POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master Thesis

An online method for condition monitoring and prognostics of hydraulic systems

Advisor
Prof. Luigi Mazza

Co-Advisor:
Prof. Andrea Vacca

Candidate
Alberto Ascoli

Anno accademico 2017 – 2018

Abstract

Engineering systems, such as aircrafts, hydraulic, electronic and electrical systems are becoming more complex and are subjected to failure modes that impact adversely their reliability, availability, safety and maintainability. In particular, hydraulic systems are challenging from the condition monitoring point of view due to the non-linear equations that describe behavior of the fluid. For this reason, numerous efforts have been made to improve the reliability of hydraulic system, leading to the development of complex algorithms for diagnostics and prognostics. In the present research, both diagnostic and prognostic algorithms have been developed, considering the case of an hydraulic crane available at the Purdue's Maha Fluid Power Research Center. Based on the architecture, three components have been analyzed as possible faults in the system: the fixed-displacement pump, the meter-in valve and the cylinder. Among all the modern approaches, a data-driven, neural network based method has been exploited based on a simulation model of the machine through which the behavior of the system is predicted. Moreover, a realistic simulation has been designed, in order to be as close as possible to the real system set-up. Then, a validation of this approach has been performed on the target machine. In summary, the diagnostic algorithm is capable to understand the intensity of the fault and to discern which is the component that is failing also during simultaneous failures; the prognostic algorithm can properly estimate the Remaining Useful Life (RUL) based on the Weibull distribution. Notably, both algorithms use a limited set of sensors taking also advantage of the implemented controller.

Contents

List of Figures

Chapter 1

Introduction

In all types of engineering systems, machine failure is a possible cause of severe losses in terms of downtime and costs; last but not the least, hazardous conditions can be experienced by operators after a failure leading to unsafe working environment. Since each component, by design, undergoes deterioration and, after a while, critical failure, the only smart solution to this problem is a intelligent and effective maintenance. In it's broader definition, maintenance is defined in either in the repairing of something which is already faulty or to prevent a possible damage before it will take place. As will be explored in the next sections, in order to to perform maintenance on a engineering system, a monitoring of each component of the machine has to be performed, identifying the which ones are more willing to fail and so establish an intelligent frame able to define the actual health of the system or understand when it is failing.

Once these preliminary analysis are carried on and all the information data are gathered, it is possible to set up the way to use them, keeping in mind the goal of taking preventive actions to avoid the occurrence of critical breakdowns and the related consequences. In this field, the current research brings some interesting highlights, studying an smart framework for diagnostics and prognostics, dealing with both actual health and prediction of future status. The present research will be mainly focused on the condition monitoring of the components of a hydraulic system and the related strategy to define the most effective maintenance approach. Eventually, once all the necessary parameters have been processed, a method to estimate the actual condition of the machine is explored and some investigating on the future life of the component is pursued. In particular, the final purpose of the present dissertation consists in the definition and the validation of diagnostics and prognostics techniques on hydraulic systems. In order to pursue this goal, an hydraulic crane has been chosen as reference machine: monitoring the behavior of this machine, through a numerical model, and introducing some faulty components, as an old pump and a damaged valve and a failing cylinder, it has been possible to design and realize an algorithm able to do as promised. In order to do so, an analysis on the system is done, and a deep understanding of the variable of the machine is provided. Afterwards, both diagnostic and prognostic algorithm are designed and validated. Eventually, the actual experiment set-up is explored.

1.1 Condition monitoring and health management

Reliability has always been a crucial aspect in the evaluation of industrial products and equipments. Good product design is an essential ingredient for high reliability products. However, even well designed products deteriorate over time since they are operating under stresses and loads in the environment. Downtime affects the productive capability of the product and so produces a costs in terms of reduced output. Thus, maintenance has an important role to ensure a satisfactory level of reliability and availability during the useful life of a physical asset. The latest predominant policies try to minimize unnecessary costs and system downtime [1, 2].

The earliest and easiest maintenance strategy is the so called *Breakdown maintenance*, in which the plant is allowed to run until a fail occurs and then restored to good health [2]. This approach may be satisfactory for some systems in which redundancy is present. Although the high simplicity of this approach, the consequences of a system breakdown can lead to dangerous and costly scenarios since breakdowns often occur at the most inconvenient time, creating undesirable disruption to operation. Another huge problem of this maintenance approach failure of a single component can cause further damage on the machine, increasing downtime and costs. As soon as a more complex system is considered, disadvantages become not neglectable and so the maintenance process has to be addressed in a different way.

A distinct improvement on the previously described method is the *Time-based preventative maintenance* where a strict schedule is established for component replacement. The replacing intervals are determined by a combination of manufactures' data and operational experience, based on statistical information [2]: this is a big drawback because there can be high costs associated to premature or delayed replacement. In situation in which failure is not allowed to occur, like aerospace and safety applications, the premature replacement cost can be tolerated. Another big issue is the fact that the actual conditions of the plant are not taken into account, so components are replaced even if still healthy creating additional costs.

Therefore, more efficient maintenance approaches such as *Condition-based maintenance* (CBM) have been implemented to handle the situation [1]. It represents a new maintenance philosophy, whereby maintenance activities are only performed when there is objective evidence of an impending fault or failure condition, ensuring safety, reliability, and reducing overall total life costs [3]. A CBM program consists of three key steps, as also shown in Fig. 1.1:

- Data acquisition, to obtain data relevant to system health;
- Data processing, to handle and analyze the data or signals collected for better understanding and interpretation of the data;
- Maintenance decision-making , to recommend efficient maintenance policies.

Diagnostics and prognostics, ideally, should be incorporated to a CBM system. The distinguishing factor between these approaches lies in the way data are processed [3]. *Diagnostics*' aim is to detect the fault when a failure occurs, taking into account the current condition of the component [4], while *prognostics* tries to estimate the future status of it. The origin of the word "diagnostics" comes from the greek word that indicates the idea of *discerning*. Fault diagnosis is concerned with detecting, isolating, and identifying an impending, or incipient, failure condition in a system. The term fault implies that the

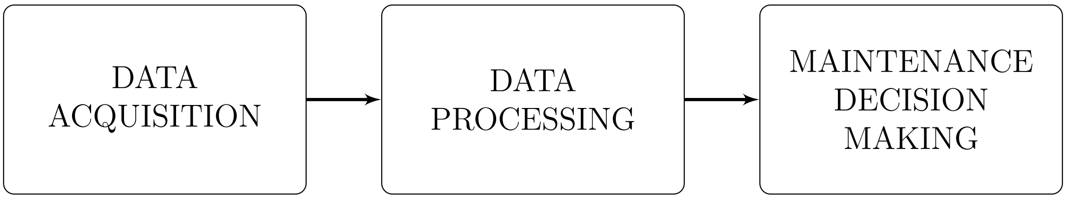


Figure 1.1: CBM main steps.

system under observation is still operational, but cannot continue operating indefinitely without maintenance intervention [3, 5]. The main phases can be then summarized:

- *Fault detection* involves identifying the occurrence of a fault, or failure, in a monitored system, or the identification of abnormal behavior which may be indicative of a fault condition;
- *Fault isolation* involves identifying which component/subsystem/system has a fault condition, or has failed;
- *Fault identification* involves determining the nature and extent of a system fault condition or failure.

To embrace the benefits of a truly condition-based maintenance philosophy additional features, beyond the diagnostics ones, are required. Prognostics capabilities are designed to provide maintenance personnel with insight into the future health of a monitored system [3]. To understand the realm of diagnostic and prognostic capabilities consider Fig. 1.2. At the start of the components life, it is considered to be in proper working order and, after some time, an incipient fault condition develops in the component. As time progresses, the severity of the fault condition increases until the component eventually fails. If the system is permitted to continue operating, there is the potential that further damage may be caused to other secondary components or systems.

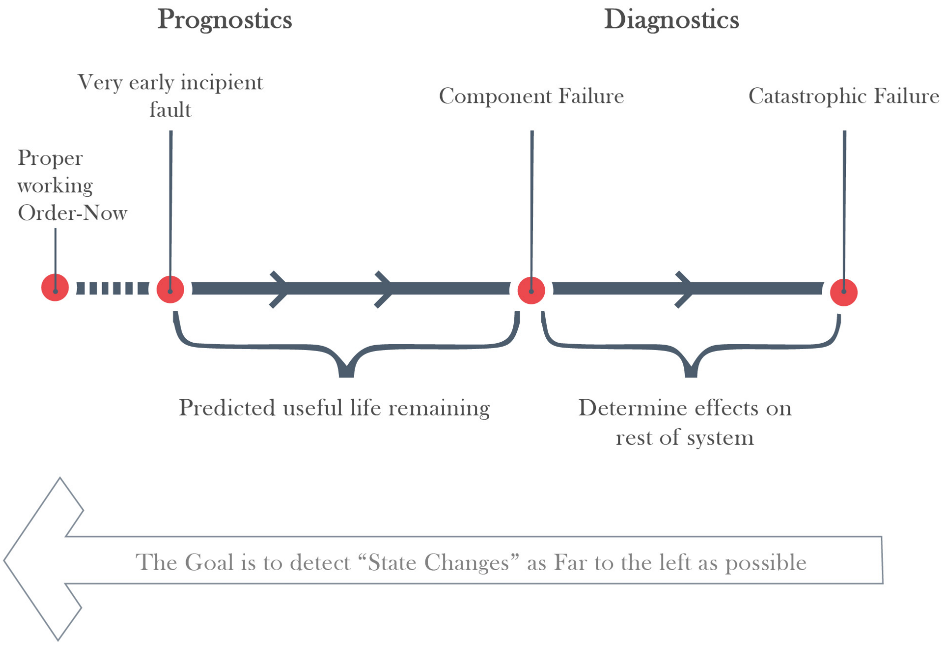


Figure 1.2: Failure progression timeline.

