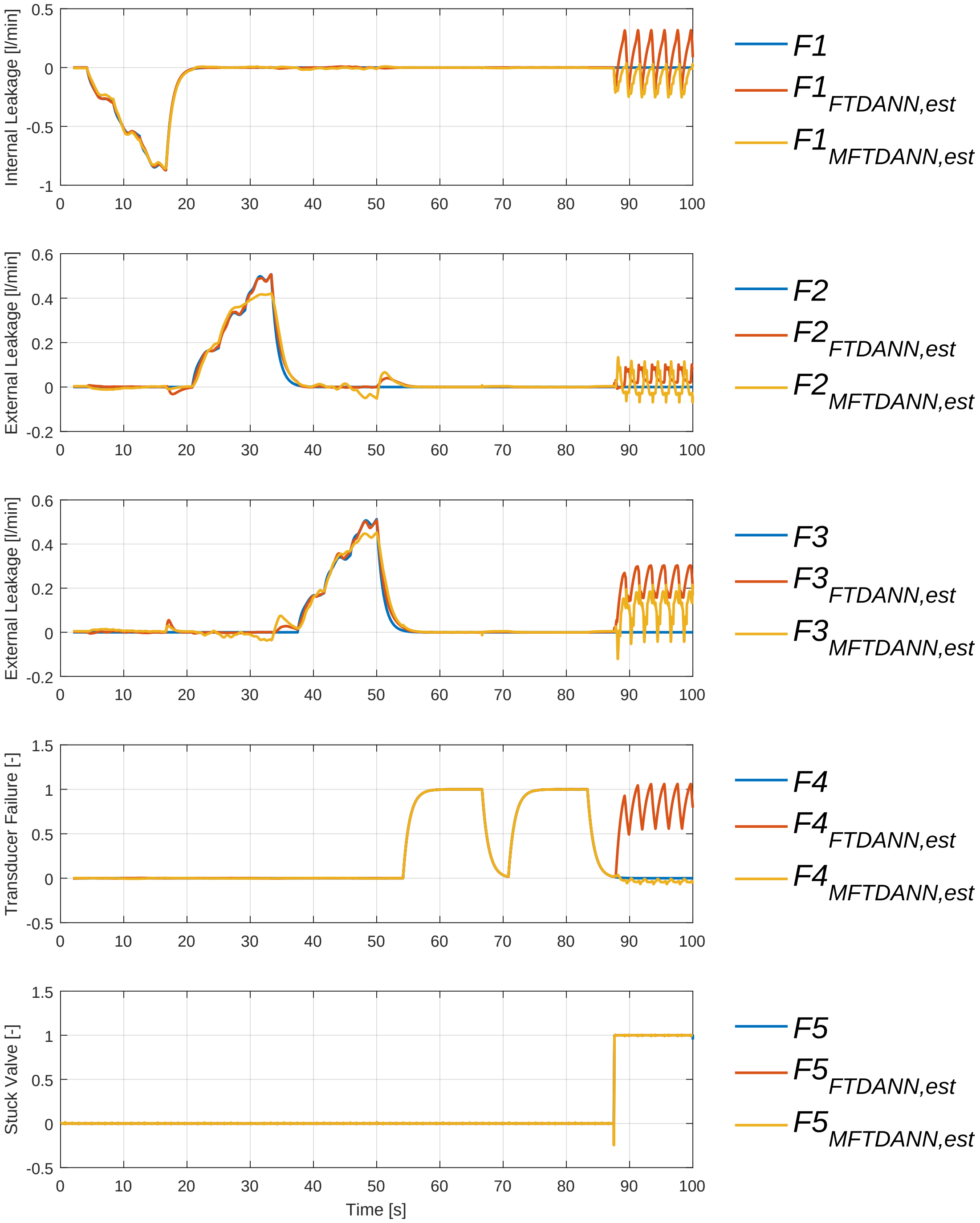


Figure I.19: **Sine wave input**: 2 second EMA of estimated and actual simulated faults for the multilayer model-based scheme.

