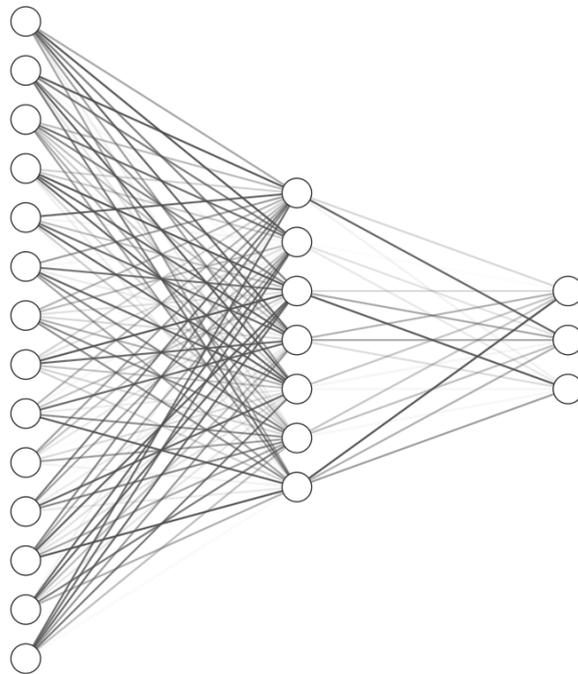

Gas leakage detection in hydraulic accumulator

- A neural network approach -



Master Thesis
OES10-11-E19

Aalborg University
Energy

Copyright © Aalborg University 2015

This Master thesis was typeset with online LaTeX editor Overleaf. Front page image of neural network architecture was created with online tool NN-SVG. Block diagrams were drawn with online tool draw.io. Simulations and graphs were created using MATLAB and Simulink from MathWorks®.



AALBORG UNIVERSITY
DENMARK

Energy
Aalborg University
<http://www.aau.dk>

Title:

Gas leakage detection in hydraulic accumulator - A neural network approach -

Theme:

Scientific Theme

Project Period:

Spring Semester 2019

Project Group:

11

Participant(s):

Alessandro Racca

Supervisor(s):

Jesper Liniger

Copies: 1

Page Numbers: 42

Date of Completion:

May 31, 2019

Abstract:

This Master thesis describes a model-based fault detection approach using a Nonlinear AutoRegressive neural network with exogenous input (NARX). The fault of interest is a gas leakage in hydraulic accumulator. Accumulator considered is part of the hydraulic circuit that controls pitch angle in a wind turbine. In order to generate training data for the supervised learning of the network, a model of the accumulator is considered. NARX network is trained in open loop configuration with simulated load flow rate as input and supply pressure as output. NARX model of a healthy system is then fed with faulty input-output data as part of model-based FDD. Open loop network is proven to be too accurate to be used for fault detection, predicting the actual output instead of the healthy output. It is concluded that a closed loop NARX is preferred for fault detection.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Contents

Preface	vii
1 Introduction	1
1.1 Wind turbine pitch system overview	1
1.2 Fault detection and diagnosis	2
1.2.1 Terminology	4
1.3 Reliability of pitch system	4
1.4 FDD theory	5
1.4.1 Fault detection methods	5
1.4.2 Fault diagnosis	7
1.5 Preliminary research question	7
2 Fault analysis	9
2.1 Accumulator model	9
2.1.1 Governing equations	10
2.1.2 Gas mass estimation	13
2.2 Process simulation	13
2.2.1 Load flow rate	13
2.2.2 Simulated process	14
2.3 Simulation results analysis	15
2.3.1 Supply pressure	16
2.4 Fault analysis with frequency response	16
3 Literature review	17
3.1 FDD on hydraulic accumulator	17
3.2 Research question update	18
3.3 FDD with artificial neural networks	18
3.4 Research question after literature review	20
4 Artificial neural networks	21
4.1 ANN, machine learning and deep learning	21
4.2 Perceptron neuron model	22

4.3	Universal approximation theorem	23
4.4	ANN parameters and network learning concept	23
4.5	ANN for dynamic systems	25
4.6	NARX network	25
4.7	NARX training and use for prediction	26
4.8	Governing equations of closed loop NARX	27
4.9	Training method and training algorithm	28
4.10	Chapter conclusion and ANN limitations	29
5	FDD with NARX network on simulated data from accumulator model	31
5.1	Residual generation using open loop NARX	31
5.2	Training $NARX_{ol}^{180}$	32
5.2.1	Choice of input and output variables scaling	32
5.2.2	Training data division	33
5.2.3	Number of hidden neurons and time delays	33
5.2.4	Open loop network	34
5.3	Training results of closed loop network $NARX_{cl}^{180}$	35
5.4	FDD results of open loop network $NARX_{ol}^{180}$	37
6	Conclusion	39
	Bibliography	41

Preface

This Master thesis is the conclusion of the Master Programme in Sustainable Energy Engineering - Offshore Energy Systems. The Master Programme is offered in Esbjerg by Aalborg University.

I would like to sincerely thank my supervisor Jesper Liniger for his supervision and support throughout the thesis work.

Aalborg University, May 31, 2019

Alessandro Racca
<aracca17@student.aau.dk>

Chapter 1

Introduction

The objective of this Master thesis in Sustainable Energy Engineering is to detect gas leakage in an hydraulic accumulator. Hydraulic accumulators are used in different fields such as automotive (for braking energy recovery), marine engineering (active heave compensation in offshore winch systems) and wind turbines pitch systems just to mention a few.

As batteries store electrical energy in form of voltage for instance, hydraulic accumulators are energy-storage devices within a hydraulic circuit in which a liquid is held at high pressure by a spring, weight, or high-pressure gas [1]. Different types of hydraulic accumulators are used, but the ones with pressurised gas usually perform better and are more compact [2]. Pressurised-gas-filled accumulators are bladder-type or piston type depending on how the fluid is separated from the gas. This gas is subject to leakage and such undesired event causes the accumulator to loose its capacity to store energy. For a proper functioning of the hydraulic system accumulator is part of, the mass of pressurised gas should ideally remain equal to design value (or at least never drop below a certain threshold). In Chapter 2, a more detailed explanation of the accumulator functioning will be presented.

A fault detection and diagnosis (FDD) scheme able to detect and diagnose the level of gas is therefore useful since many hydraulics systems have to be stopped for gas refill as part of their maintenance. An accurate and robust scheme could reduce this downtime, resulting in safety and economical benefits.

1.1 Wind turbine pitch system overview

One of the applications of hydraulic accumulator is within wind turbines (WT) pitch system. WT convert the energy of the into electrical energy by taking advantage of the lift force acting on the blades. Taking an aerofoil immersed in a flow as a simple example of blade section, it is intuitive to understand that lift force (and therefore the power extracted from the wind) can be modified by rotating the aero-

foil of a certain angle. The angle between the tip chord of the aerofoil and the rotor plane is called pitch angle [3]. The pitch system is the a WT subsystem dedicated to adjust the pitch angle. Pitch angle typically varies from -2° to 30° at a maximum rate of $\pm 10^\circ/\text{s}$ [4]. The pitch system has two main functions. During normal operations, it contributes in assuring a constant power output from the WT. The system is also involved in emergency situations, when a speed limitation is needed. The pitch can be adjusted so that the blade are in stall condition (aerodynamic brake). Therefore, pitch system is involved both in operation and safety.

Considering a hydraulic pitch system, many components are involved. As already introduced, it is not in the scope of this thesis to analyse multiple faults that could occur in the whole system. Therefore a description of the hydraulic system is briefly provided here based on [5].

Within the hydraulic system, a fluid (typically oil) flows from a supply circuit to two other circuits: the actuation/safety circuit and the locking circuit. Supply is located in the nacelle and is therefore is not rotating. Actuation/safety and locking are sitting in the hub and rotating with it. Rotating connectors assure flow rates are delivered to the rotating parts.

Actuation and safety part is composed by actuators, valves and safety devices (accumulators). Actuators are double acting differential cylinders. The position of the piston within the cylinder is related to the pitch angle. The piston is designed so that a fully extended piston correspond to a stalled blade. This is remarkable because in case of emergency shutdown, when all the valves are de-energised, the pressurised fluid stored in the accumulators flows to the actuators and brings pistons to fully extended position, providing aerodynamic brake. As long as the fluid in the accumulator is pressurised, emergency shutdown can be performed even without the supply pump.

Lastly, the locking part of the hydraulic system intervenes in case of normal shutdown. Normal shutdown happens when turbine is stopped on purpose (wind speed too low or too high or scheduled maintenance). The pitch cylinders are brought to fully extended positions and locking cylinder is engaged, stopping the blade in a fixed pitch angle.

The three circuits can be seen in Figure 1.1. The pitch system is considered to fail if the intended function in any of the modes is not achieved. In this thesis, the emergency shutdown is the operational mode of interest since a failure in the accumulator is critical in this mode. In the following section an overview of fault detection and diagnosis is presented.

1.2 Fault detection and diagnosis

As described in [6], since 1960 process automation was a growing trend in industry. In parallel, there was a sensible progress in the areas of sensors, actuators,

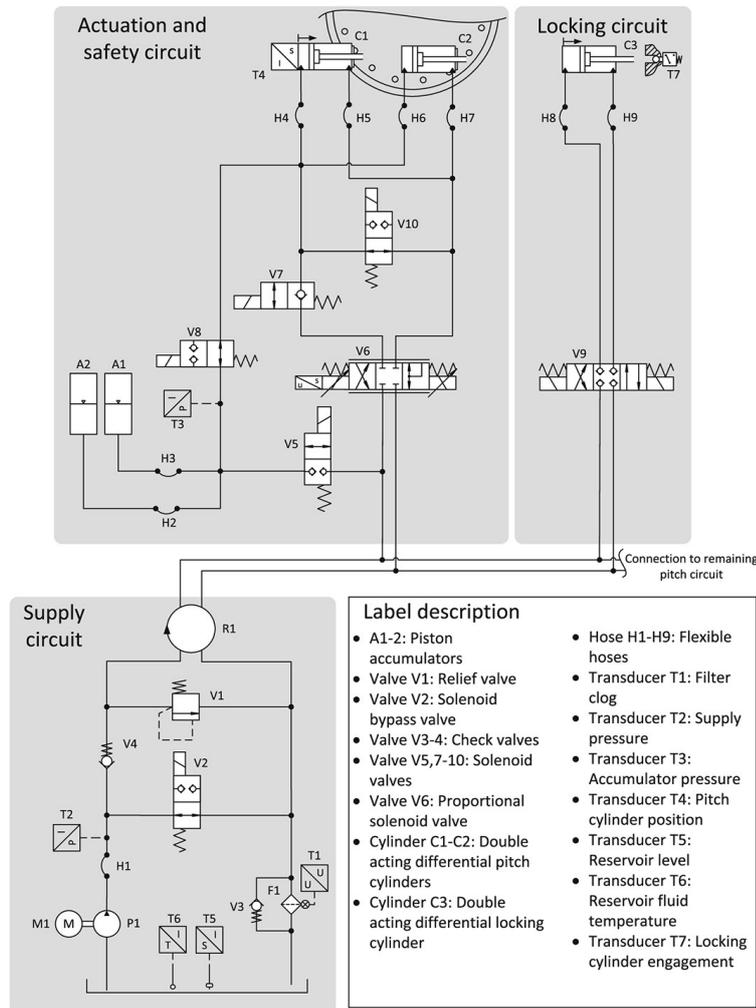


Figure 1.1: Generalised diagram of a fluid power pitch system [5]

bus-communication systems, and human-machine inter-faces. The more control systems improved, the more *supervision* of technical processes was needed and evolved. Supervision is aimed at showing the present state of a process, indicating undesired or unpermitted states, and taking appropriate actions to avoid damage or accidents. When the behaviour of a process deviates from normal, it could be the result of a *fault*. In this thesis, definitions from [6] will be used.