The application domain of diagnostics has typically occurred at the point of component failure, or on the interval between component failure and eventual system-wide failure. However, if a fault condition can be detected at an early incipient stage, then maintenance actions can be delayed until the fault progresses to a more severe state, but before failure occurs. The interval between the detection of an incipient fault condition and the occurrence of failure defines the realm of prognostics. Assuming the existence of a sufficient interval, commonly referred to as the *lead-time interval* (LTI), between incipient fault detection and system failure, a range of operational and maintenance advantages are enabled and realized. With sufficient warning of upcoming maintenance events, remedial work can be planned in advance, with the necessary resources and personnel allocated as necessary. This capability is key to reaping the benefits of a truly condition based maintenance and delivering major costs savings reducing downtime. From a high-level perspective, prognostics has the potential to deliver major improvements over more traditional maintenance approaches along with an increase in safety of operating complex machinery and processes. This differs from more traditional maintenance approaches, in which system failure typically occurs without prior notice, leading to delays in organizing the necessary personnel to restore the equipment reducing downtime as much as possible. Furthermore, the costs associated with equipment failure while working can often be way higher than the costs associated with repairing the failed component, especially in large manufacturing facilities or on safe applications: faults and failure of critical equipment in a manufacturing facility lead to long downtime. Such failures can potentially reduce the overall throughput of a manufacturing facility, resulting in incurred costs which can go far beyond the actual maintenance repair costs.

In short, to enable the benefits of prognostic capabilities, maintenance staff need a reliable and trustworthy estimate of how long a system can keep operate safely, i.e. the *Remaining Useful Life* (RUL) of the system. The generation of accurate predictions of RUL is the challenge presented in the development of prognostic algorithms. Since prognostics is associated with predicting the future, a large uncertainty has to be included. Indeed, the task of prognostics is considered to be significantly more difficult task than diagnostics, since the evolution of equipment fault conditions is subject to stochastic processes which have not yet happened. The ISO standard defines the prognostics as a sequential process with four main steps [6]:

- *Pre-processing*: at this step the system identifies all the existing failure modes along with symptoms and determines the potential future failure modes.
- *Existing failure mode prognostics process*: a study of all existing failure modes is performed, the severity and the Estimated Time To Failure (ETTF) are then calculated.
- *Future failure mode prognostics process* : the most probable future modes, the influence factors between them and the existing modes are estimated.
- *Post-action prognostics*: in the last step, the prognostics system proposes the maintenance actions to be done in order to avoid, reduce or delay the failure mode effects.

1.2 State of the art

Diagnostic and prognostic techniques are anything but unrelated. A lot of literature is present for diagnostics, especially for engineering systems, since maintenance problems started to be costly with the increasing complexity of the technological processes. Compared to diagnostics, the literature for prognostics is much smaller but still relevant efforts have been made for RUL estimation. The approaches are similar for both diagnostics and prognostics and can be divided into *statistical approaches*, *model-based approaches*, and *AI approaches*.

Statistical approaches are a common method used for fault diagnosis based on probability distribution models of failure which goal is to predict when the breakdown is occurring. Since these methods are based on statistics, they rely on historical failure data of the components. A widely applied technique is statistical process control (SPC), in which, if a signal deviates from defined control limits, this may be indicative of a fault condition. Another employed statistical approach is principal component analysis (PCA) and partial least squares (PLS). PCA is often applied to big datasets to reduce a number of related variables to a smaller set of uncorrelated variables [3]. The basic principle of PCA for fault diagnostics is to derive a model using a dataset of normal fault-free behaviour and future observations are then compared with this model using statistical measures: if the measured statistics exceed a defined limit, a potential fault condition is triggered.

Model-based fault diagnostic approaches employ a mathematical model of the system under observation. Using such a model, estimates of system/process outputs are generated which are then compared with the actual process outputs generating a residual signal used to identify potential fault conditions. During fault-free operation, the value of residual signal should be approximately zero, indicating that the model, which describes fault-free behaviour, accurately replicates the actual behaviour of the system. In the situation where the value of the residual signal changes from zero, appropriate processing and analysis has to be applied to the residual signal and then provide to a decision logic routine which is used to map the behaviour of the residual signal onto a specific fault condition. This process is described as *residual evaluation*. The natural consequence of this approach is that a more accurate model will lead to more accurate prediction results. Despite of the accuracy, the main problem with model-based approaches is exactly the intrinsic difficulty in the characterization of a physical model of the system and its equations. Eventually, this kind of approach is strictly related to the system and doesn't allow any kind of portability.

Artificial Intelligence (AI) approaches can be divided into data-driven and expert-based system (ES) approaches. The first category includes all those methods that are mostly based on the input-output data informations, regardless of the exact system that produces those results. These methods are also known as black box models and are strongly related to machine learning and pattern recognition problems, and include also the artificial neural network technique used in this research. The second approach is the expert-based system one and it relies on the knowledge of human experts about the monitored system: the laws of reasoning are translated into IF-THEN logic and used into a suitable mathematical tool, like for example fuzzy logic inference systems. Both methods have, of course, pros and cons. Data-driven approaches result main strength is simplicity: once a suitable amount of input-output data has been collected and provided to the black-box model, it can be trained so that a prediction of future behaviour is possible.

Although the modelling represent one of the main advantages of data-driven method, it may also turn into its main weakness: if not enough data are available, the results can be poor. In case of expert system approach, instead, the numerical computation may be a really tough task in case too many expert-based rules have to be taken into account, with a consequent exponential increasing of computational time. Furthermore, since the algorithm is based on well defined rules, it may really un-adaptive in situations that were not considered during the definition of the laws.

1.3 The reference machine

As previously introduced, the aim of this research is to develop the proposed algorithm and to validate it on the reference machine, an Atlas 125.1 crane, whose structure is shown in Fig. 1.3 These machines are usually truck-mounted and are supplied by fixed displacement

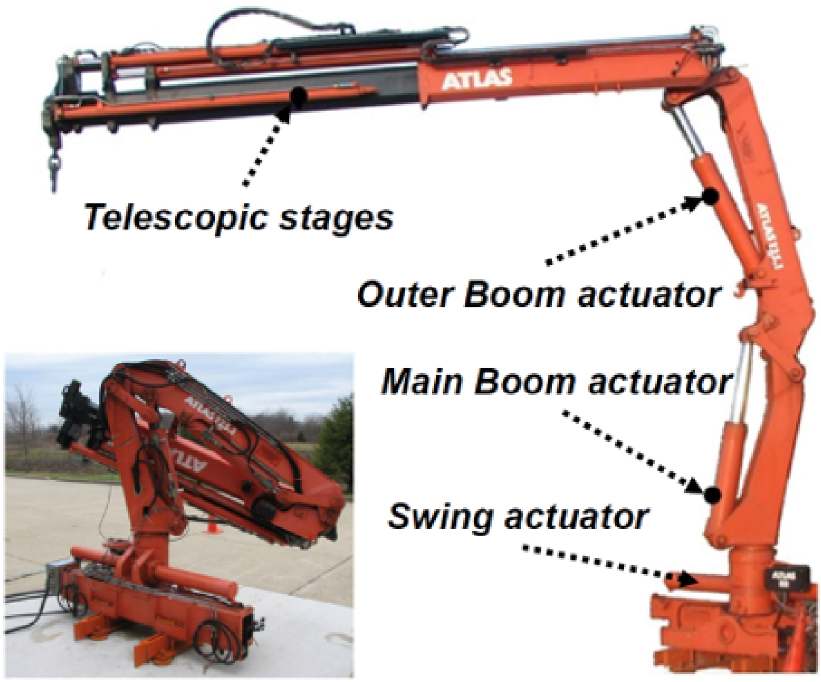
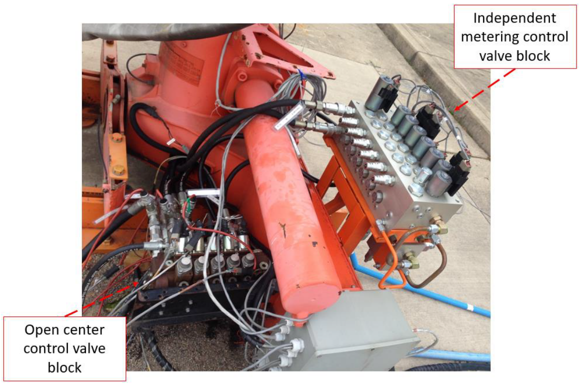


Figure 1.3: ATLAS 125.1 crane structure.

pumps. The arm is operated through four actuators: the swing, the main boom, the outer boom and the telescopic stage. In this activity, only the outer boom cylinder is considered: the swing angular position and the main boom position are always kept constant and the telescopic stages are kept to their minimum extension. A standard case study is considered: starting from a completely folded position of the outer boom of the crane, the cylinder extension for a certain amount of time is performed. Since the working condition is without any payload connected, it is possible to make the diagnostic and prognostic algorithms as independent of the working pressure, maintaining a generalized case study. The machine is equipped with two hydraulic valve blocks which can be interchanged: a standard open center valve block and an independent metering valve block, as shown in Fig. 1.4a: in this activity, the independent metering solution is considered [7]. The power supply block is equipped with a fixed-displacement external gear pump, Casappa® PL-20, with a displacement of 19 cm^3 , powered by an electric motor at 1800 rev/min.



(a) Valve blocks.

(b) Flow generation unit.