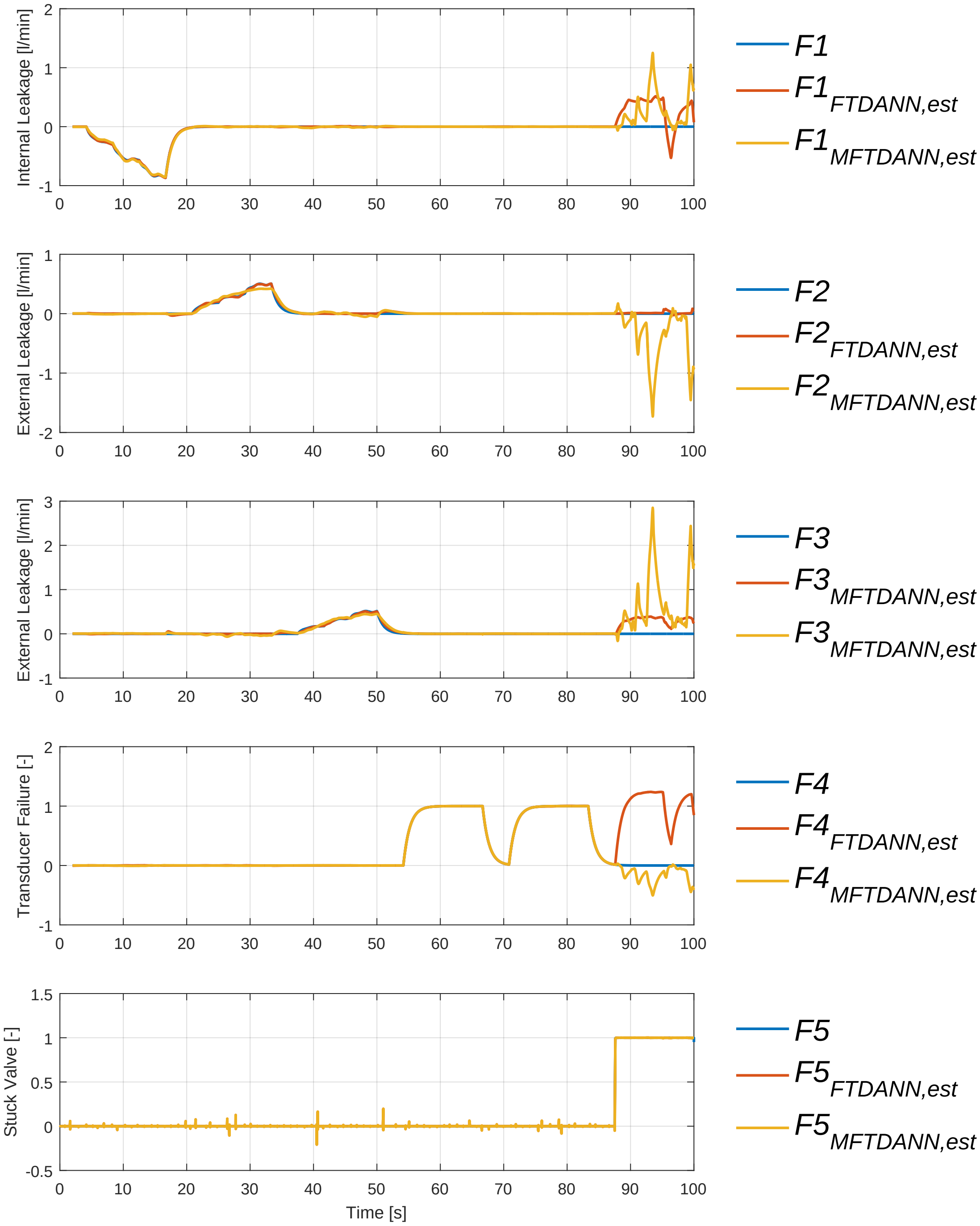


Figure I.20: **Sine wave with Gaussian white noise input**: 2 second EMA of estimated and actual simulated faults for the multilayer model-based scheme.