1.2.1 Terminology

Fault

A fault is an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual, standard condition.

It is important to remark that a fault is a state within the system that may cause a reduction in or loss of the capability of a functional unit to perform a function. A fault may initiate a failure or a malfunctioning.

Failure

A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions.

A failure is therefore an event that can result from one or more different faults.

Failure rate

The failure rate λ is defined as the instantaneous rate of failing elements dn_f/dt related to still functioning elements $n(t)$.

$$\lambda(t) = \frac{1}{n(t)} \frac{dn_f(t)}{dt} \quad (1.1)$$

Failure rate can also be considered as:

$$\lambda(t) = \frac{1}{\text{number of functioning elements}} \frac{\text{number of failures}}{\text{time period}} \quad (1.2)$$

The failure rate plays a role in the estimation of reliability, where reliability is the ability of a component, process or a system to perform a required function correctly under stated condition within a given scope, during a given period of time.

In the following section some of the concepts presented here will be used to present the reliability of the pitch system and the accumulator.

1.3 Reliability of pitch system

Several studies investigating reliability of wind turbines. As found in [5], according to some studies, the pitch system is the most unreliable sub-assembly of the entire wind turbine, contributing over 20% to both the overall turbine failure rate and downtime. Figure 1.2 shows the failure rate distribution for fluid power pitch system. It can be seen that accumulators failures contribute as approximately 10% of the total failure rate. Even though downtime related to accumulator failure mode is not specified in [5], it is clear that accumulators contribute to downtime as well.

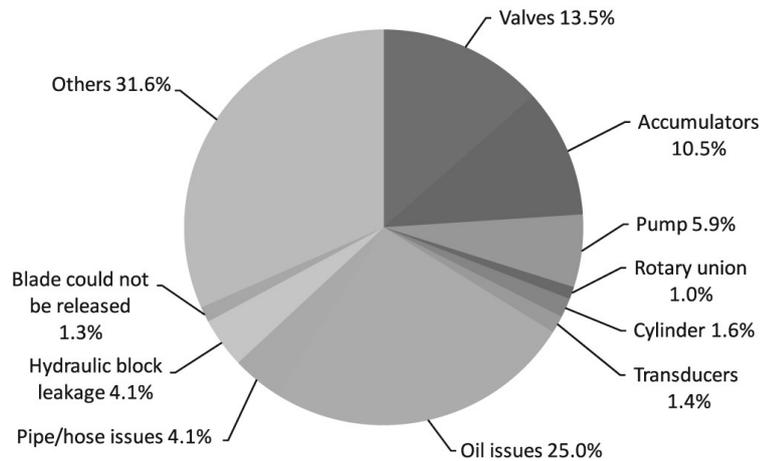


Figure 1.2: Failure rate in percentage as found in [5]

Accumulator contribution to total failure rate is therefore motivating this work. With an efficient and reliable FDD scheme to estimate gas leakage in the accumulator (and therefore the level of gas), failure rate and downtime would probably be reduced. Moreover, being able to predict the mass of gas remained in the accumulator, would help planned maintenance operations, positively impacting on downtime.

1.4 FDD theory

This section is aimed at providing the reader a theoretical background on FDD on top of the terminology. As already mentioned, detection and diagnosis are defined differently and involve different methods. The following sections are based on [6], a comprehensive book on the topic that will be recurrently used throughout this thesis.

1.4.1 Fault detection methods

Within fault detection, a major division can be considered: fault detection with signal models and fault detection with process models. In this thesis, only fault detection with process models will be considered.

Fault detection with process models

As can be seen in Figure ??, fault detection with signal models process the output signal y . If more measurements can be extracted from the system, such as input u , fault detection with process models can be performed. Input and output signals

can be fed to a process model. Process models simply describe the relationship between input u and output y . The estimated output from the model is then used for feature generation. Features such as difference between model estimated output and process output y are then compared with features from normal behaviour and symptoms are generated.

It is clear that this fault detection method is model-based and therefore relies on the accuracy of the model. A model has to be enough precise in order to express deviations as result of process fault [6]. Thus, in [6], it is suggested to apply process-identification methods before applying any model-based fault detection method.

Identification of a non-linear dynamic process can be performed by estimating model parameters (parameter estimation). One of the techniques for parameter estimation involves artificial neural networks (ANN). ANN are used for single-input-single-output (SISO) processes but can be extended to multiple-input-multiple-output (MIMO) processes. They are general model structures allowing small to medium disturbances and can reach a very large accuracy. Therefore ANN are used for design of non-linear controllers, learning controllers and fault detection. ANN are mentioned here because nowadays their applications are widespread (from modeling to image recognition for instance) and it is important to frame ANN in a fault detection context. A more detailed analysis of ANN is found in Chapter 4.

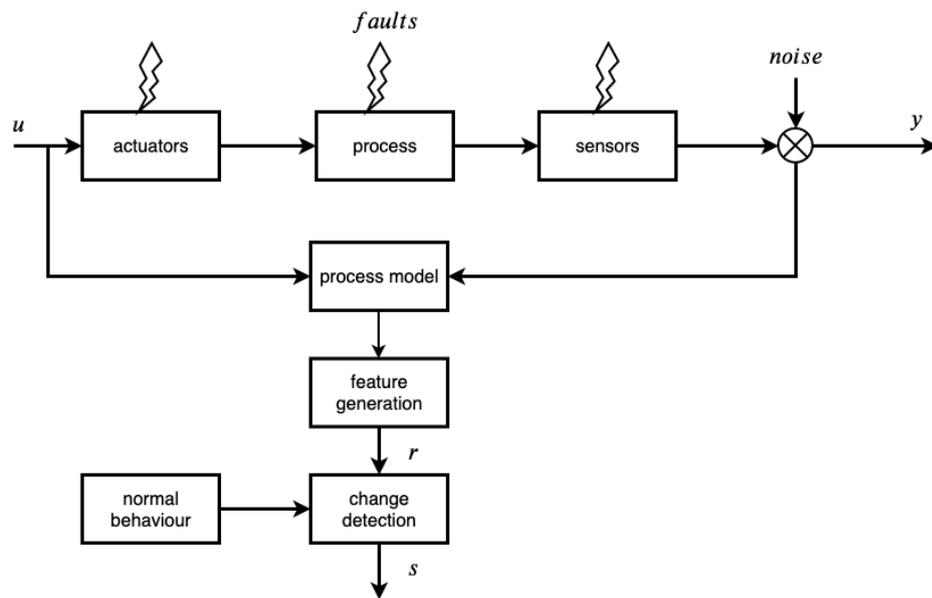


Figure 1.3: Scheme for fault detection with process model inspired by [6]

Some other process models often used while performing fault detection are

listed below.

- parameter estimation
- state observers (Kalman filters, extended Kalman filters, ...)
- parity equations

To conclude the section and sum up, Figure 1.3 shows the scheme from fault detection with process models.

1.4.2 Fault diagnosis

As described in [6], fault diagnosis consist of determination of fault type and fault details, like size, location and time of detection. Fault diagnosis comes after fault detection since it is based on the symptoms of the fault detection methods. It is important to remind here that a symptom s is an unusual change of a feature from its normal value. In a fault free situation symptoms are zero. In a physical system, propagation of faults to intermediate steps (events) and then to observable symptoms, often follows physical cause-effect relationships. To track these relationships analytically is hard because physical laws may be unknown or too complicated. Fault diagnosis proceeds the opposite way, from symptoms observation to fault diagnosis. This procedure implies the inversion of causality. Intuitively, this is an issue, since causality is not reversible or the reversibility is ambiguous [6]. Moreover, intermediate events between symptoms and fault are not always visible. Following the descriptive approach found in [7], fault diagnosis methods can be divided in two groups:

- *classification methods*, experimentally trained methods that can be when no information on fault-symptoms causalities is available
- *reasoning or inference methods*, are applicable when fault-symptoms causalities can be expressed as if-then-rules.

1.5 Preliminary research question

So far the introduction was aimed at giving the reader an overview of the problem area and some basic knowledge of the theoretical background (FDD). A preliminary research question is formulated as following:

How can a gas leakage be detected in hydraulic accumulator during its operation in a wind turbine pitch system?

The research question will be scoped down during the thesis, taking advantage of the theory on FDD and on related prior work.

Chapter 2

Fault analysis

This chapter is dedicated to the fault analysis. In the introduction the fault of interest was mentioned and will be described more in detail in this chapter while presenting accumulator model.

The purpose of fault analysis is to investigate whether the fault results in abnormalities of system states. The concept is to define a system (i.e. define the process inputs and outputs), introduce a fault in the it and analyse the faulty system behaviour to understand the impact of the fault.

In fact, it is not always possible to introduce faults in a physical process. Therefore simulation is often involved in fault analysis. In order to simulate the system response, generating output data, a model has to be developed.

2.1 Accumulator model

Piston type pressurised-gas-filled accumulators are considered in this thesis. Gas chamber is usually filled with nitrogen (N_2) through a gas valve. It is introduced at a pre-charge pressure that is determined by the pressure range required by the application [2]. Fluid chamber is filled with oil and has one port through which a load flow can enter the accumulator. A simplified representation of an accumulator is shown in Figure 2.1.

The incoming flow rate is denominated q_{pump} because in the hydraulic circuit fluid is flowing from the supply side. The out-coming flow rate is called q_{load} because this flow rate flows to the pitch actuators and depends on the pitch angle activity. Even though not directly measurable, flow rate to accumulator q_{accu} is also considered.