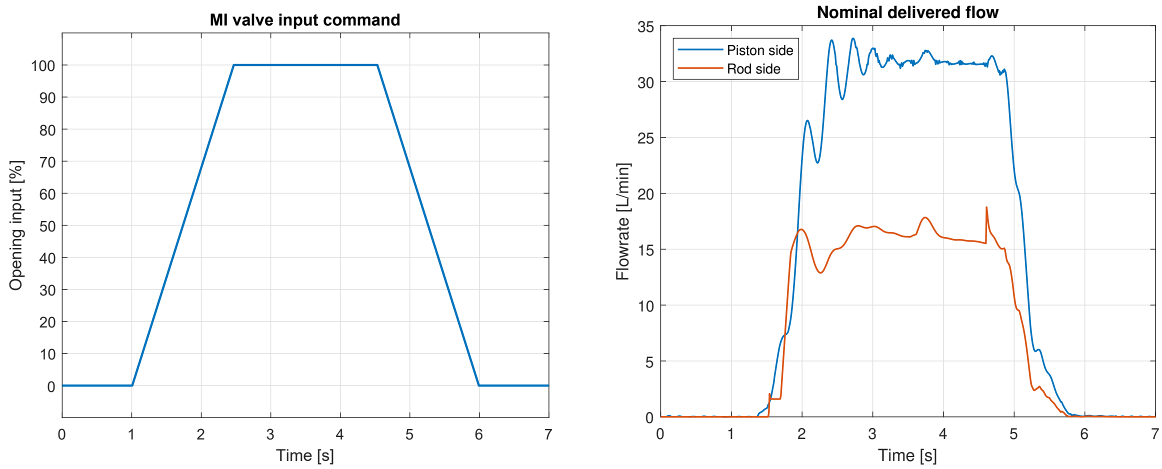
Figure 1.4: Machine equipment.

Furthermore, a priority valve is present in the system to enable the *Load Sensing* (LS) functioning: the LS pressure is equal to the pressure that is needed to move the actuator and so to win load force; it also allows flow-sharing function giving flow to other actuators if requested. A manifold contains the independent metering section for each actuator; each section is composed of four 2/2 cartridge valves, two for the meter-in and two for the control of the discharge flow (meter-out). The telescopic stages are actuated through a standard closed center 4/3 distributor (Parker P70) and a valve for the sequential actuation of the pistons. The system is of *Load Sensing Post Compensated* type (LSPC) and a post-compensator is installed downstream each meter-in valve. In such a system, the pressure drop across the meter-in valve is constant and the velocity of the actuator is directly proportional to the valve opening which is directly defined by the operator. The meter-out valve is independently controlled and therefore an additional degree of freedom is available to improve the machine performance.

For this activity, a numerical model of the system is used: it is built in the *LMS AMESim* environment and it simulates quite accurately the actual behavior of the machine. This tool was also used to understand the behavior of the system along with the control strategy. The controller has been implemented in *Matlab/Simulink* and a Co-Simulation between *Simulink* and *AMESim* has been arranged to replicate the behavior of the machine taking advantage of the controller.

1.3.1 Healthy response

First of all it is necessary to describe the healthy response of the system in order to better understand the behavior of the system in working condition with the temperature of the oil to 40 ℃. Moreover, the working cycle imposed to the machine is the one used for the case study: few seconds are waited without any command to let the solver converge to a stable solution; later on, the input starts increasing with constant slope opening the area passage till the maximum value; the full command is kept for 2 s and then the valve is closed with the same slope as of the opening. The input command percentage for the unfolding of the outer boom cylinder is shown in Fig. 1.5a. Starting from these informations, it is possible to show the initial response of the system to the test stimuli provided. In Fig. 1.5b, the evolution of the flow delivered by the pump to/from the actuator chambers is plotted. Since the system is healthy the maximum flow coincides almost with the nominal one provided by the pump for the piston side, since the meter-in



(a) Opening input provided to the meter-out valve.

(b) Nominal flowrate during the case study.

valve is kept completely open; for the rod side, the flow results to be less due to geometry of the chambers. Moreover, it is possible to appreciate the working condition described: until the second one, in fact, no flow passes through the valve, while it's allowed to pass when the controller command is given.

In order to have a complete overview of the situation it is now possible to show the results related to the another monitoring parameter of the system, the controller input provided to the meter out valve. Fig. 1.5 represents exactly the evolution if this variable across the entire simulation. In the first part, in particular, the valve is throttled in order to keep the control of the velocity of the actuator and to avoid cavitation: if the meter out is excessively opened the return chamber is completely empty and the crane falls down. In the final part of the simulation, instead, the position of the arm is practically out of the overrunning condition and the signal can go to the maximum value to optimize energy efficiency.

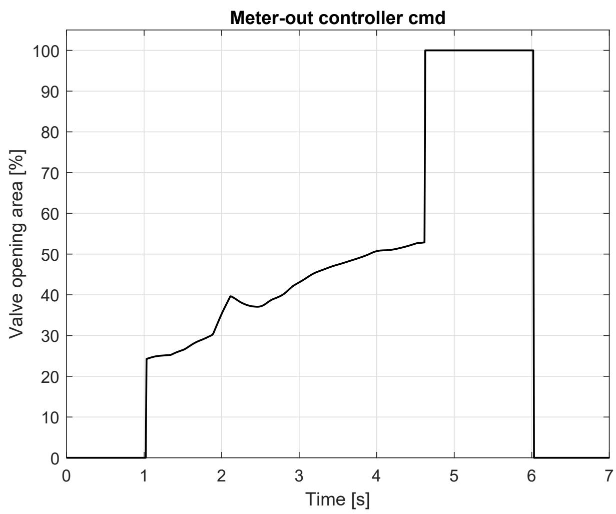
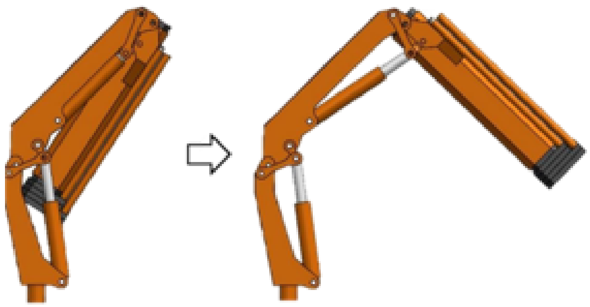


Figure 1.5: Opening input provided to the meter-out valve.

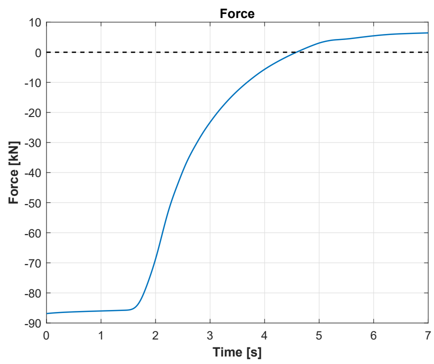
1.3.2 Control strategy

In the independent metering system of the crane the velocity of the actuators is directly proportional to the meter-in valve opening area, defined directly from the operator command. The control strategy in this research is relative to the extension phase of the cylinder (unfolding), but the same strategy can be applied to the retraction phase. The controller distinguishes between the case of resistive load and overrunning load:

- A load is called resistive when it acts in the opposite direction to the motion of the actuator: the meter-out valve is kept completely open to optimize the energy efficiency.
- A load is called overrunning when it acts in the same direction of the motion, helping the actuation: the outlet flow is throttled to avoid cavitation and to keep controlling the velocity of the actuator.



(a) Case study: unfolding of the outer boom.



(b) Force during the extension.

Figure 1.6: Control strategy case study.

In the considered LSPC system, the actuator velocity is proportional to the meter-in valve opening until the LS pressure is higher that the tank pressure and the anti-cavitation valves are closed (meter-out). The limit condition between resistive and overrunning load is given by calculation the force F_{cyl} exerted by the cylinder, neglecting friction effects:

$$F_{cyl} = p_{piston} \cdot A_{piston} - p_{rod} \cdot A_{annulus}$$

where p_{piston} and p_{rod} are the piston side and rod side pressures which are measured through sensors installed in the cylinder. Based on the above relation, if

- $F_{cyl} > 0$, the load is resistive;
- $F_{cyl} < 0$, the load is overrunning.