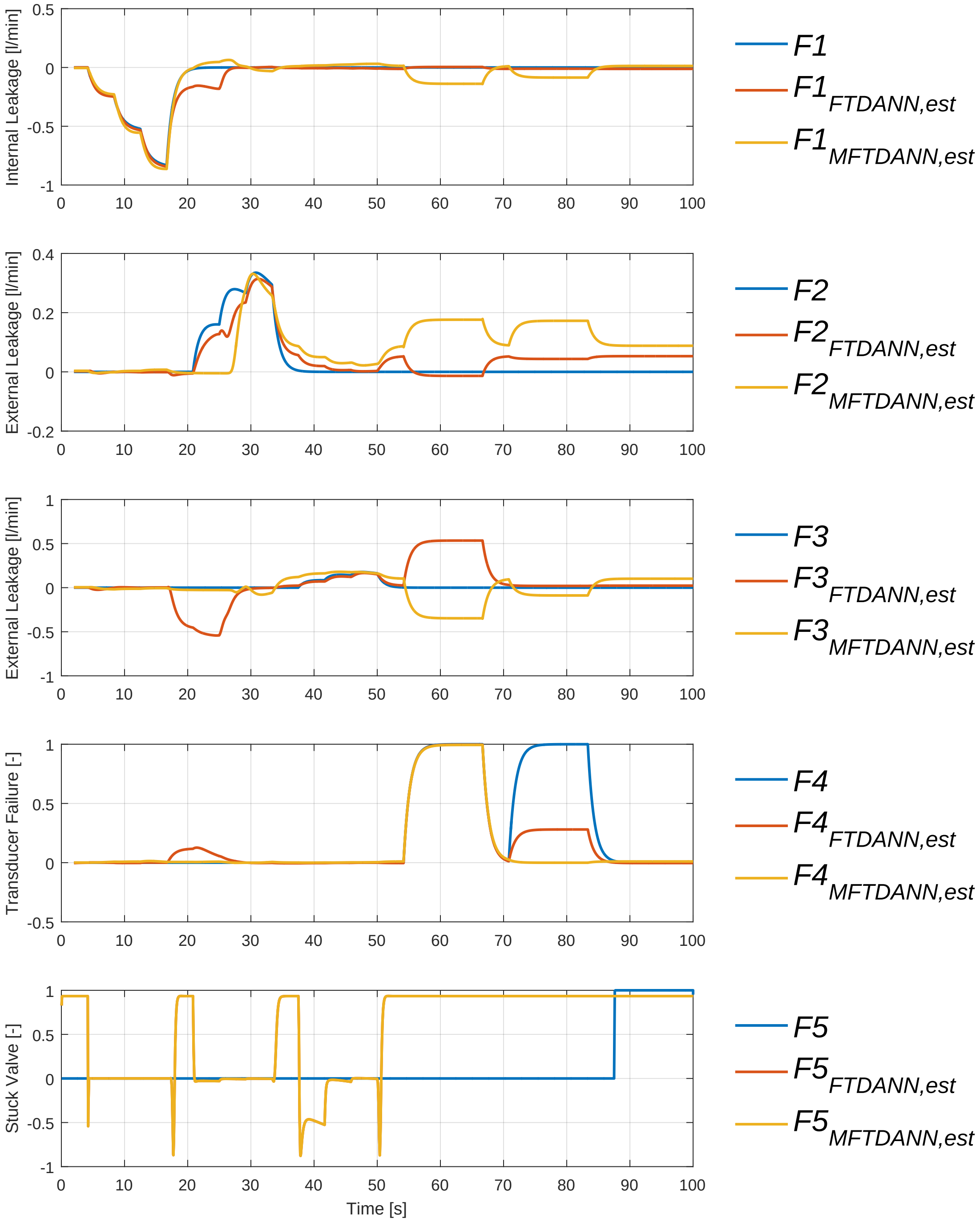


Figure I.21: **Constant input**: 2 second EMA of estimated and actual simulated faults for the multilayer model-based scheme.