Gas and fluid are separated by a piston. Since piston is sealed, friction between piston and accumulator walls can affect the pressure levels. Moreover, the piston seal is one of the leakage paths for the gas. As can be found in [8], gas and fluid leakages are typical failure modes. Gas can leak directly out of the accumulator

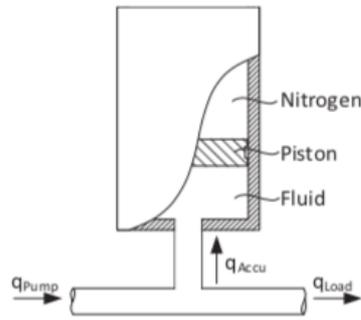


Figure 2.1: Accumulator

or internally leak into fluid. In this work, gas leakage is analysed regardless the leakage path and it is the only fault considered.

As already mentioned in the introduction, gas leakage reduces accumulator ability to store energy. With not enough nitrogen on the gas side, in case of critical failure, fluid in the accumulator has not enough pressure to bring pitch cylinders to fully extended position (aerodynamic brake).

Following the same approach as [8], gas leakage is regarded as a slow process. This assumption is based on the fact that one of the main leakage paths is through the piston annulus and then within the fluid. For convenience, the pre-charge pressure p_{pc} instead of the mass of gas m_g is used to indicate how much gas is remaining in the accumulator. Pre-charge pressure is then the *measure* associated to the gas leakage fault.

Pre-charge pressure is defined as the pressure of the gas introduced in the accumulator when no fluid is present (i.e. gas volume is equal to the total volume of the accumulator). It is important to remark that pre-charge pressure depends on gas temperature. As found in [8], in WT applications pre-charge pressure ranges between 180 and 50 bar. The same range is used in this work.

2.1.1 Governing equations

In this section, accumulator model is presented. This model will be used to generate input-output data to analyse the fault. Model is based on prior work from [8] and [9]. It is assumed that fluid is not compressible (i.e. constant density), and piston mass and friction between piston and fluid and piston and accumulator walls are neglected. This leads to the assumption that pressures on both sides of the piston are equal. Fluid pressure is denoted as supply pressure p_s .

Three governing equations are considered. Continuity equation describes the relationship between the variation over time of gas volume, pump flow rate and load flow rate. This equation is derived from the conservation of mass for in-

compressible fluids.

$$\dot{V}_g = -q_{accu} = q_{load} - q_{pump} \quad (2.1)$$

When pump flow rate is greater than load flow rate, gas volume decreases over time. \dot{V}_g is used to indicate time derivative of gas volume. If flow rates are expressed in SI units, \dot{V}_g is then $\text{m}^3 \text{s}^{-1}$.

The gas behaviour is modelled using the Benedict-Well-Rubin (BWR) equation of state. BWR was proven by several studies to be more accurate than the ideal gas law [8]. The gas is not be considered ideal in this application. BWR express relationship between gas pressure (i.e. supply pressure), gas volume and gas temperature T_g for a gas of mass m_g .

$$p_s = \frac{RT_g}{\frac{V_g}{m_g}} + \frac{B_0 RT_g - A_0 - \frac{C_0}{T_g^2}}{\left(\frac{V_g}{m_g}\right)^2} + \frac{b RT_g - a}{\left(\frac{V_g}{m_g}\right)^3} + \frac{a\alpha}{\left(\frac{V_g}{m_g}\right)^6} + \frac{c \left(1 + \frac{\gamma}{\left(\frac{V_g}{m_g}\right)^2}\right) e^{-\frac{\gamma}{\left(\frac{V_g}{m_g}\right)^2}}}{\left(\frac{V_g}{m_g}\right)^3 T_g^2} \quad (2.2)$$

Constants for nitrogen are presented in Table 2.1 and are found in [9].

Constant	Value	Unit
A_0	0.1068	$(\text{m}^3/\text{mol})^2$
B_0	$4.07 \cdot 10^{-5}$	m^3/mol
C_0	816.58	$(\text{m}^3/\text{mol})^2 \text{K}^2 \text{Pa}$
a	$2.54 \cdot 10^{-6}$	$(\text{m}^3/\text{mol})^3 \text{Pa}$
b	$2.3276 \cdot 10^{-9}$	$(\text{m}^3/\text{mol})^2$
c	0.07381	$(\text{m}^3/\text{mol})^3 \text{K}^2 \text{Pa}$
γ	$5.3 \cdot 10^{-9}$	$(\text{m}^3/\text{mol})^2$
α	$1.2720 \cdot 10^{-13}$	$(\text{m}^3/\text{mol})^3$
R	8.134	$\text{m}^3/\text{mol Pa/K}$

Table 2.1: BWR constants

These constant are not in SI units, since unit mol is used instead of kg for mass. Therefore, m_g has to be expressed in mol when used in BWR in order to obtain p_s in Pa. Molar mass MM_{N_2} for nitrogen is used to obtain nitrogen mass in grams.

$$M_g = m_g \cdot MM_{N_2} = m_g \cdot 28\text{g/mol} \quad (2.3)$$

The last governing equation is a energy equation. The mass of gas within the gas chamber can be defined as a closed system. As known from thermodynamics, a closed system during compression, changes its volume and temperature in general. From first law of thermodynamics, assuming a quasi-static compression process, and considering gas pressure and temperature homogeneously distributed within

gas chamber, it is possible to describe the gas temperature change in time \dot{T}_g .

$$\dot{T}_g = \frac{T_a - T_g}{\tau} - \frac{T_g}{c_v} \frac{\partial p_s}{\partial T_g} \frac{\dot{V}_g}{m_g} \quad (2.4)$$

Time derivative of gas temperature therefore depends on gas temperature, ambient temperature T_a , thermal time constant τ , specific heat capacity at constant volume c_v , time derivative of gas volume, gas mass and partial derivative of supply pressure over gas temperature. This partial derivative can be obtained analytically from BWR, thanks to the fact that BWR includes a set of constants and V_g and T_g are not function of p_s .

Time derivative of gas temperature is related to the thermal power exchanged between the system (i.e. gas mass) and the ambient. As can be seen from first term of previous equation, the bigger temperature difference, the bigger the heat exchange in general. Heat exchange is a very complex topic and some approximations on thermal time constant τ have to be made in order to have a realistic approximation of heat exchange dynamics. In [8], approximation from Rotthäuser [10] is used.

$$\tau \approx 0.3 \cdot 10^{-5} \cdot p_{pc} V_a^{0.33} + 86.2 \cdot V_a^{0.49} \quad (2.5)$$

Thermal time constant can be computed knowing accumulator volume V_a and pre-charge pressure. Considering a pre-charge pressure of 140 bar and a 25 L accumulator, a thermal time constant of 32.8 s is obtained.

In this thesis though, a thermal time constant of 60.8 s is used thanks to the work performed in [9]. Thermal time constant was estimated experimentally using a 25 L accumulator. Accumulator was charged with a constant supply pressure. This pressure leads piston to move, compressing and heating the gas. When piston stops moving, gas is warmer than ambient. Piston position is recorded, supply pressure is held constant and gas is left cooling down. As gas cools down under constant pressure, its volume reduces and causes piston to move. The isobaric process stops when the gas temperature and the ambient temperature are equal. Final piston position is then recorded. By measuring the time between two steady state piston positions, thermal time constant is obtained. In [9], experiments are performed with different pre-charge pressures and with different supply pressures. It is found that there is a negative correlation between supply pressure and thermal time constant for different pre-charge pressures and a positive correlation between pre-charge pressure and thermal time constant for different supply pressures. It is concluded to use the average of constants obtained in experiments with supply pressure ranging from 180 to 200 bar with a pre-charge pressure of 140 bar. The same value (60.8) is used in this thesis. The thermal time constant changes with the pre-charge pressure, but it will be considered constant in this work.

2.1.2 Gas mass estimation

BWR can be used to estimate the mass of gas for a given pre-charge pressure and pre-charge temperature T_{pc} . BWR is solved numerically. With 140 bar as pre-charge pressure and 22 °C as pre-charge temperature for instance, a mass of 3.93 kg is obtained.

2.2 Process simulation

The continuity equation, the BWR equation, the energy equation and the thermal time constant approximation are used to simulate the process. The constants used in the simulation that were not mentioned before are summarised in Table 2.2.

Constant	Value	Unit
c_v	20.8040	(J/molK)
T_a	22	°C
V_a	25	L
T_{pc}	22	°C
τ	60.8	s

Table 2.2: Other constants used in simulation

As found in [8], a realistic range for pre-charge pressure is 180-50 bar when pre-charge temperature is 22 °C. This same range will be used in the simulation.

2.2.1 Load flow rate

The load flow rate used in the simulation was provided from the work performed in [8]. It was obtained considering fluctuations of wind speed around rated wind speed. At rated wind speed, the WT ideally extracts from the wind rated power. It is then in the control objectives to operate as close as possible to the rated power. This can be done by pitching the blades. Fluctuations around rated wind speed then correspond to changes in the pitch angle and eventually in load flow rate activity. The load flow rate data were provided as 500 s experiments with a sampling frequency of 200 Hz, resulting in a considerable amount of measurements. It is then decided to down sample load flow rate. This operation has to be done with a down sampling factor that does not ignore the relevant dynamics of the process. Power spectrum for a load flow rate signal can be seen at the bottom of Figure 2.2. By looking at the power spectrum, it can be seen that the energy of the signal is mainly concentrated below 10 Hz. A down sampling factor of 20, corresponding to a new sampling frequency of 10 Hz is applied.