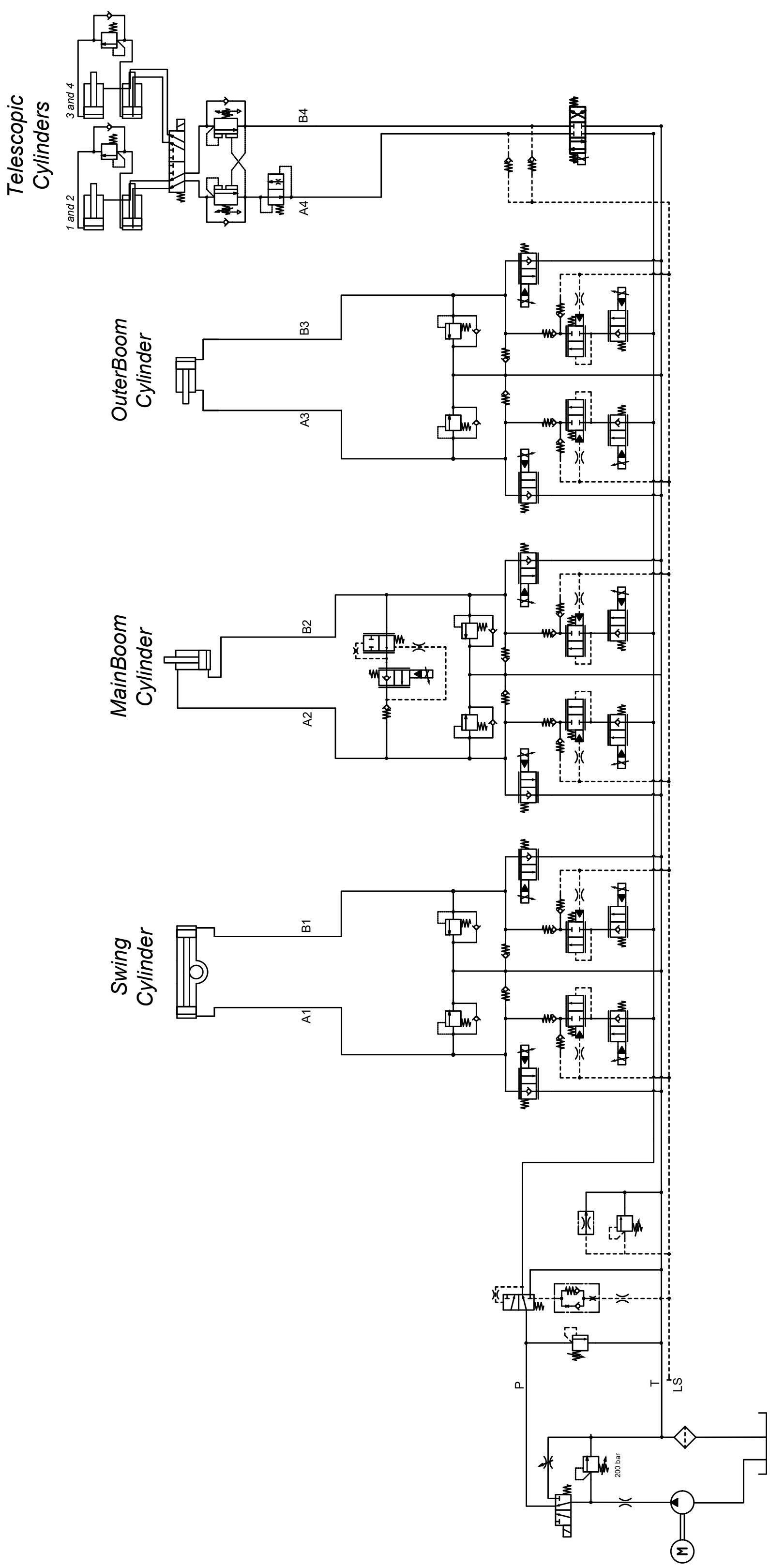


Figure 1.7: Hydraulic ISO schematic of the Atlas 125.1 crane.

Chapter 2

Background

The current chapter is devoted to the understanding of the basic concepts that will be used in the following chapters. Indeed, some terminology is introduced that is peculiar of this dissertation. The first section is committed to the description of the faulty components, to the reasons behind such a choice and, eventually, to their implementation in the modeling environment and on the reference machine. The attention is then moved to the explanation of the main ideas regarding the AI method used for condition monitoring, the artificial neural network, starting from their structure, through the atomic elements description till the algorithms used to define them. Lastly, the main characteristics and applications of Weibull distribution are examined and discussed; in addition particular attention will be paid to the relation between this probabilistic distribution and the failure evolution definition used for this research.

2.1 Failure modes

When facing a large plant operation, with its expected distribution of faults, it's required to introduce a methodological approach to investigate and resolve faults. A popular approach in literature for this problem is the *Failure Modes and Effects Analysis* (FMEA). The FMEA method is a structured approach to fault diagnosis, fault correction, quality improvement, and it combines the considered fault data and the experience of the plant operations team. This process is a continuous investigation and leads to improved understanding of the behavior of the machine. Its main advantages are [2]:

- it aims to recognize and evaluate the actual and potential failure modes.
- it aims to recognize the cause of the failure modes.
- it identifies sections that could eliminate or reduce the chance of failure.
- it documents the corrective process

This approach is meant to analyze a complex system and, among all the possible hazardous scenarios, chose the most threatening ones in order to monitor them. As a result, applying this procedure to a hydraulic system, it is possible to obtain the components which are more meaningful to be analyzed. In this research, this analysis has been done considering most common product failures throughout the fluid power industry. As stated in the *Introduction* chapter, based on the reference machine, three components have been considered important to be analyzed: the fixed-displacement pump, the meter-in electric piloted valve and the cylinder actuator (outer boom manifold). In details, the failures that are considered in this research are:

- Significant flowrate reduction at the outlet of the pump, caused by a loss of volumetric efficiency;
- Progressive spool blockage of the valve, due to distortion and contamination in the circuit.
- Velocity reduction of the actuating cylinder, due to leaks on rod and piston guides.

The volumetric efficiency is a measure of a hydraulic pump's volumetric losses through internal leakage and fluid compression, and it is defined as

$$\eta = \frac{Q_{act}}{Q_{ideal}} = \frac{Q_{ideal} - Q_{leak}}{Q_{ideal}}$$

where Q_{ideal} is the ideal flow rate of the pump, Q_{leak} is the amount of fluid that goes back to the inlet of the pump. Essentially, by increasing the leakages in the chambers of the pump, the volumetric efficiency η is reduced, and the actual flow rate delivered to the system is decreased. The η parameter is never unitary in any real system and its values are strongly reliant on the working conditions. The two main parameters that influence the volumetric efficiency are the speed of rotation, provided by the motor, and working pressure, imposed by the system.

$$\eta = \eta(P, n)$$

In this study a simplified approach is adopted: considering the working conditions in which the reference machine operates, it's possible to make the volumetric efficiency independent from the rotational speed of the motor. This is possible since the considered crane operates on vehicles that provide constant rotational speed at the motor. It's also possible to consider the volumetric efficiency independent from the working pressure since no pay load is applied during the unfolding: in this situation the repeatability of the experiment is guaranteed, and the pressure is taken out of the equation.

The second fault, instead, is one of the most common in directional control valves. It can be caused by several different situations: a distortion of the structure with the corresponding difficulty of motion of the spool; also, contaminations or fluid poor quality may affect the correct behaviour of the valve; eventually, some opening errors can be also related to a malfunction of the solenoid that provides the signal to the valve. In the current research, anyway, the cause of the fault is not deeply analysed, and a generic lack of opening capability is considered.

Regarding the third fault, a hydraulic actuator can suffer from two types of leakages: internal or external leakage. In this case, just internal leakages due to a fault in the sealing are considered, with some fluid going from the rod side to the piston side of the cylinder, reducing the velocity of the actuator.

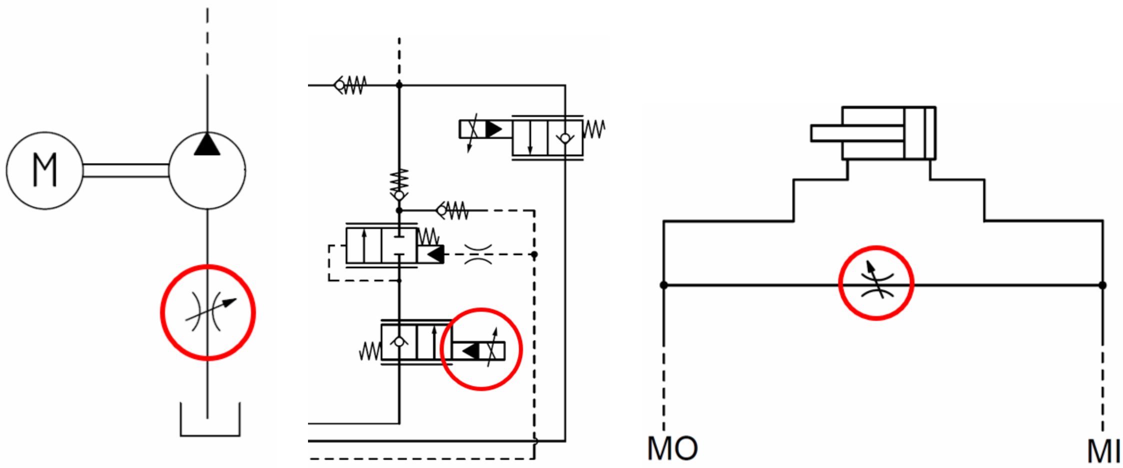
Data-set creation

The proposed algorithm is data-driven, and so it requires tests on the machine in faulty conditions. Since it's not always possible to define how much a component is faulty, the model has been exploited as a virtual test-rig to simulate the system response in healthy and faulty conditions. This approach allowed to collect data to use for the training phase and to test the methodology without executing actual experiments on the reference machine. By changing the value of the parameters in the simulator, it's possible to replicate a fault inside the component and to see the behavior of the system when the fault has occurred. The training and validation data-sets have been created changing the values of the internal parameters in the AMESim environment until a complete failure is experienced. Thanks to the assumptions on known working pressure and constant rotational speed of the motor, the implementation of the degradation of the pump volumetric efficiency can be implemented in simulation changing progressively the internal parameter Volumetric efficiency. Since the goal of the research is to validate the developed algorithm on the reference machine, the validation data-sets used to test the NN, before the experiments, have been created replicating the experiment on the simulator.

To replicate a loss of volumetric efficiency of the pump, without using a faulty component, an orifice upstream the pump has been installed: in this way, reducing the area of passage for the flow, a negative pressure is experienced at the inlet of the pump, now working in cavitation. Cavitation is the formation of vapor bubbles within a liquid at low-pressure regions that occur in places where the liquid has been accelerated to high velocities and it produces extensive erosion of the material, additional noise from the resultant knocking and vibrations, and a significant reduction of efficiency. Using this ploy, it's possible to reduce the flow at the outlet of the pump, simulating a loss in volumetric efficiency.