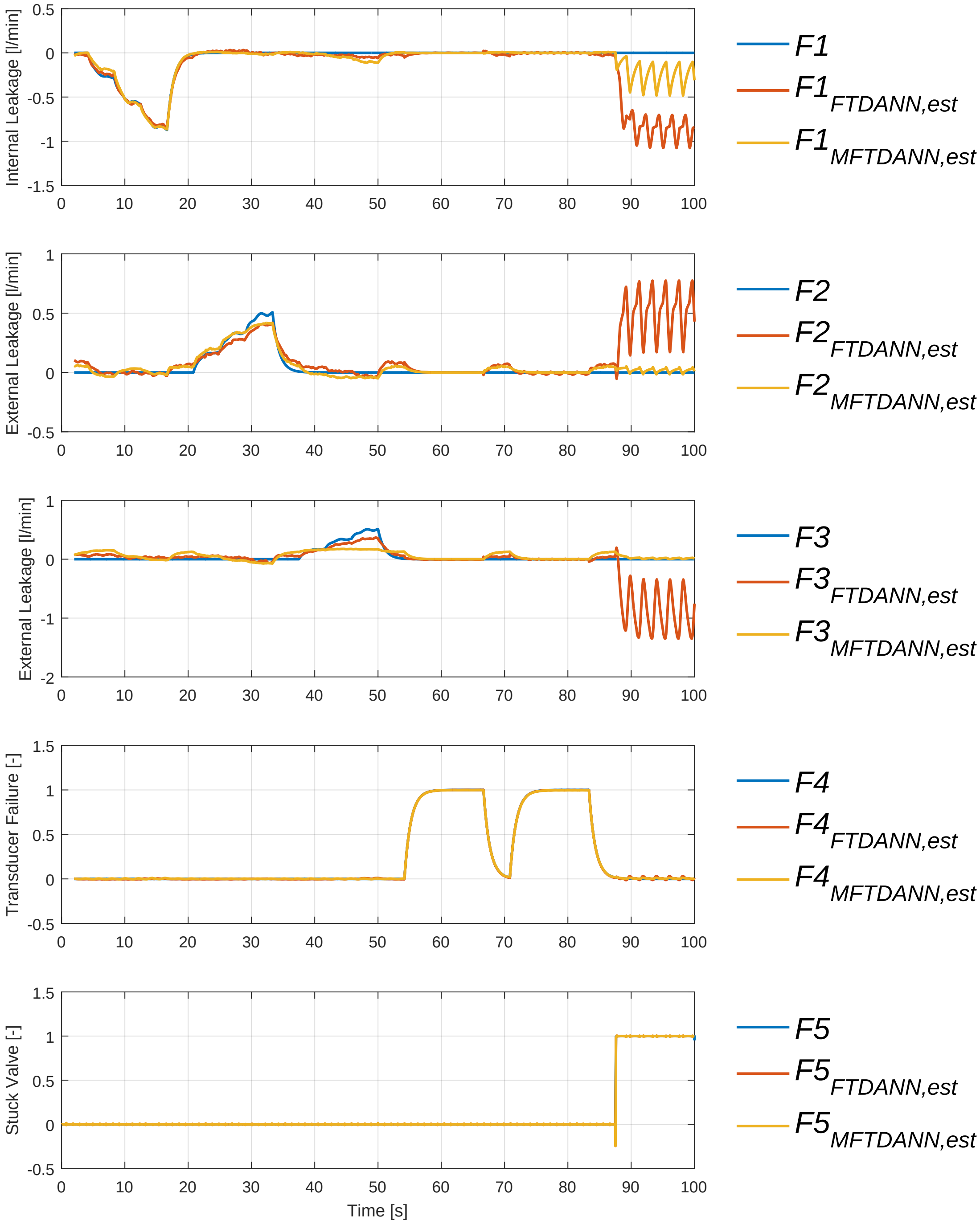


Figure I.22: **Sine wave input**: 2 second EMA of estimated and actual simulated faults for the multilayer data driven scheme.