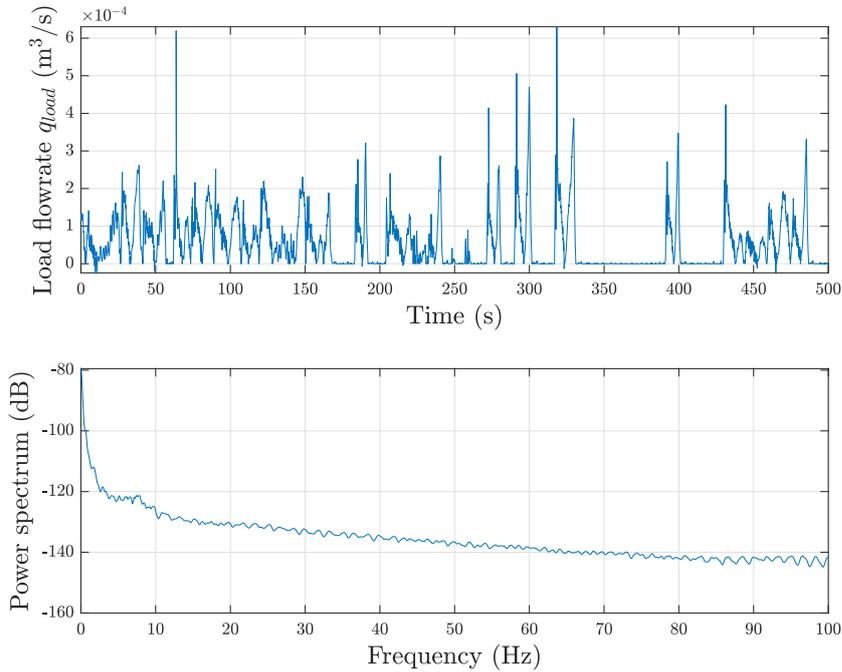


Figure 2.2: Load flow rate signal and power spectrum

2.2.2 Simulated process

In this section, the process governed by presented equations is simulated in three different situations:

- high pre-charge pressure $p_{pc} = 180$ bar
- medium pre-charge pressure $p_{pc} = 100$ bar
- low pre-charge pressure $p_{pc} = 50$ bar

The main objective of these simulation is to have an overview of the process behaviour and to observe the effect of a gas leakage fault considering three different values of fault measurement p_{pc} .

For the same load flow rate, supply pressure p_s , pump flow rate q_{pump} , accumulator flow rate q_{accu} , gas temperature T_g and ambient temperature T_a are plot The results of simulation for high, medium and low pre-charge pressure are shown in Figure 2.3. Note that from original experiments of 500 s, a 100 s transient time is neglected because it is related to the initial condition of the integrators of the model.

Within this simulation, a fixed displacement pump is used, providing a pump flow rate of 20 L/min ($3.3 \cdot 10^{-4} \text{m}^3/\text{s}$). The pump is active until supply pressure

reaches 200 bar. Then pump is kept off until supply pressure drops below 170 bar. This pump activity is comparable to how WT pumps are operated.

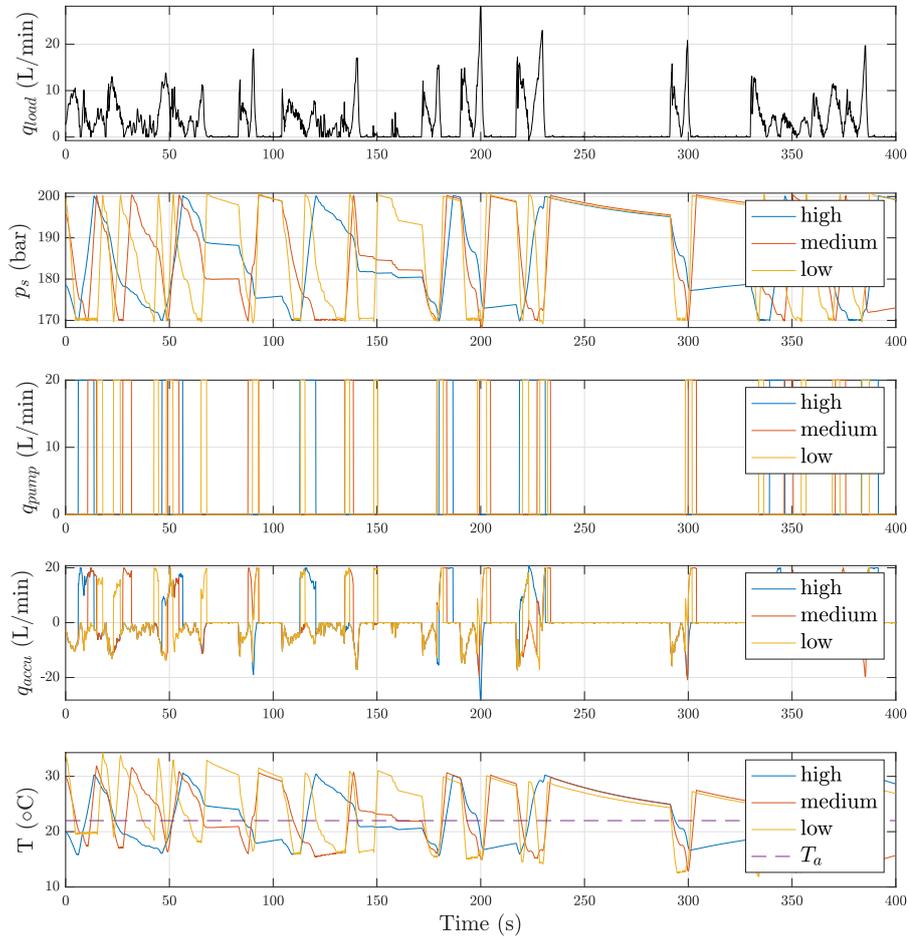


Figure 2.3: Simulated process with three levels of pre-charge pressure

2.3 Simulation results analysis

In this section simulation results are compared and briefly commented.

2.3.1 Supply pressure

By looking at simulated supply pressure for different pre-charge pressures, it is clear that with high p_{pc} , the supply pressures cycles between 170 and 200 bar more frequently than with lower pre-charge pressures (considering same length simulations and same load flow rate).

This is because of several reasons. A higher p_{pc} correspond to a higher gas mass on the gas side of the piston and compressing a higher mass requires more energy. Another reason is that the pump cycles more frequently when a fault has occurred. Pump flow rate frequency might seem a good fault indicator, but this frequency could be also due to load flow demands.

The supply pressure signal from simulation seems to contain some indication of the gas leakage. It is important to remark that signal is obtained from a model. Model uncertainties, process noise, sensor noise and other disturbances could lead to a much different real supply pressure signal, where fault indication is not clear.

2.4 Fault analysis with frequency response

In this section the fault analysis used in [8] is summarised. Using the governing equations presented before in this chapter, a linearised model of the accumulator is obtained. Linearised model is given as a transfer function in Laplace domain

$$\Delta P_s(s) / \Delta Q(s)$$

where $Q = q_{pump} - q_{load} = q_{accu}$. Frequency response of transfer function is plot for high, medium and low pre-charge pressure. It is observed that for frequencies between 1 mHz and 1 Hz, lower pre-charge pressures correspond to higher gain. The variations in the accumulator flow rate are amplified in the supply pressure when fault occurs. This fault analysis suggests that supply pressure signal is worth to be used for FDD.

Chapter 3

Literature review

In this chapter, the research done throughout the thesis is described. The literature review is split into two parts by an update of the research question.

3.1 FDD on hydraulic accumulator

Regarding prior work, this section will focus exclusively on FDD research on hydraulic accumulators. In [8], an accumulator model is developed and simulated supply pressure is used in a Multiresolution Signal Decomposition (MSD) scheme. MSD method considers oscillatory time-limited waveforms (wavelets). Wavelets allow simultaneous time and frequency domain analysis. Signal is decomposed and details coefficients are extracted. Each detail coefficient describes the original signal in a certain frequency range. Detail coefficient root mean square (RMS) are then analysed for different pre-charge pressures and RMS are used as comparative measure for detecting a fault. It is concluded that for nominal operating conditions, RMS of detail coefficient of Level 10 is the most sensitive to gas leakage fault. The robustness of this fault detection method is then verified by computing the RMS of the mentioned detail coefficient for operating condition different from nominal, changing mean wind speed and turbulence intensity class, ambient temperature and external fluid leakage levels. It is concluded that this method is not robust for all the variations and that ambient temperature has the biggest influence on the RMS. By considering a reduced ambient temperature range, RMS of detail coefficient of level 9 is selected for fault detection. This method showed best results during nominal operating conditions of a WT in ambient temperature range 22-60 °C, but it was also remarked that robustness is degraded by lower ambient temperature.

The fault detection method presented in [8] is a signal-based method and, apart from simulation purposes, a model of the process is not required. The method can be successfully applied directly on measured data. In [9] a model-based fault de-

tection method is applied to detect a gas leakage in a hydraulic accumulator. Based on the same three equation presented in Chapter 2, a nonlinear state space model is obtained. The model is then used to create an Extended Kalman filter (EKF). The purpose of the filter is to produce optimal state estimation, taking process and measurement noise into account. Kalman filters and extended Kalman filters work as state observers. If a Kalman filter modelled on a fault-free system is fed with input-output data from a faulty process, the process output and the estimated output should theoretically differ. Residuals (difference between process output and estimated output) should then be non zero. In [9], this approach is used. A bank of EKF modelled with different pre-charge pressures is designed. The EKF with residuals closest to zero is considered to indicate the level of pre-pressure. Maximum Likelihood Estimation (MLE) is used to analyse the residuals. The bank of EKF showed good results both with simulation data and with experimental data. This method though still relies on the accuracy of the model and tuning the Kalman filter requires knowledge on process and measurement noise, which sometimes is hard to obtain. The fault detection problem can be turned in an identification problem. This leads to an update of the research question.

3.2 Research question update

Model-based FDD methods often require a certain level of knowledge of process structure. There are some methods however, such as artificial neural networks (ANN) that do not require this knowledge (or at least very little). Therefore it is possible to scope down the research question from its first formulation in Chapter 1.

How can artificial neural networks be used for gas leakage detection in a hydraulic accumulator for a wind turbine pitch system?

The following section will focus on research work involving FDD, hydraulics systems and ANN.

3.3 FDD with artificial neural networks

[11] is not specifically investigating a gas leakage on hydraulic accumulator, but hydraulic actuator cross-line leakage (fluid leaking from one chamber of the actuator to the other) and other faults. Fault diagnosis is thought in terms of parameter estimation. A state space model of the system is first created and an output vector is defined. Output vector generated by a healthy process is then fed to a neural network with a Multi-Layer Perceptron Network (MLP) architecture. MLP has 150 input neurons (output vector has three variables and 50 values per variable are sampled) and 3 output neurons, which are three model parameters (i.g. leakage

coefficient, supply pressure and friction coefficient). The network is trained with supervised training, meaning it is fed with output vectors as input and at the same time it is given the target values of the parameters as network output. Once network has been trained, it is validated on both simulation data and experimental data. It is important to remark that in [11], a multi-fault environment is considered and that robustness of the method is not achieved for all the faults. The results obtained however, are very interesting, and the method used is clearly stated and surely repeatable. It is observed that, considering article is 20 years old, the softwares used for creating, training, validating and using MLP an ANN in general have evolved a lot since then. MATLAB for example, supports perceptron networks *for historical interest*, while suggesting to use other networks such as patternnet to classify inputs into target classes. The reviewed article from [11], by using ANN for model parameter estimation, is essentially using ANN for classification.