Regarding the meter-in valve fault, as said above, a progressive spool blockage is considered. To obtain such a result, a suitable degrading gain parameter is introduced in the model to reduce the action of the control input. This fault is easily replicated for the validation data-set imposing a maximum allowed valve opening command.

The fault in the actuating cylinder has been modeled introducing a bypass orifice between the piston chamber and the rod chamber, simulating an internal leaking flow due to faulty seals. This technique has been used for both training and validation data-sets and represents a real testing configuration.



(a) Pump fault. (b) Valve fault. (c) Cylinder fault.

Figure 2.1: Faults experimental replication.

2.2 Artificial Neural Network

The artificial neural networks (ANNs) discussed in this text are remotely related to their biological counterparts [8]. ANNs, due to their ability to learn and generalize non-linear functional relationships between input and output variables, can be regarded as a thinking mathematical structure, capable of handling complex and various situations. They provide a flexible tool for learning and identifying faults inside a system [9]. The first development of ANNs is credited to McCulloch and Pitts [10] in the first half of the 20^{th} century, who developed the theory about how information is learned by neurons within the brain. In recent decades, with the availability of increasing computing power, ANNs have emerged as a powerful tool for classification and regression problems and have been applied to countless different applications across almost every relevant application domain. The primary feature of ANNs, which has made them so popular, is their ability to learn highly-complex nonlinear functional relationships between input and output training data [3]. This feature is extremely helpful when solving different pattern recognition problems. Their another attractive property is the self-learning ability: a neural network can extract the system characteristics from historical training data using the learning algorithm, requiring little or no *a-priori* knowledge about the process. In general, artificial neural networks can be applied to fault diagnosis in order to solve both modelling and classification problems [9]. According to Fausett [11], the general assumptions artificial neural networks are based on, are:

- Informations processing is done my multiple simple elements called *neurons*;
- Signals are shared between neurons over connection links;
- Each connection link has an associated weight and bias value;
- Each neuron applies an activation function to determine its output;

In this research, just *feed-forward* neural networks are considered: connections between the nodes do not form a cycle. A neural network is then characterized by its architecture, its method used to determine the weights on the connection links (*training algorithm*) and its activation function. As said before, the atomic element that represents the center of thinking is called neuron. The McCulloch-Pitts model [10] is the fundamental, classical neuron model and it is described by the equation

$$y = \sigma \left(\sum_{i=0}^{n} w_i u_i + b \right) \qquad i = 1, 2, ..., n$$

where u_i denotes neuron inputs, b is the bias or threshold, w_i denotes synaptic weight coefficients, $\sigma(\cdot)$ is the non-linear activation function. There are many modifications of the above neuron model. This is a result of applying different activation functions. In recent years, sigmoid and hyperbolic tangent functions have been most frequently used [9].
One of the fundamental advantages of neural networks is that they have the ability of learning and adapting. From the technical point of view, the training of neural network is nothing else but the determination of weight coefficient values and biases between the neighboring processing units. The fundamental training algorithm for feed-forward multilayer networks is the *Back-Propagation* (BP) algorithm. It gives a prescription how to change the arbitrary weight value assigned to the connection between processing units

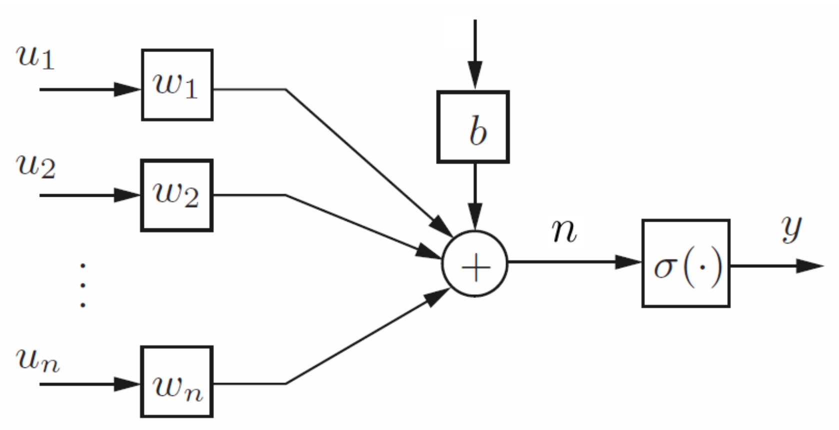


Figure 2.2: Neuron structure.

in the neighboring layers of the network. This algorithm is of an iterative type and it is based on the minimization of a sum-squared error utilizing the optimization gradient descent method [9]. Within the *Matlab* environment, various objective functions can be chosen for the training phase: in this project the mean square error (MSE) between the target output and the estimated one. Besides the above techniques, there are many other modifications of BP, which have proved their usefulness in practical applications, like the *Levenberg-Marquardt* algorithm and the *Bayesian Regularization*.

While training process sets weights and biases of the connection links, the transfer function associated to each layer of neurons has to be imposed in the design phase of the network. According to Demuth [8], nine standard activation functions can be chosen for the neuron: in this project, the so colled *log-Sigmoid* function is used for the hidden layer, while a *Linear* one for the output layer, as explained in Table 2.1. As said before, the overall generic structure of the neural network is given by three or more layers:

- An input layer, which is given by the same number of neurons as the input set;
- One or more hidden layers, that represent the intelligent core of the network;
- An output layer, that prepares the output set to be used;

The main task of the input layer is preliminary input data processing $u = [u_1, u_2, ..., u_n]^T$ and passing them onto the elements of the hidden layer. Data processing can include scaling, filtering or signal normalization, among others. The number of inputs varies

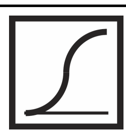
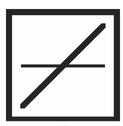
Activation function	Formulation	Symbol
Log-Sigmoid	$y = \frac{1}{1+e^{-n}}$	
Linear	$y = n$	

Table 2.1: Neuron transfer functions $\sigma(\cdot)$ used in this project.

according to the application and has to has to be properly addressed in order to provide useful informations to the network classifier: provide a constant or redundant signal to the optimization process add complexity to the system without producing a real benefit to the classification.

The fundamental neural data processing is carried out in hidden and output layers. It is necessary to notice that links between neurons are designed in such a way that each element of the previous layer is connected with each one of the next layer. The second stratum then is composed by one or more layers of neurons. In many application, one single hidden layer is enough to provide good results. Increasing the number of layers with a small set of training data may lead to a long and inaccurate optimization process: for this reason, also the number of neurons in a single layer has to remain compliant to the data.

The output layer provides the final manipulation of data and thus generates the network response vector $y = [y_1, y_2, ..., y_m]^T$ [9]. In this project, as already said, even following the general neuron structure as in Fig. 2.2, the transfer function used has a different shape. After all these considerations, a graphic idea of the overall structure of a generic ANN can be seen in Fig. 2.3

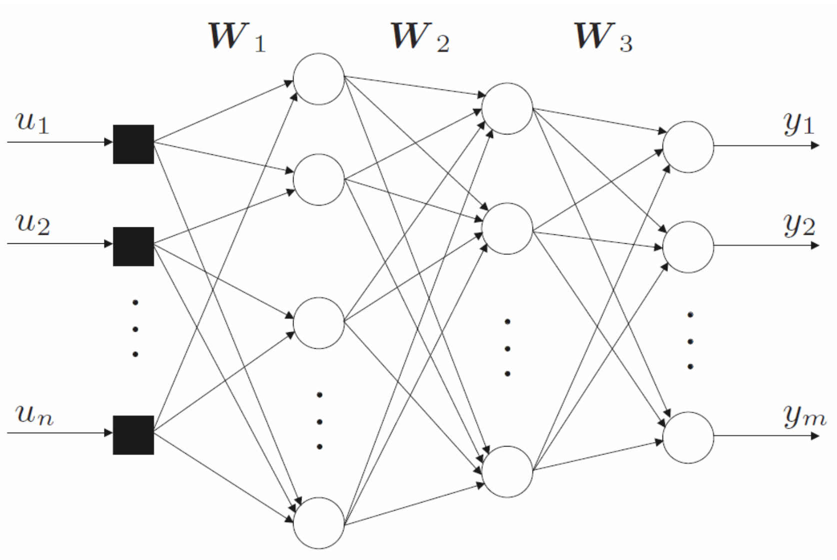


Figure 2.3: General neural network structure.

Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of hidden layers through which data passes in a multi-step process of pattern recognition. Earlier versions of neural networks such as the first perceptrons were shallow, composed of one input and one output layer, and at most one hidden layer in between, as done in this research for the isolated fault analysis. More than three layers (including input and output) qualifies as “deep” learning. So *deep* is a strictly defined, technical term that means more than one hidden layer. In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer’s output. The further you advance into the neural net, the more complex the features your nodes can recognize, since they aggregate and recombine features from the previous layer. This approach has been used for the multi-fault analysis of the diagnostic algorithm.