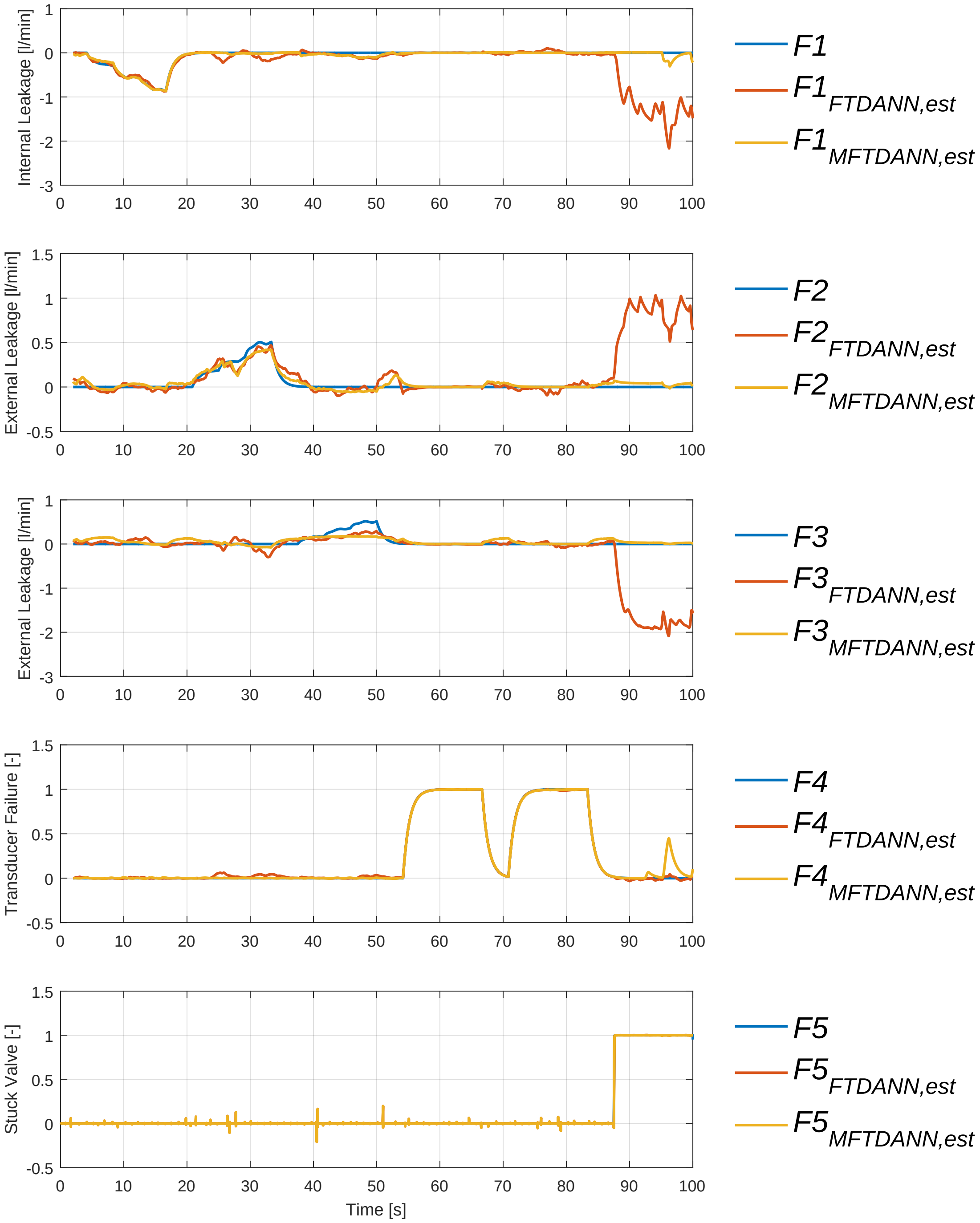


Figure I.23: **Sine wave with Gaussian white noise input**: 2 second EMA of estimated and actual simulated faults for the multilayer data driven scheme.