In [12] a MLP is used to predict the future system states based on the current inputs and states. MLP is therefore not used to solve a static classification problem, but within a dynamic system for prediction. With sufficient training and a suitable architecture, ANN is used to represent a nonlinear dynamic process. Authors of [12] argue that ANN used for fault classification as in [11], could give incorrect fault information because they only take into account system output (i.e. output vector). If system input changes, issues with diagnosis may occur, especially with nonlinear systems. In order to solve this issue, both input and output of monitored system can be fed into a *processor*, like a neural network and residuals can be generated. In this article, neural network is essentially used as an alternative to traditional state estimators such as Leunberger observers or Kalman filters. Claimed advantages of neural networks are powerful nonlinear mapping properties, noise tolerance and self learning. Moreover, in the same article residuals are fed into another neural network, which has been previously trained as a classifier. The second ANN is therefore used as fault classifier, indicating whether there is fault or not in the process. This fault detection method is proved to work at least for that experimental setup, but admits it is impossible to train the networks when the fault size it is too small. In addition, such scheme should also be proven robust to noise.

The FDD scheme developed in this [7] is very similar to the one presented in [12]. Instead of using a MLP for system state prediction (note that [12] was published in 1994), a Nonlinear AutoRegressive network with eXogenous input (NARX network) is used. NARX is a more sophisticated architecture than MLP and it is claimed to be a good and robust to noise state estimator. Residuals are then inspected with the help of another network architecture, a Focused Time Delay Artificial Neural Network (FTDANN). The FTDANN is trained to map the residuals of the NARX to the corresponding fault (multi-fault environment). The FDD scheme developed is particularly interesting because MATLAB provides a

GUI dedicated to NARX. NARX creation, customisation, training, validating, testing, and deploying is optimised for the user, making this architecture particularly attractive because a lot of coding work is already performed by MATLAB. In this Master thesis, it is then decided to adopt an approach similar to the one in [7], at least to what concern NARX network. Therefore research question is further scoped down.

3.4 Research question after literature review

ANN are a wide and expanding subject. After research presented in literature review, it is decided to investigate more on ANN and to try to develop a model-based FDD scheme using NARX architecture, considering its advantages compared to simpler MLPs. The research question therefore becomes:

How can a NARX network be trained, validated and used to detect a gas leakage fault in a pitch system hydraulic accumulator?

The following Chapter will present the basics of ANN and then describe extensively NARX networks.

Chapter 4

Artificial neural networks

ANN are composed by mathematically formulated neurons. The neurons are interconnected to form a network that allows the description of relationships between input and output signals [6]. ANN can be considered as universal function approximators, mapping inputs u to outputs y in general. Within a neural network, there are adjustable parameters that are unknown beforehand. These parameters have to be tuned by processing u and y signals through the network. This is called *network training*. The way neurons are connected determines the network architecture and how the network *learns* the input output relationships. A very good beginners reading on neural networks can be found in [13]. Before moving to the model of a neuron, it is important to summarise some terminology.

4.1 ANN, machine learning and deep learning

Neural network, machine learning and deep learning are topics and techniques that are sometimes confused between each other. Before presenting artificial neural networks, it is important to remark the differences between those three areas. At the very beginning, networks were created to describe, model and mimic the behaviour of biological neurons of human brain. This explains the origin of their name. Born as mathematical structures, ANN then evolved to programming paradigms which enable computers to learn from observational data [13]. In a neural network algorithm, the computer learn by itself how to solve an assigned problem by using observational data.

Artificial neural networks became then some of the computational methods used within *machine learning*. Machine learning is defined a group of statistical methods that enable machines to "learn" tasks from data without explicit programming. Neural networks are used within machine learning to solve regression and clustering problems [14].

A computer that learns from experience sounded very promising, but only

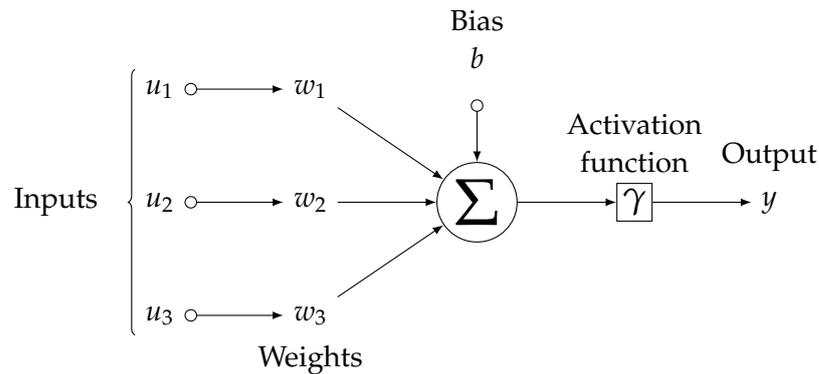


Figure 4.1: Perceptron with three inputs and one output

after 2006 new learning techniques were discovered, allowing ANN to solve more complex problems [13]. These more complicated networks were then redefined *deep neural networks*, in contrast with simpler architectures existing before (*shallow neural networks*). The ensemble of techniques and architectures of deep neural networks is also referred to as *deep learning*.

Machine learning and deep learning should not be confused. Even if they both involve neural networks and use them for modelling problems, there are two major differences [15]. In machine learning, the user has to select the features model is trained with. In deep learning, the algorithm automatically selects useful features. Moreover, deep neural networks include more layers between input and output layers compared to shallow neural networks used in machine learning, resulting in more complex architectures.

Regarding the NARX network mentioned in the research question before, it can be considered a shallow neural network as long as there is only one hidden layer.

In order to better understand network layers and why networks are used for modelling, it is useful to investigate the simplest network, a perceptron neuron model.

4.2 Perceptron neuron model

One of the fundamental blocks of a neural network is the perceptron. Perceptrons are basically created by adding multiple inputs (a vector of inputs for example) to simple single-input single-output neurons. An example of perceptron is shown in Figure 4.1.

Proceeding from left to right in Figure 4.1, the network transforms input vector \mathbf{u} into output y by using weights (also called gains) \mathbf{w} , a bias b and an activation function γ . Here, notation indicates that \mathbf{w} is a vector.

$$\mathbf{w} = [w_1 \quad w_2 \quad w_3] \quad (4.1)$$

The inputs are multiplied by the gains, all these products are summed together and then this sum is shifted of some bias. The biased weighted sum is then fed to an activation function that eventually returns the output. Activation function is sometimes referred as transfer function and common activation functions are sigmoid functions, step functions, hyperbolic tangent functions and linear functions. This input-output process can be summed up with the activation equation, expressing the neuron output as a function of weights and bias [16].

$$y = \gamma(\mathbf{w}\mathbf{u} + b) = \gamma(w_1u_1 + w_2u_2 + w_3u_3 + b) \quad (4.2)$$

The output of the perceptron is also called neuron activation. In this example, the output is a scalar because the perceptron is composed by only one neuron. If other *rows* were added below, each of them with its own weights, sum, bias and transfer function, the resulting architecture of multiple neurons would produce a vector output. This column of neurons is defined as a layer. A layer includes the weights, the multiplication and summing operations, the bias and the transfer function. The array of inputs, vector u , is not included in or called a layer. The first layer exposed to the input vector is called input layer, and the one whose activations are the final output is the output layer. Every layer in-between is defined as hidden layer. An example of a multi-layer neural network is shown in Figure 4.2. The network has R inputs and N outputs. Moreover, the arrows are all pointing in the output direction, meaning the activation information is passed only from left to right, from input to output. Such a network is a feedforward network. Other possibilities exist, like backward, lateral and recurrent.

It is remarked that activation of each neuron is a value between 0 and 1. Inputs and outputs of ANN need to be scaled within this range when network is used.

4.3 Universal approximation theorem

ANN were defined before as universal approximators. This is mathematically proven with the universal approximation theorem. A mathematical formulation of the theorem is found in [7]. Here, it is enough to say that a feedforward neural network with one hidden layer containing finite hidden neurons and with non-constant, bounded and monotonically-increasing continuous arbitrary activation function γ , is a universal approximator.

4.4 ANN parameters and network learning concept

The behaviour of a single neuron is defined by the numerical values of its weights and bias. Transfer function does not play a role because it is usually the same for every neuron in the net. Weights and bias are therefore the network parameters that can be adjusted in order to get the network to map input and output

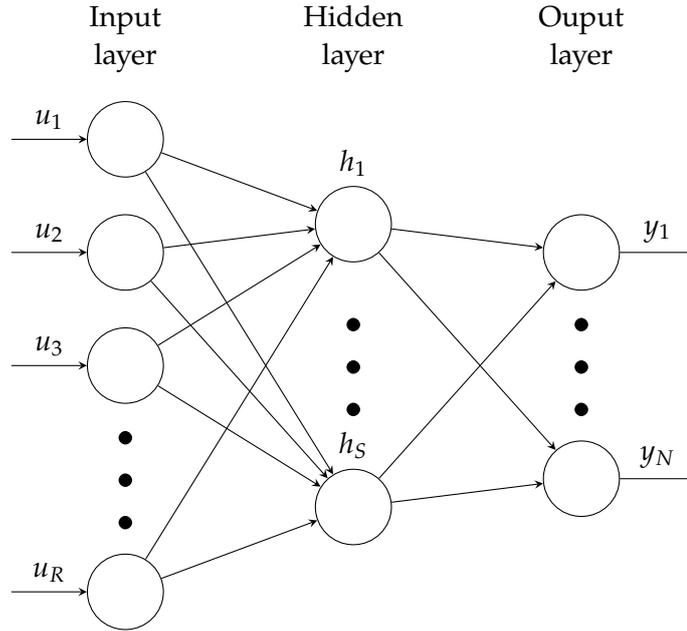


Figure 4.2: Multi-layer neural network

correctly. To learn for a network essentially means to adapt weights and biases so that the network solve the approximation problem. This adaptation is performed by simultaneously feeding the network with input and with target output. If only weights are considered, a quadratic loss function *SSE* (Sum of Squared Errors) can be defined as

$$SSE(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N-1} e^2(n) \quad (4.3)$$

where

$$e(n) = y(n) - \hat{y}(n) \quad (4.4)$$

is the model error for the n -th element of the output vector. Model error is defined as the difference between the target output (also called measured output) y , and the network estimated output \hat{y} .

The correct values for the weights are the ones that minimise the quadratic loss function. They are found by solving

$$\frac{dSSE(\mathbf{w})}{d\mathbf{w}} = 0 \quad (4.5)$$

It has to be noted that for a network as in Figure 4.2 with R elements in the input

vector and S hidden neurons, the weight vector \mathbf{w} becomes the matrix W .