Bayesian regularization

A brief explanation on this training algorithm is due to the fact that in this research it has been largely used since it provided the best results in this application. ANNs are powerful tools capable to modeling any continuous nonlinear function, given a suitable training data. Surely, some problems may arise as they can overfit data, be overtrained and lose their ability to predict well and also optimization can be time consuming [12]. However, by modifying the standard back-propagation neural network including a regularization step on Bayesian statistics, the benefits can be retained and reduce some of the disadvantages. Bayes' theorem, sometimes called the *inverse probability law*, says that conditional probability can be used to make predictions in reverse, since it's possible to find the conditional probability of an event A given the event B one and the independent probabilities of events A and B:

$$P(A \mid B) = \frac{P(B \mid A)\ P(A)}{P(B)}$$

As explained before, the back propagation algorithm is a training process that leads to minimization of an objective function, that in this project results to be the mean square error. This is an iterative procedure and so it's needed to define a stopping criterion for the optimization process, in order to avoid *over-training*.
Bayesian regularized ANNs (BRANNs) attempt to overcome these problems by incorporating Bayes' theorem into the regularization scheme . Since the goal of this digression is to prove advantages and improvements of this training algorithm, a detailed explanation of the Bayesian Regularization process can be found in [12]. BRANNs show these advantages:

- They are difficult to overtrain, as an evidence procedure provides an objective criterion for stopping training and removes the need for a separate validation set.
- They are difficult to overfit, because they calculate and train on the effective number of parameters. This is considerably smaller than the number of weights in a standard fully connected back-propagation neural net.
- They are inherently insensitive to the architecture, as long as a minimal architecture has been provided.

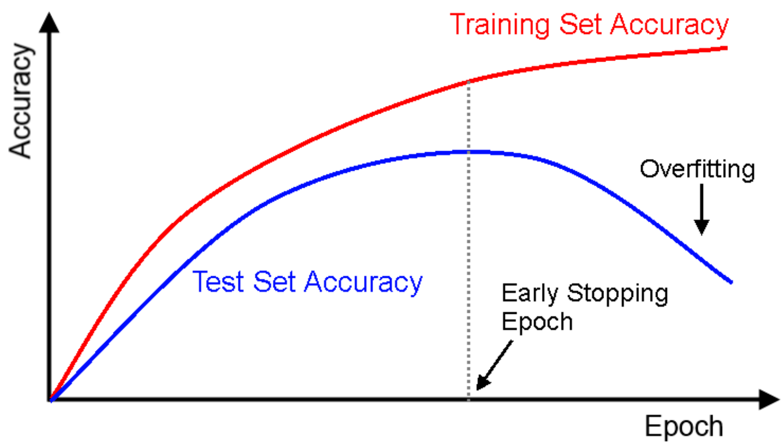


Figure 2.4: Over-fitting and early stopping over epochs.

2.3 Weibull failure distribution

The Weibull distribution is one of the most widely used lifetime distributions in reliability engineering. Indeed, it is a powerful tool in the prognostics context, addressing two main aspects of the problem: it can be used as probability distribution for the length of the life or to model the evolution of a parameter over time. In this thesis, the Weibull distribution is mainly exploited as the description of the evolution over time of the faults that can affect the system, as defined also in Lorenzoni [13]. According to Nelson [14] it can be exploited in accelerated tests to describe mean life duration of a large variety of components, from mechanical elements to electrical one, passing through hydraulic units. First of all it is necessary to introduce the main aspects that characterize such a distribution. The cumulative distribution function of failure, in particular, is represented by the following expression

$$F(t) = 1 - e^{-(t/\alpha)^{\beta}} \qquad t > 0$$

The peculiar variables of the distribution are the *shape parameter* β and the *scale parameter* α, both of them positive definite. Regarding the Weibull distribution as a tool to estimate the average life of a generic component, the variable t represents the actual life spent by the considered unit: increasing the age of the piece, the probability of failure tends to one. As a result, $F(t)$ represents the probability of failure before time t, where t represents the generic age variable considered. In addition, within this field of application of the Weibull function, it is possible to give a deeper meaning to the previously introduced variables. The scale parameter α, also known as *characteristic life of the component*, stretches or contracts the failure distribution along the age axis and it represents the mean duration of the life of the class of elements that contains the analysed one: considering, for example, the case of the investigation of a certain valve, α represents the mean lifespan of the entire population of valves with the same characteristics, based on historical data, and it has the same measurement unit of t. The dimensionless parameter β, also known as Weibull slope, is used to represent the speed of the component to reach a faulty state. Weibull distributions with $\beta < 1$ have a failure rate that decreases with time, also known as infantile or early-life failures; distributions with β close to or equal to 1 have a fairly constant failure rate, indicative of useful life or random failures; while the ones with $\beta > 1$ have a failure rate that increases with time, also known as wear-out failures [15]. This is one of the most important aspects of the effect of β on the Weibull distribution. These comprise the three sections of the classic *bathtub curve*. A mixed

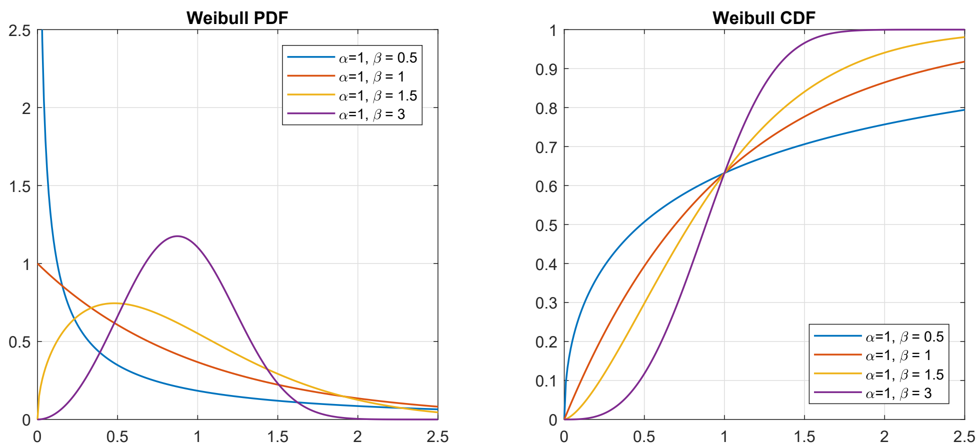


Figure 2.5: Weibull PDF and CDF.

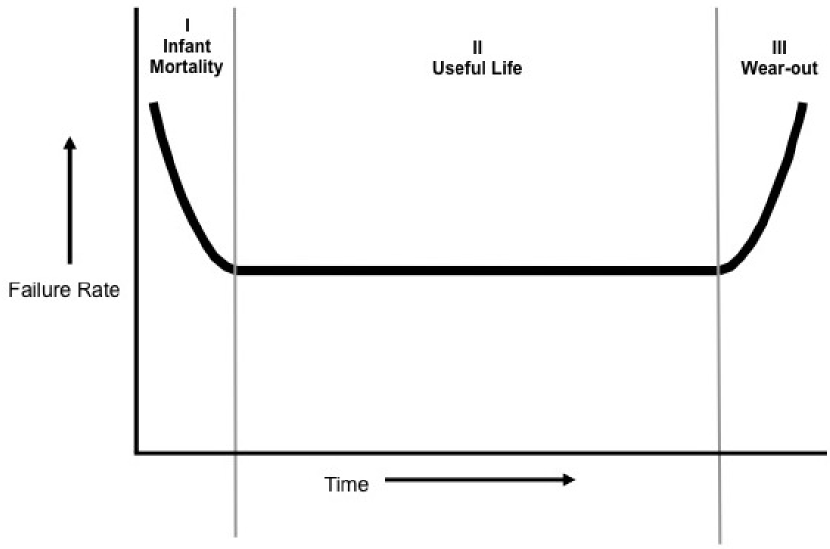


Figure 2.6: Bathtub curve.

Weibull distribution with one sub-population with $\beta < 1$, one sub-population with $\beta = 1$ and one sub-population with $\beta > 1$ would have a failure rate plot that was identical to the bathtub curve. An example of a bathtub curve is shown in Fig. 2.6.
In order to use of the Weibull distribution as a function that can approximate the trend of collected data, it is necessary to change representation and move to the so called *hazard function*. In its broader application, it represents the index that quantifies the risk of failure of a component in the next infinitesimal time instant, provided that no faults have been occurred till that time. It is defined as follows:

$$h(t) = \lim_{\Delta t \to \infty} \frac{F(t + \Delta t) F(t)}{\Delta t \cdot S(t)}$$

where $S(t)$ is the so called *survivor function*, that is the complementary value of $F(t)$, and Δt stands for the infinitesimal time instant considered. Starting from the definition of the survivor function, it is possible to derive the expression as function of α and β

$$h(t) = \frac{\beta}{\alpha^{\beta}} t^{\beta - 1} \qquad t > 0$$

Starting from this mathematical expression, it's possible to define the so called *generalized* or *universal failure rate function*, which can be considered as a mighty tool for data fitting with large noise contributions. In order to achieve this result, two variables have to be added to the standard hazard rate function: a *scale parameter* K, in order to rescale the standard curve in such way that all kind of data set can be approximated and a *bias parameter* Y, necessary to indicate the initial value of the experimental data when the age of the component is zero. As a consequence, the resulting new fitting function can be written as below

$$\hat{z}(t) = Y + K \frac{\beta}{\alpha^{\beta}} t^{\beta - 1}$$

where $\hat{z}(t)$ represents the generic data set that has to be fitted by the approximating function. By properly tuning the parameters, it is possible to approximate almost all kind of experimental data coming from failing systems.

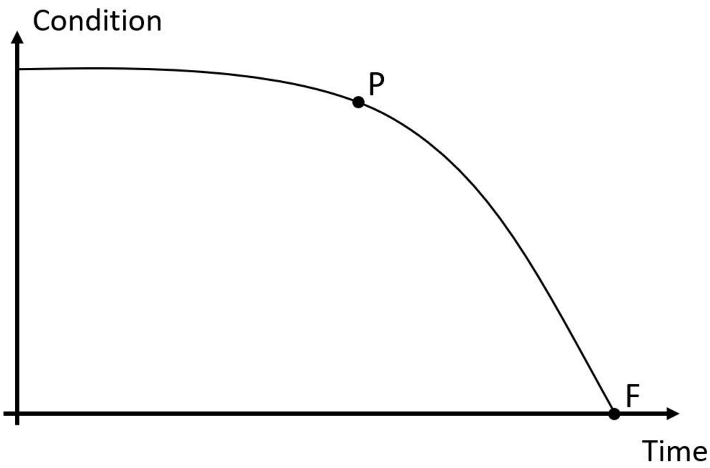


Figure 2.7: P-F curve.