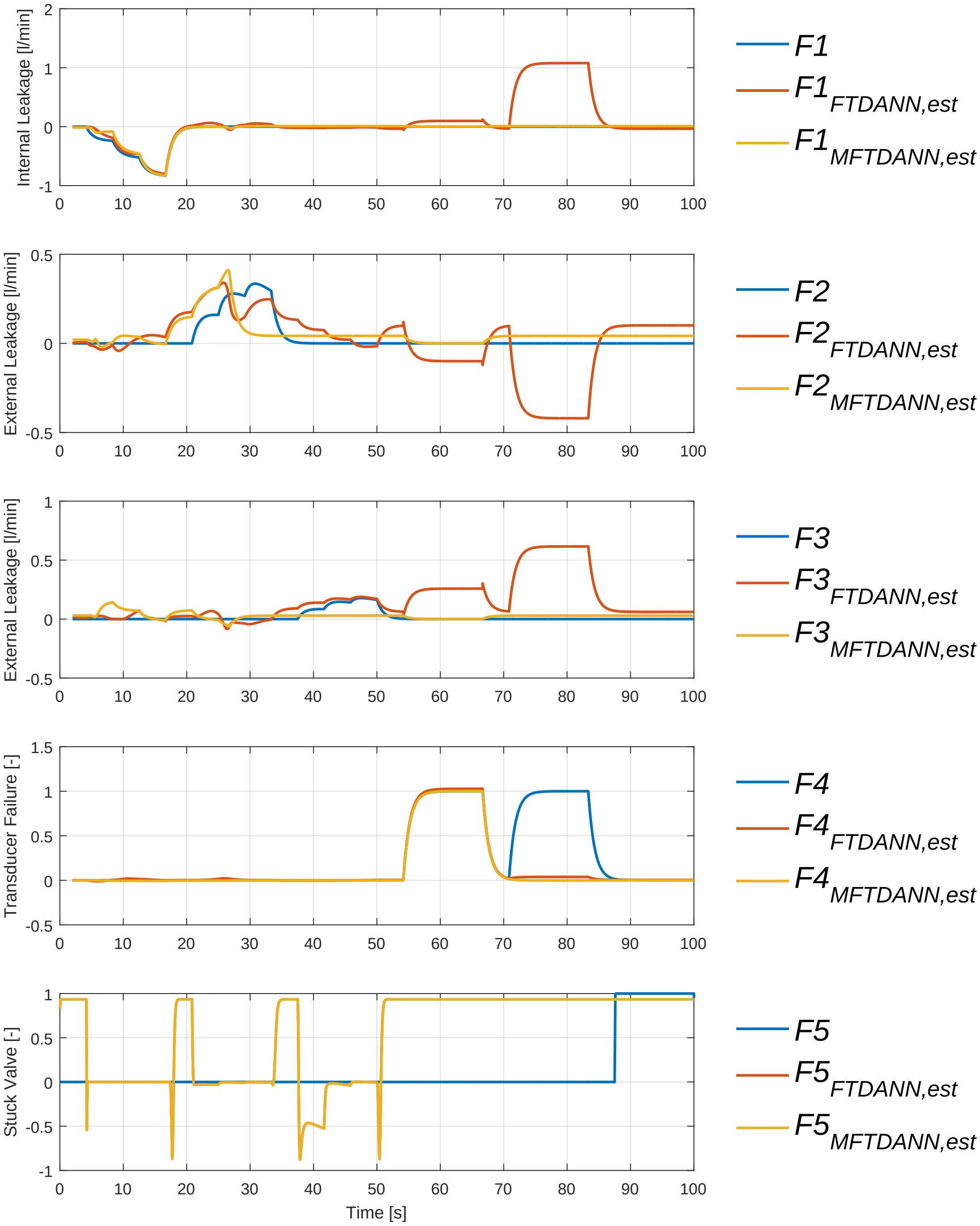


Figure I.24: **Constant input**: 2 second EMA of estimated and actual simulated faults for the multilayer data driven scheme.