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \dots & \dots & \dots & \dots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix} \quad (4.6)$$

Gradient methods for numerical optimisation are often applied. A very famous method is the back-propagation method.

In this section, a multi-layer ANN with one hidden layer was presented because of its simplicity. Some basic equations were explained and the concept of training described. More complicated architectures can solve more complicated problems, but involve more parameters. More parameters need to be adjusted and back-propagation error algorithm might result in long training time.

4.5 ANN for dynamic systems

Multi-layer perceptron networks discussed so far allow identification of nonlinear static systems. Input and output vectors are not functions of time. As shown in literature review, MLP networks can be used in fault detection to solve fault classification problems (i.e. parameter estimation). These static networks are essentially memoryless [6]. ANN for static systems can be extended with dynamic elements. In other words, the input vector is a discrete time input signal $u(k)$ that is first passed through a filter (a delay filter in the simplest case) and then to the network. Output vector $y(k)$ can also be delayed and used as input to the network. In a very general form, a dynamic network is a function f that computes output $\hat{y}(k)$ based on the current and previous inputs and outputs.

$$\hat{y}(k) = f_{NN}[u(k-1), \dots, u(k-d_u), \hat{y}(k-1), \hat{y}(k-2), \dots, \hat{y}(k-d_y)] \quad (4.7)$$

where d_u and d_y are time delays on input and output respectively. Note that such equation allows the dynamic neural network to predict one step ahead. In the following section, a kind of dynamic network (NARX) is presented more in detail.

4.6 NARX network

The Nonlinear AutoRegressive network with exogenous inputs (NARX) is a recurrent dynamic network. The NARX model is based on the linear ARX model, which is commonly used in time-series modeling [16]. NARX is defined by the same general equation already presented for dynamic neural networks. It is important to remark that two configurations for NARX exist:

- Parallel architecture or closed loop

- Series-Parallel architecture or open loop

The two configurations are shown in Figures 4.3 and 4.4 respectively. In a parallel NARX, the network is fed with delayed inputs and estimated outputs (TDL in Figures 4.3 and 4.4 stands for Tapped Delay Line). In a series-parallel NARX, the network is fed with delayed inputs and process (measured) output. The defining equation for a parallel NARX is then

$$\hat{y}(k) = f_p[u(k-1), \dots, u(k-d_u), \hat{y}(k-1), \hat{y}(k-2), \dots, \hat{y}(k-d_y)] \quad (4.8)$$

while for a series-parallel network,

$$\hat{y}(k) = f_{s-p}[u(k-1), \dots, u(k-d_u), y(k-1), y(k-2), \dots, y(k-d_y)] \quad (4.9)$$

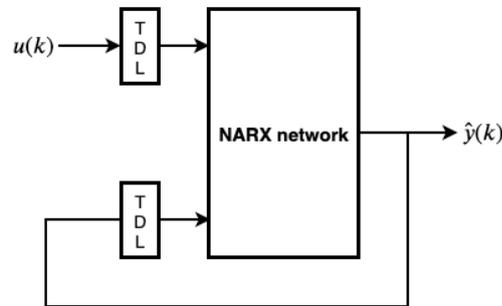


Figure 4.3: Parallel architecture

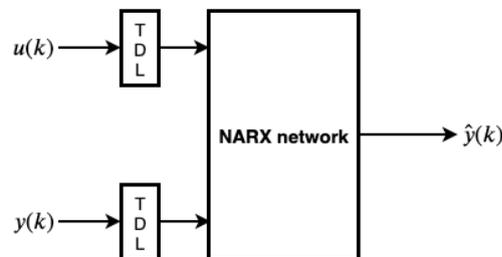


Figure 4.4: Series-Parallel architecture

4.7 NARX training and use for prediction

When using the open loop configuration, the measured output is available to the network and therefore this technique is preferred for network training. According to MathWorks [16], training in open loop has two advantages. First, the input to the network is more accurate. Second, the open loop neural network has a purely feedforward architecture and is therefore possible to apply static back-propagation

for training. MathWorks suggestion is to create a network in open loop, train, validate and test its performance. Then the network should be transformed to closed loop for multi-step ahead prediction.

In Chapter 5, a NARX neural network is trained with simulated data from an accumulator model and work flow suggested by MathWorks is applied and analysed.

4.8 Governing equations of closed loop NARX

In this section, the governing equations of a closed loop NARX network are shown keeping the same notation that was used for the perceptron. The equations will be referred to Figure 4.5. It is decided to consider a finite number of input and output delays in order to obtain a less heavy notation for this example. 2 input delays and 2 output delays are considered. Note that circles below input layer are not neurons and are therefore coloured in grey instead of black. Hidden layer is composed of 5 neurons. They all include the same transfer function γ_1 , but each of them has its own weights and bias. First neuron from top of the hidden layer for example, has weights

$$\begin{bmatrix} w_{1,1}^{(1)} & w_{1,2}^{(1)} & w_{1,3}^{(1)} & w_{1,4}^{(1)} \end{bmatrix} \quad (4.10)$$

and bias $b_1^{(1)}$. Using this notation, weights and bias for all the hidden neuron can be grouped in matrix $W^{(1)}$ and vector $b^{(1)}$.

$$W^{(1)} = \begin{bmatrix} w_{1,1}^{(1)} & w_{1,2}^{(1)} & w_{1,3}^{(1)} & w_{1,4}^{(1)} \\ w_{2,1}^{(1)} & w_{2,2}^{(1)} & w_{2,3}^{(1)} & w_{2,4}^{(1)} \\ \dots & \dots & \dots & \dots \\ w_{5,1}^{(1)} & w_{5,2}^{(1)} & w_{5,3}^{(1)} & w_{5,4}^{(1)} \end{bmatrix} \quad (4.11)$$

$$b^{(1)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ b_3^{(1)} \\ b_4^{(1)} \\ b_5^{(1)} \end{bmatrix} \quad (4.12)$$

Output neuron has weights

$$W^{(2)} = \begin{bmatrix} w_{1,1}^{(2)} & w_{1,2}^{(2)} & w_{1,3}^{(2)} & w_{1,4}^{(2)} & w_{1,5}^{(2)} \end{bmatrix} \quad (4.13)$$

bias

$$b^{(2)} = \begin{bmatrix} b_1^{(2)} \end{bmatrix} \quad (4.14)$$

and transfer function γ_2 .

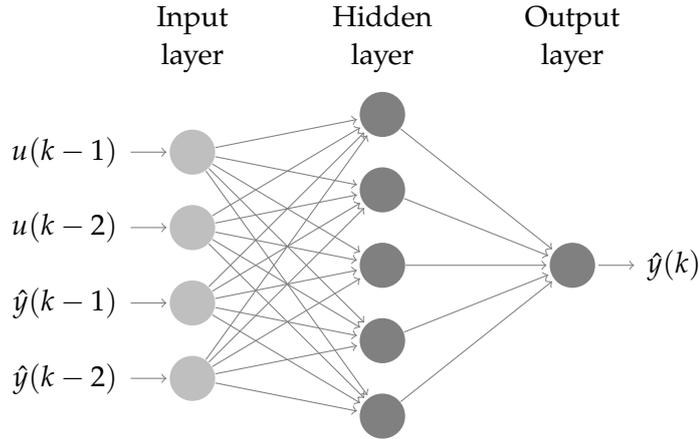


Figure 4.5: A closed loop NARX network with $d_u = 2$, $d_y = 2$ and 5 hidden neurons

As was shown with the perceptron, the activation of a neuron is obtained by computing the weighted sum of previous layer, biasing it and feeding it to the transfer function.

The governing equation of closed loop NARX is then

$$\hat{y}(k) = \gamma_2[W^{(2)}(\gamma_1(W^{(1)}\xi(k-1) + b^{(1)})) + b^{(2)}] \quad (4.15)$$

where

$$\xi(k-1) = \begin{bmatrix} u(k-1) \\ u(k-2) \\ \hat{y}(k-1) \\ \hat{y}(k-2) \end{bmatrix} \quad (4.16)$$

This equation produces an estimated output $\hat{y}(k)$ when given estimated outputs and inputs at least at time $k-1$. Equation can be shifted in time of one time delay to explicit prediction.

$$\hat{y}(k+1) = \gamma_2[W^{(2)}(\gamma_1(W^{(1)}\xi(k) + b^{(1)})) + b^{(2)}] \quad (4.17)$$

In the next section, NARX training algorithms are compared.

4.9 Training method and training algorithm

The training method selected for NARX in this thesis is the supervised training. This method was also applied in [7] and it is chosen because the input and desired output are both available. The concept of supervised training is providing the network input and desired output, and then to adjust the network parameters to minimise the error between desired output and estimated output. The weights are adjusted with gradient based methods. Gradient based methods use computation

of the gradient of the Jacobian matrix. The Jacobian matrix includes the partial derivative of each error with respect to the weights. In a very intuitive way, the gradient indicates in which *direction* the weights should be updated in order to minimise the error. Since the weights are updated starting from the output error and proceeding backwards (from right to left in architecture like the one in Figure 4.5), this is referred as error back-propagation. Error back-propagation involves numerical computations and training algorithms are used to optimise these numerical computation, in order to have network outputs converge to desired outputs faster and more accurately. Training algorithms are then functions that update weights and biases according to some optimisation. MATLAB provides three pre-coded training algorithms within its Neural Network Toolbox: the Levenberg-Marquard Algorithm (LMA), the Bayesian regularisation (BR) and the Scaled Conjugate Gradient (SCG). The LMA is proven to be fast and stable [17] and was successfully used in [7]. For these two reasons, LMA will be used in this thesis as training algorithm.

4.10 Chapter conclusion and ANN limitations

In this chapter, a basic knowledge on ANN was given to the reader. This knowledge is considered enough to understand the behaviour of a network. In particular, NARX network architecture was described in detail, since it will be used for gas leakage detection in the next chapter.

To conclude the theoretical part on ANN, it is very important to remark that ANN are universal approximators under certain condition, and when used to solve identification problems they are *black-box* models. It is possible to reach the a high desired accuracy on estimated output of a ANN when using training data, but when the input to network is different from the one used for training, it can not be guaranteed that model is still accurate. In other words, out of the *safe zone* of the training data, ANN estimated output is not guaranteed to be close to real output of the process. This is a limit of all black-box model structures in general.

Chapter 5

FDD with NARX network on simulated data from accumulator model

In this chapter, a NARX network will be trained to detect a gas leakage in a hydraulic accumulator. The data used to train the network were obtained from accumulator model simulation as presented in Chapter 2. In this thesis, a model-based FDD is performed. Knowledge about the process is required to select which process output(s) are carrying fault information and can be used to obtain fault features.