As widely explored in Moubray [16], the condition of a component can be described by a degrading curve where is possible to distinguish two main phases: the healthy phase, when the condition of the element remains practically flat, and as soon as the fault starts to appear, the curve heads exponentially toward the complete failure. The described behavior is shown in Fig. 2.7, where the two represented dots mean, respectively, the point in which the effects of the fault become measurable (P) and the one in which the complete failure occurs (F). In order to precisely build these kind of curves for each component of a system, it would be necessary to monitor the component for its entire life and verify when an incipient fault is detected and when the breakdown occurs. Since this operation is time consuming and expensive, and also really complex due to the fact that each component is different when out of the manufacturing process, an alternative approach has been explored to generate the failure curves. Since one of the goals of this project is the definition of a possible procedure for RUL estimation, the evolution of the fault is imposed qualitatively and the hazardous rate function has been chosen to approximate the shape introduced by Moubray [16]. Using of the four available parameters (K,Y,α, β), it is possible to accurately regulate the shape of the function and so producing several different evolutions over time of the analyzed faults for every component. In this way, different working conditions can be considered.

Chapter 3

Diagnostic algorithm

In this chapter, an explanation of the designed diagnostic algorithm is carried out considering the unfolding of the outer boom actuator. In particular, two main scenarios are considered: the first and simpler one is related to the situation in which just one component fails at a certain time, while the second one includes the analysis of multiple faulty components. Since the diagnostic algorithm is based on a data-driven neural network approach, two distinct data-sets have to be defined for the training and validation phases. Furthermore, since faults usually don't occur independently from each others, several situations in which the faults are advancing together have been taken into account. Thanks to this analysis, a more realistic scenario is considered. As already explained in the *Introduction* chapter, three faulty components are considered in the hydraulic system: the fixed-displacement pump, the meter-in valve and the actuating cylinder [7]. In this research, in order to define the actual health of the system, four levels are defined as described in Table 3.1. These levels are arbitrary but represent a reasonable choice for the approach validation.

The valve fault represents a spool blockage instance, where the spool is not able to open completely due to contamination in the fluid: a 95% command is considered for the healthy case, while a 60% command indicates a component failure; the values in between are intermediate conditions that the algorithm has to detect.

The pump fault, as already explained, pertains to a loss of volumetric efficiency: since a unitary value for the efficiency is not physically feasible, a 0.95 value is considered as healthy condition; in order to maintain the research as much as possible close to the real application, a complete failure of the pump is reached at a 0.6 efficiency value.

A different approach is carried out for the cylinder fault, in which the levels are defined considering the actual flow loss in the piston chamber of the cylinder: a complete sealing is considered for the healthy condition, while a complete failure is considered when more the 20% of the piston flow coming from the meter-in valve goes into the other chamber of the

Health level	Pump efficiency [%]	MI Valve cmd [%]	Cylinder flow loss [%]
L0	95	95	0
L1	85	85	<10
L2	75	70	<20
L3	60	60	>20

Table 3.1: Health levels for the diagnostic algorithm.

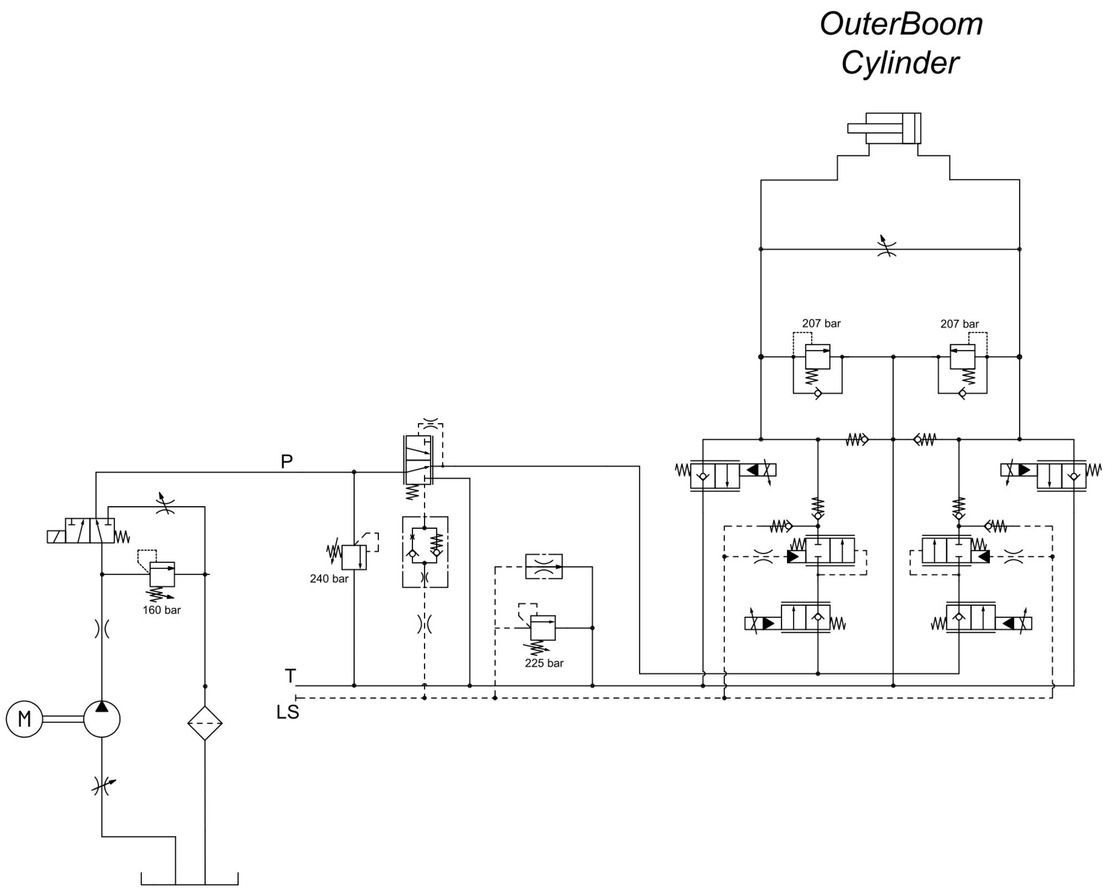
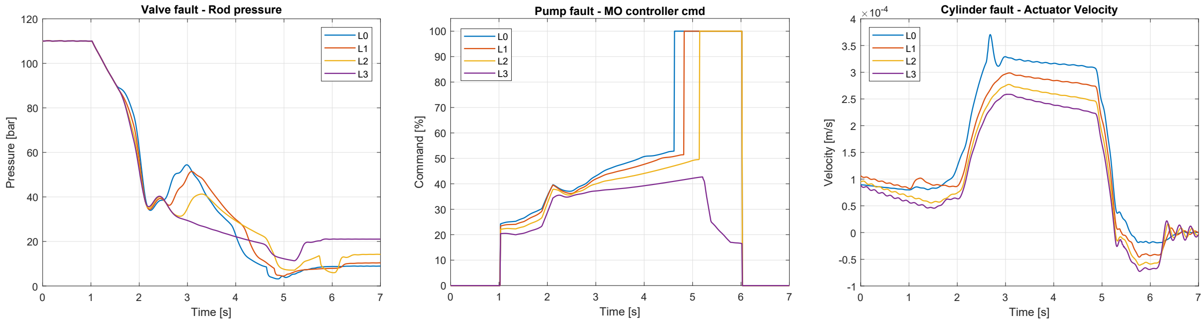


Figure 3.1: Simplified schematic of the system.

piston. A simplified schematic of the interested case study is shown in Fig. 3.1. Following the idea explained in the *Failure modes* section, the meter-in valve fault is artificially and experimentally simulated limiting the input command to the valve; the pump fault is artificially simulated varying the volumetric efficiency parameter in the simulator and experimentally simulated through an orifice mounted at the inlet of the pump in order to let it work in cavitation. The cylinder fault is reproduced both in the simulator and in the actual experiment using a variable bypass orifice between the two chambers of the piston.

3.1 Single Fault

In this section, the isolated faults analysis is considered. Before going into the details of the NN design and the data-sets used for training and validation, a brief introduction about the healthy and faulty parameters is carried on. Eventually, the design of the neural network is explored, along with training and validation data-sets: two groups fault behaviour with different characteristics are simulated to achieve this result. These data-sets are *acquired* and then fed to the neural network firstly as training parameters and later as monitored values for the real-time implementation. In the end of this section, the results of the proposed approach are shown.



(a) Valve fault. (b) Pump fault. (c) Cylinder fault.

Figure 3.2: Parameter curves over time during healthy and faulty situations.

Simulation set-up

Once the health levels for diagnostics are defined, it is necessary to simulate the system in all the different faulty situations. Assuming a constant temperature of 40℃, the circuit is simulated in order to get all the data for training and validation phases. As mentioned before, these data-sets are different in order to provide an unbiased evaluation of the algorithm. Repeatability is ensured with a defined cycle that is repeated in every simulation: few seconds are waited without any command to let the solver converge to a stable solution; later on, the input starts increasing with constant slope opening the area passage till the maximum value; the full command is kept for 2 s and then the valve is closed with the same slope as of the opening. The input command percentage for the unfolding of the outer boom cylinder is shown in Fig. 3.3. Also, the initial conditions are ensured in every simulation thanks to the *Planar mechanics* model: indeed, the force applied to the outer boom cylinder is kept constant during the simulation, since the load is fixed and the main boom angle results to be constant.