I.3 Experimental Results

The obtained and applied parameters are shown below.

I.4 General Parameters

Index	Description	Value	Unit
α_{th}	Thermal expansion coefficient	0.007	[°C]
β_{oil}	Max fluid stiffness	13000	[bar]
c_{ad}	Adiabatic constant	1.4	[-]
C_D	Orifice discharge coefficient	0.61	[-]
k_{tan}	Velocity crossover constant for Coulomb friction	1e3	[-]
$\epsilon_{A,0}$	Relative volume of air in the fluid	0.01	[%]
p_S	Supply pressure	180	[bar]
p_T	Tank pressure	1	[bar]
T_{oil}	Fluid temperature	55	[°C]
ρ_{oil}	Fluid density at 15°C	910	[kg/m^3]
μ_{oil}	Fluid dynamic viscosity at 15°C	0.0425	[Ns/m^2]

Table I.1: General parameter values.

I.5 Parameters for Hydraulic Test Cylinder

Index	Description	Value	Unit
α_t	Effective area ratio	0.7500	[-]
$b_{c,n}$	Viscous friction coefficient of system in negative direction	2000	[Ns/m]
$b_{c,p}$	Viscous friction coefficient of system in positive direction	3000	[Ns/m]
$c_{s,n}$	Stribeck parameter in negative direction	20e-3	[m/s]
$c_{s,p}$	Stribeck parameter in positive direction	30e-3	[m/s]
$D_{P,t}$	Diameter of piston	80	[mm]
$D_{r,t}$	Diameter of piston rod	40	[mm]
$F_{c,n}$	Coulomb friction constant in negative direction	400	[N]
$F_{c,p}$	Coulomb friction constant in positive direction	1000	[N]
$F_{s,n}$	Maximum static friction constant in negative direction	300	[N]
$F_{s,p}$	Maximum static friction constant in positive direction	2200	[N]
$m_{P,t}$	Mass of moving parts	20	[kg]
$V_{dead,t}$	Dead volume	1.5e-4	[m^3]
$V_{hose,A,t}$	Hose volume in A	2.0e-4	[m^3]
$V_{hose,B,t}$	Hose volume in B	2.0e-4	[m^3]
$x_{P,max,t}$	Maximum stroke length	0.5	[m]

Table I.2: Parameters for the position servo cylinder.

I.6 Parameters for Hydraulic Load Cylinder

Index	Description	Value	Unit
α_l	Effective area ratio	0.6094	[-]
$x_{P,max,l}$	Maximum stroke length	0.4	[m]
$D_{P,l}$	Diameter of piston	40	[mm]
$D_{r,l}$	Diameter of piston rod	25	[mm]
$m_{P,l}$	Mass of moving parts	20	[kg]
$V_{dead,l}$	Dead volume	0.8e-4	[m^3]
$V_{hose,A,l}$	Hose volume in A	8.6e-5	[m^3]
$V_{hose,B,l}$	Hose volume in B	7.5e-5	[m^3]

Table I.3: Parameters for the load servo cylinder.

I.7 Parameters for MOOG Servo Valve 1 & 2

Index	Description	Value	Unit
k_V	Gain between normalised opening area and voltage drop	0.1	[m/V]
U_{DC}	The range of input control voltage	±10	[V]
Δp_n	Nominal pressure drop	35	[bar]
Q_n	Nominal flow (40 l/min)	6.67e-4	[m^3/s]
ζ	Damping of the servo valve	0.8	[-]
ω_n	Natural frequency of the servo valve	60	[Hz]

Table I.4: Parameters for the used MOOG D633-331 servo valves.