5.1 Residual generation using open loop NARX

The approach used in this chapter is summarised in Figure 5.1. A NARX open loop network that is trained with data from a simulated healthy process (high pre-charge pressure) is able to perform one-step-ahead prediction \hat{y}_{ol}^{180} , where *ol* stands for open loop. Network is named $NARX_{ol}^{180}$ because it is trained with a pre-charge pressure of 180 bar. Note that network in Figure 5.1 is in open loop

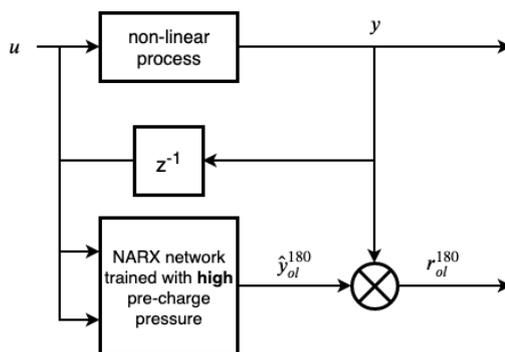


Figure 5.1: Open loop $NARX_{ol}^{180}$ network for residual generation

form, since delayed measured output is accessible to network. When measured output is compared with estimated output, residuals r_{ol}^{180} are generated.

$$r_{ol}^{180} = y - \hat{y}_{ol}^{180} \quad (5.1)$$

These residuals are selected as fault features. If $NARX_{ol}^{180}$ network is an accurate model of the healthy process, residuals generated when network is fed with healthy input/output data, should ideally be zero. Zero residuals represent normal behaviour. Non-zero residuals are considered symptoms of gas leakage. The success of these FDD method depends on the model accuracy since the change in the residuals could not be caused by a fault but by model uncertainties, therefore detecting a fault in a healthy process. Before using $NARX_{ol}^{180}$ for output estimation the network has to be trained.

5.2 Training $NARX_{ol}^{180}$

As suggested by [16], training of a NARX net should be done in open loop. A good starting point for NARX training is MATLAB Neural Network Time Series App. This GUI can be used to generate a MATLAB script for network training. With the help of the script, training can be easily repeated by just running it again. Training an ANN indeed is a trial and error process, since there is no general rule for an initial network features (number of hidden neurons and time delays).

5.2.1 Choice of input and output variables scaling

The NARX network is able to process multiple inputs and outputs. In this thesis, the supply pressure is selected as network output and the load flow rate as network input for two main reason. First, because the impact of a gas leakage on supply pressure was already investigated in [8], proving the variation in load flow are amplified in the supply pressure. Secondly, the supply pressure can be measured in a WT. If pitch activity is also known, and load flow rate can be estimated from it, the same NARX network architecture could potentially be trained and used on experimental data.

$$y(k) = p_s(k) \quad (5.2)$$

$$u(k) = q_{load}(k) \quad (5.3)$$

Following [11] approach on signal processing, network inputs are scaled between 0 and 1, because neurons activation is within this range.

$$q_{load,sc} = \frac{q_{load}}{q_{load,max}} \quad (5.4)$$

During training, process output as well has to be scaled, since it is used as target for network training. After training is finished, network estimated output \hat{y}_{ol}^{180} is

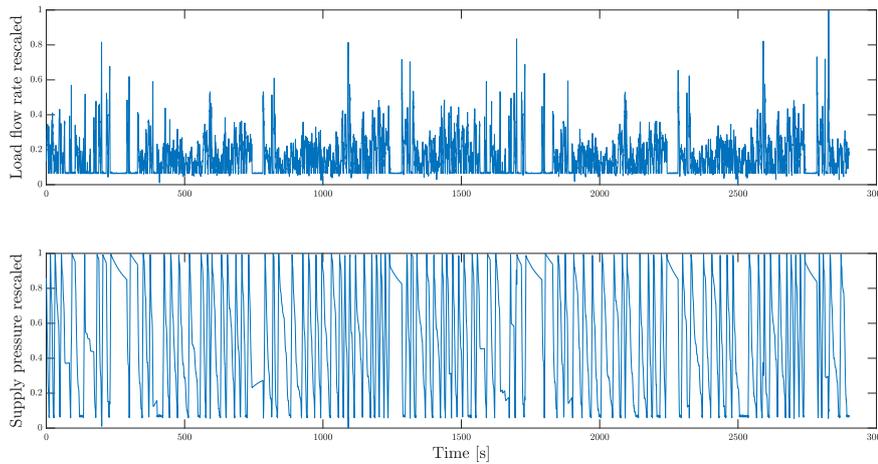


Figure 5.2: Scaled training data

then scaled back within the range $[y_{min} \ y_{max}]$. A q_{load} time series of 3000 seconds (5 min) is used to simulate a supply pressure time series output. The scaled time series are shown in Figure 5.2.

5.2.2 Training data division

Training targets are split into three:

- training (70%)
- validation (15%)
- testing (15%)

Percentages are suggested by MATLAB. Division percentage can be adjusted, but since this training data is obtained with simulation and any desired amount of data can be produced, it is decided to keep percentages suggested by MATLAB. Training targets are presented to network during training and weights and bias are adjusted according to training targets. Validation targets are used to observe the how in-training network performs with data it has never seen and therefore to improve network generalisation. When generalisation stops improving, the training stops as well. Testing targets are not used in training and provide an independent measure of network performance.

5.2.3 Number of hidden neurons and time delays

Since there is not a rule to determine the best number of hidden neurons and time delays d_u and d_y , it is decided to start with a simple architecture as the one pre-

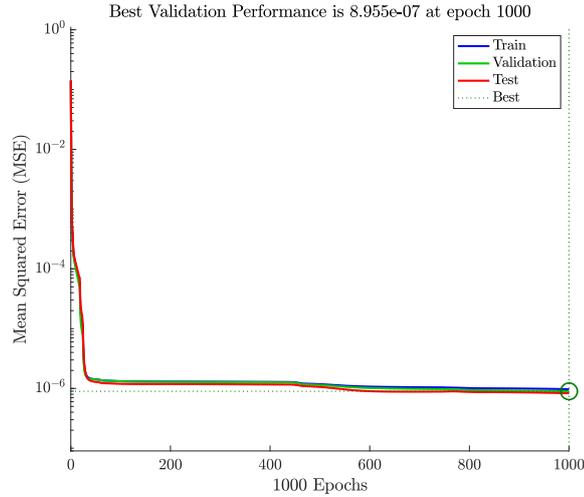


Figure 5.3: Performance of open loop network

sented in Figure 4.2, with 5 hidden neurons and 2 time delays for both input and output. Fewer hidden neurons result in less training time, but as a general rule more neurons improve model accuracy [11]. At the same time, high number of hidden neurons increases the potential overfitting. Overfitting happens when the performance of the network on training targets keeps improving but the performance on test targets gets significantly worse [16]. For feedforward neural networks, performance is often defined as a sum of squared error (SSE), also called Mean Squared Error (MSE).

It is decided to start with a simple architecture and to evaluate the open loop performance in a fault detection scheme as in Figure 5.1. If the residuals of the open loop network can not be used for model-based fault detection, a more accurate model needs to be obtained by increasing network accuracy.

5.2.4 Open loop network

$NARX_{ol}^{180}$ is trained with training data from a high pre-charge process. Training is performed several time because at every training a new set of random initialised weights and biases is used. MSE is always found around 10^{-6} , as shown in Figure 5.3.

The open loop network architecture is shown in Figure 5.4. MATLAB automatically selects the transfer functions γ_1 and γ_2 . A hyperbolic tangent is selected for hidden neurons and a linear function for output neuron. Open loop network $NARX_{ol}^{180}$ is considered a good model because of its low values of MSE and because the network response shows a good fitting as shown in Figure ???. Before using $NARX_{ol}^{180}$ in the FDD scheme, some considerations on closed loop are made.

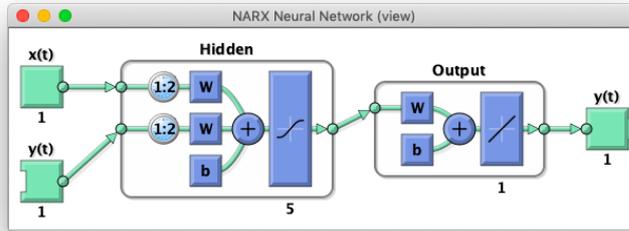


Figure 5.4: Open loop NARX trained with high pre-charge process $NARX_{ol}^{180}$

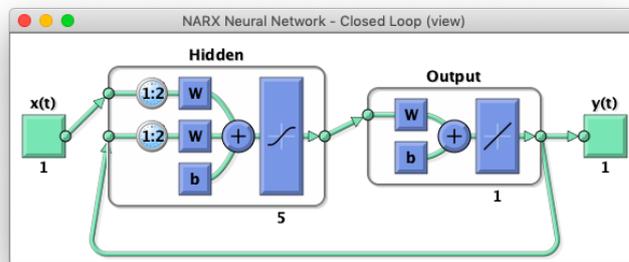


Figure 5.5: Closed loop NARX trained with high pre-charge process $NARX_{cl}^{180}$

5.3 Training results of closed loop network $NARX_{cl}^{180}$

In [7], following MATLAB recommendations, NARX is trained in open loop (series-parallel) and then transformed in closed loop (parallel) configuration for multi-step ahead prediction. Open loop NARX is said to have the drawback of being too accurate. In a FDD scheme where the NARX is intended to replicate the healthy process and generate non-zero residuals when exposed to a faulty process, an open loop network could predict the actual faulty output, generating zero residuals and not detecting the fault. Following this recommendation, $NARX_{ol}^{180}$ is turned in close loop. Closed loop configuration is named $NARX_{cl}^{180}$ and can be seen in Figure 5.5. $NARX_{cl}^{180}$ response on a high pre-charge process and network residuals are shown in Figure 5.6. It is clear that closed loop network is not a good predictor. This is due to the fact that closed loop network is solely relying on past inputs and past predicted output to make its prediction. Once the predicted output starts being inaccurate, the worse following predictions are. In order to obtain a more accurate closed loop network, open loop could be retrained with more hidden neurons and delays until closed loop performance is satisfying. Given such a bad