APPENDIX J

MATLAB code for Multilayer FTDANN

```matlab
%Create the concurrent data series used for training of
%the Multilayer Focused Time Delay Neural Network
run('Main');
nFaults = 7;
Mode = 1;
ParaData = 1;
FaultTrajectory = 1;
TrajectorySelect = 1;
Time = 20;
FaultSequence(1:nFaults) = 0;
i = 1;
h = waitbar(0,'Please wait...');
% Training Sequence: 1 = Sine+Gaussian, 2 = Sine, 3 = Constant, 4 = Steps
for TrajectorySelect = 1:3
    FaultSequence(1:nFaults) = 0;
    FaultSequence(1) = 1;
    sim('NonlinearModelRev10');
    y1(i,:) = sim2nndata(LeakageFlowsExtended.Data);
    u1(i,:) = sim2nndata(Input.Data);
    i = i+1;
    waitbar(i / 21)
    for Fault = 2:nFaults
        FaultSequence = circshift(FaultSequence,[1,1]);
        FaultTrajectory = 1;
        sim('NonlinearModelRev10');
        y1(i,:) = sim2nndata(LeakageFlowsExtended.Data);
        u1(i,:) = sim2nndata(Input.Data);
        i = i+1;
        waitbar(i / 21)
    end
end
close(h)
```

```
%% Merge Vectors
u_mul = catsamples(...
 ul(1,:), ul(2,:), ul(3,:), ul(4,:), ul(5,:), ul(6,:), ul(7,:),...
 ul(8,:), ul(9,:), ul(10,:), ul(11,:), ul(12,:), ul(13,:), ul(14,:),...
 ul(15,:), ul(16,:), ul(17,:), ul(18,:), ul(19,:), ul(20,:), ul(21,:),...
  'pad');
y_mul = catsamples(...
 yl(1,:), yl(2,:), yl(3,:), yl(4,:),yl(5,:), yl(6,:), yl(7,:),...
 yl(8,:), yl(9,:), yl(10,:), yl(11,:),yl(12,:), yl(13,:), yl(14,:),...
 yl(15,:), yl(16,:), yl(17,:), yl(18,:),yl(19,:), yl(20,:), yl(21,:),...
  'pad');
%% Configuration of Multi Layer FTDANN
% Create the ANN object
net = distdelaynet({1:6,1:6},6);
% Layers
net.numInputs = 1;
net.numLayers = 3;
% Bias connections
net.biasConnect = [1; 1; 1];
%Weights connection
net.inputConnect = [1; 1 ; 1];
%Input Connect
net.inputConnect = [1; 0; 0];
%Layer Connect
net.layerConnect = [0 0 0; 1 0 0; 0 1 0];
%Output Connect
net.outputConnect = [0 0 1];
%Hidden layer 1
net.layers{1}.size = 12;
net.layers{1}.transferFcn = 'tansig';
net.layers{1}.initFcn = 'initnw';
net.layers{1}.name = 'Hidden';
%Hidden layer 2
net.layers{2}.size = 12;
net.layers{2}.transferFcn = 'tansig';
net.layers{2}.initFcn = 'initnw';
net.layers{2}.name = 'Hidden';
%Output layer
net.layers{3}.size = 4;
net.layers{3}.name = 'Output';
%Input layer
net.inputs{1}.exampleInput = size(u_mul);
net.inputs{1}.size = 5;
%Config training options
net.initFcn = 'initlay';
net.performFcn = 'mse';
net.trainFcn = 'trainlm';
net.divideFcn = 'dividerand';
net.plotFcns = {'plotperform','plottrainstate'};

%Initialise ANN
net = init(net);
view(net)

%% Training
net = train(net,u_mul,y_mul);

%% Deploy to Simulink
gensim(net,samplesize)
```