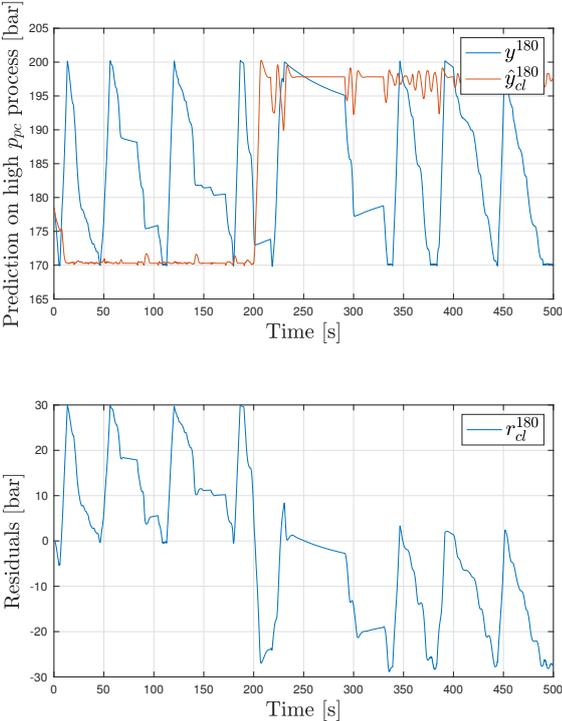


Figure 5.6: $NARX_{cl}^{180}$ response and residuals

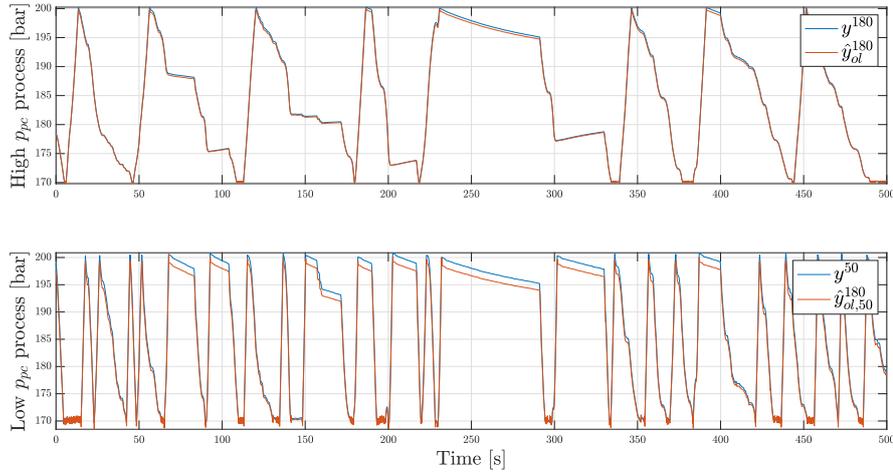


Figure 5.7: $NARX_{ol}^{180}$ response on high and low pre-charge pressure process

prediction though, it is decided not to increase closed loop performance, but to analyse the residuals of the open loop network $NARX_{ol}^{180}$.

5.4 FDD results of open loop network $NARX_{ol}^{180}$

In this section, it is investigated whether $NARX_{ol}^{180}$ could be used in the FDD scheme.

First, in order to understand if the model is too accurate, two extreme situations are considered: a healthy process with high pre-charge pressure and a faulty process with low pre-charge pressure. Being a model of the healthy process, when used for prediction on a faulty process, $NARX_{ol}^{180}$ should not follow the faulty output.

Figure 5.7 shows that $NARX_{ol}^{180}$ tracks the healthy process as expected. At a first look, the drawback of open loop network being too accurate seems confirmed. To some extent, model tracks also the faulty process. Therefore, it is decided to analyse the residuals r_{ol}^{180} , $r_{ol,50}^{180}$ more in detail. In Figure 5.8, the residuals of a medium pre-charge pressure process $r_{ol,100}^{180}$ are also included. The three residual signals visually differ from each other in mean and standard deviation. As expected, the residuals with mean closest to zero and smallest standard deviation are obtained when the model is used on the healthy process. The more pre-charge pressure decreases, the more residuals obtained with $NARX_{ol}^{180}$ increase in mean and standard deviation as shown in 5.1. This trend in the residuals is reasonable and sounds promising for fault detection. In this thesis, gas leakage fault can be detected by analysing open loop NARX residuals.

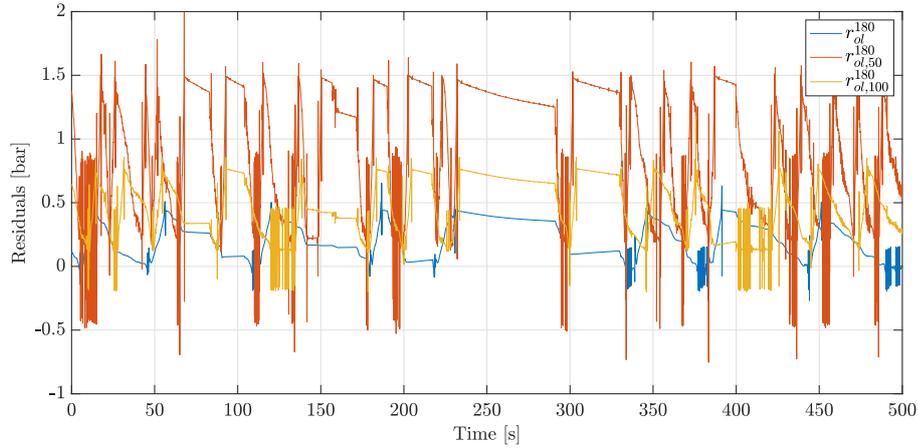


Figure 5.8: Residuals obtained with $NARX_{ol}^{180}$ for different pre-charge pressure processes

	r_{ol}^{180}	$r_{ol,100}^{180}$	$r_{ol,50}^{180}$
mean [bar]	0.268	0.455	0.9074
standard deviation [bar]	0.1459	0.2244	0.4950

Table 5.1: Residuals of $NARX_{ol}^{180}$

In fact, it has observed that these residual have order of magnitude of a fraction of bar. Therefore it is believed that open loop configuration is not suitable for fault detection when measurement noise is involved. This method would not be robust to measurement noise, since open loop configuration uses measured output for one-step-ahead prediction. The drawback of prediction being too accurate with open loop is confirmed.

It is concluded that the closed loop configuration is the best choice when using NARX for FDD. Improving closed loop NARX to generate residuals for model-based FDD is worth to try, but it is unfortunately not done in this thesis due to lack of time.

Chapter 6

Conclusion

As pointed out at the end of Chapter 5, residuals generated by a NARX neural network in open loop configuration were able to detect a gas leakage fault. The order of magnitude of residual (a fraction of bar) however, is very small compared to the operating range considered for the supply pressure (180-50 bar).

With refer to the research question in Chapter 3, a NARX network for model-based fault detection has to be trained so that both the open loop and closed loop show relatively good performance. To do so, NARX architecture with only 5 hidden neurons and 2 time delays is too simple and not able to make good predictions. The research question was addressed in two steps. First, the theory on ANN and on NARX in Chapter 4 helped understanding the advantages and limitations of using ANN for identification problems. Then, simulation was used to implement some of the theoretical concepts, improving knowledge and understanding of NARX network. It is believed that the research question was partially answered, because some helpful concepts came up while training a NARX network, but the test of the model obtained has certainly critical aspects.

Even though the final results for FDD were not as good as expected, NARX networks and ANN in general remain a promising approach to reach the objective of the thesis. As future work, it would be interesting to look into a NARX architecture that would perform robustly and stably in a fault detection algorithm. It would also be beneficial to compare network performance on simulated data with performance on experimental data.

To conclude with a personal comment, this was my first approach to ANN. While fault detection was already familiar to me, neural networks were really an untouched topic. This is one of the reason why a lot of time was spent investigating on ANN principles, trying to avoid misleading complexity that this subject certainly involves. I can say I enjoyed the combination of fault detection and neural networks, especially considering how widespread both are nowadays in several fields of engineering.

Bibliography

- [1] Tony Atkins and Marcel Escudier. *hydraulic accumulator*. Jan. 2013. URL: <https://www.oxfordreference.com/view/10.1093/acref/9780199587438.001.0001/acref-9780199587438-e-2944>.
- [2] Peter Chapple. *Principles of Hydraulic Systems Design, Second Edition*. 2. ed. Momentum Press, 2014.
- [3] Martin O. L. Hansen. *Aerodynamics of Wind Turbines*. 2. ed. Earthscan, 2013.
- [4] Fernando D. Bianchi, Hernán de Battista, and Ricardo J. Mantz. *Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Design Advances in Industrial Control*. Springer Science Business Media, 2016.
- [5] Jesper Liniger et al. "Reliability based design of fluid power pitch systems for wind turbines". English. In: *Wind Energy* 20 (2017), pp. 1097–1110. ISSN: 1095-4244. DOI: 10.1002/we.2082.
- [6] Rolf Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. 2. ed. Springer, 2011.
- [7] Niels Christian Bender and Kris Kristensen Riisager. "A Novel Approach for Fault Diagnosis of Hydraulic Pitch Systems in Wind Turbines". MA thesis. Mechatronic Control Engineering: Aalborg University, 2016.
- [8] Jesper Liniger et al. "Signal-based Gas Leakage Detection for Fluid Power Accumulators in Wind Turbines". English. In: *Energies* 10.3 (Mar. 2017). ISSN: 1996-1073. DOI: 10.3390/en10030331.
- [9] Malte von Benzon et al. "Estimation of gas leakage in hydraulic piston accumulators for industrial wind turbines using a model based approach". In: ().
- [10] Siegfried Rotthäuser, W... Backé, and K... F. Knoche. *Verfahren zur berechnung und untersuchung hydropneumatischer speicher*. S. Rotthäuser, 1993.

- [11] W J Crowther et al. "Fault diagnosis of a hydraulic actuator circuit using neural networks - an output vector space classification approach". In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 212.1 (1998), pp. 57–68. DOI: 10.1243/0959651981539299. eprint: <https://doi.org/10.1243/0959651981539299>. URL: <https://doi.org/10.1243/0959651981539299>.
- [12] R. J. Patton, J. Chen, and T. M. Siew. "Fault diagnosis in nonlinear dynamic systems via neural networks". In: *1994 International Conference on Control - Control '94*. Vol. 2. 1994, 1346–1351 vol.2. DOI: 10.1049/cp:19940332.
- [13] Micheal A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.
- [14] MathWorks. *Machine Learning with MATLAB*.
- [15] MathWorks. *Deep Learning vs. Machine Learning: Choosing the Best Approach*.
- [16] Mark Hudson Beale, Martin T. Hagan, and Howard B. Demuth. *Deep Learning Toolbox User's Guide*. MathWorks, 2019.
- [17] M. T. Hagan and M. B. Menhaj. "Training feedforward networks with the Marquardt algorithm". In: *IEEE Transactions on Neural Networks* 5.6 (1994), pp. 989–993. ISSN: 1045-9227. DOI: 10.1109/72.329697.