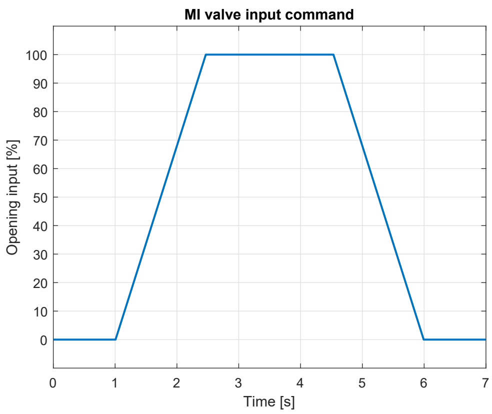


Figure 3.3: Valve input command over time.

Motor speed	$1800 rev/min$
Temperature	$\approx$ 40℃
MB Angle	$\approx$ 82°
OB init. pos.	0 mm
Payload	No

Table 3.2: Initial conditions.

The hydraulic system is modelled in the AMESim environment, and a co-simulation between this simulation tool and Matlab/Simulink is arranged to provide values of the faulty parameters to the model and the controller output, and to save all the information regarding the quantities in the circuit. In particular, since the controller has been designed to unfold both main and outer boom, the output of the main boom is kept constant at zero command.

3.1.1 Design

Following what explained in the *State of the art* chapter, the design of the neural network for diagnostics is carried out in this section. For this purpose, the *Neural Network Toolbox* in the MATLAB environment was used. Considering the basic structure [8–10], three layers were designed: the input layer, the hidden layer and the output layer. During the first part of the research, a deep analysis on the circuit was done in order to choose the smallest set of monitored parameters to feed the neural network on, following a cost-saving policy.

The results of this analysis show that at least four parameters have to be considered as input of the neural network to provide enough information to the NN to be successfully trained, as shown in Fig. 3.4. The first two considered parameters are the pressures in the chambers of the outer boom actuator. The choice is related to the fact that pressure is a crucial quantity in hydraulic systems. The third parameter is the relative position of the cylinder with respect to the initial position: this quantity has been chosen because since all the considered faults slow down the actuator due to a reduced flow at the piston side and for this reason it's a fundamental parameter. Another easier choice could have been the flow delivered at the actuator: flow rate sensors are expensive and, even if one is present on the outer boom at the rod side of the cylinder, the algorithm is challenged to understand the scenarios with different quantities and cheaper sensors. The fourth monitored parameter is the command input provided to the meter-out valve for the unfolding of the outer boom; such parameter is crucial to include the controller in the diagnostic process since the command is provided by the controller itself. As soon as the input data-set has been chosen, a manipulation has been done to enhance the results of the classifier. First of all, since the result of each simulation is an array vector with all the values of each quantity *acquired*, and the neural network works with a single value input, in each simulation, the variance of the quantity is considered for pressures and cylinder position, while the integral is used for the controller command. The variance is defined as the *expectation* of the squared deviation of a variable from its mean and defines how much a set of observations differ from each other. Choosing the variance instead of the mean value is due to the fact that a wider range is exploited.

$$Var(X) = E\left[(X-\mu)^2\right] = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_i)^2$$

Moreover, considering the sigmoid transfer function associated to each neuron, as described in the *Artificial Neural Network* section, it resulted more convenient to normalize the input value of the neuron to $[0, 1]$ range: in this way, since the input dynamics of the sigmoid function sweeps the same range of values, the whole scale is used and effectivity is improved. The normalizing value for pressures and cylinder position is given by the

Input variable	Meaning
p_{piston}	Pressure at piston side of the outer boom cylinder.
p_{rod}	Pressure at rod side of the outer boom cylinder.
x_{cyl}	Position of the outer boom cylinder.
U_V	Input signal to the meter-out valve.

Table 3.3: Neural network input vector.

variance of the quantity, while for the controller command, the integral is considered, all in healthy condition. Once the input set is defined, it is possible to address the description of the target set. In order to accurately train the classifier, a target vector is defined according to the levels in Table 3.1. During the training phase, all these quantities are known and given to the algorithm as informations for the input classification. For each training data-set, a target output is built to estimate the health of the considered components.

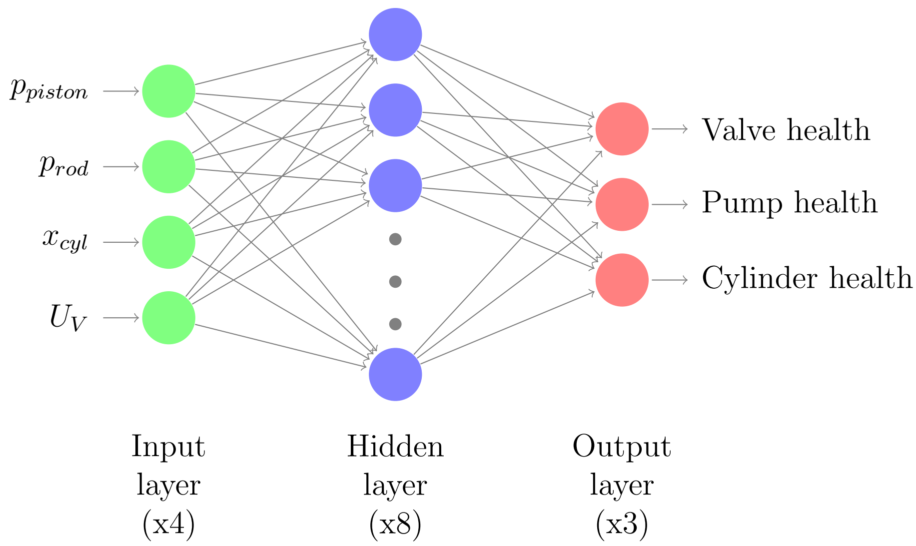


Figure 3.4: Neural network graphical representation - Single fault.

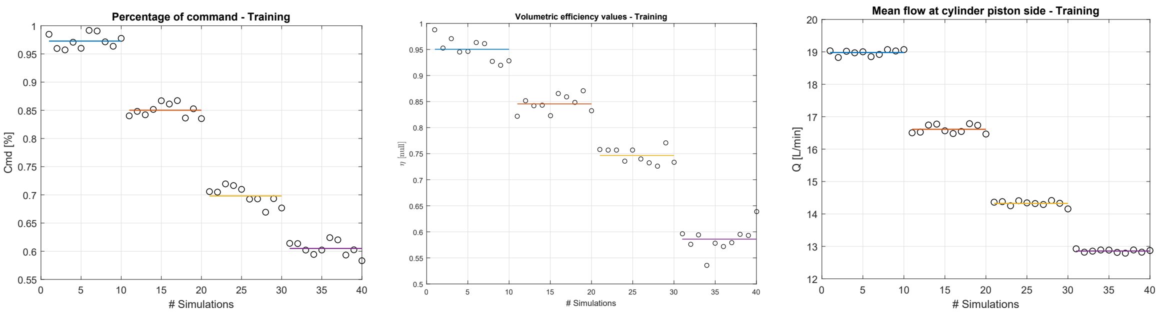
Once input and target data-sets are defined, it's possible to move to design the structure of the NN. A neuron has to be considered for each input vector, so four neurons are considered for the input layer. There is no general rule to choose the number of neurons of the hidden layer, therefore a cross-validation approach is used: after a certain number of added hidden neurons, the solver start over fitting the data and give bad estimates on the test set. The result of this analysis showed that eight neurons in the hidden layer provide the best performances. In details, the values that are modified during the optimization process are the weight matrices related to each layer. The output layer is given by three neurons since the goal of this research is to monitor the faults in the meter-in valve, pump and cylinder: the output is a number in the range [0,1] according to the normalization of the inputs. A graphical representation of the designed NN is shown in Fig. 3.4. Therefore, obtained the monitoring parameters for all the defined conditions and normalizing them as stated, a sufficient set of input data is available and a suitable neural network can be trained. During the design phase of the network several parameters can be customized in order to obtain the best results related to the analyses case, as mentioned in the previous subsection. Apart from the number of neurons, which is imposed equal to eight in this case, the starting weights and biases of each neuron are set up as randomly chosen at the beginning of the optimization procedure. Similarly, the amount of data that are assigned to each group during the training phase are split casually: even if the percentage of data assigned to training, validation and test set is well specified, the input/output pairs that will belong to each of them are randomly chosen. Moreover, promising results have been obtained by assigning 70% of data to training class and splitting the remaining part 15% in validation and 15% in test group.

3.1.2 Training

As broadly described in the *Artificial Neural Network* section, the main goal of the training phase is the tuning of the internal parameters of the network in order to minimize the specified performance index. In details, the values have to be found in the optimization process are the weight matrices of each layer as well as the bias values. Moreover, the objective function to minimize to guarantee performances is the mean squared error (MSE) between the target set and predicted one.

$$\bar{e} = MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{tar_i} - y_{est_i})^2$$

During this training phase, a Bayesian Regularization (BR) based on the Levenberg-Marquadt optimization algorithm is used: a deeper analysis on this training algorithm can be found in the *Artificial Neural Network* section. Furthermore, since an optimization is performed, several attempts were made in order to find the best result: the classifier starts from a different random value every time the algorithm is run, leading to different performances. The training phase starts from the definition of the training data-set that is used to run the simulations and acquire the parameters for the NN. As already explained in the *NN data-set creation* subsection, a white-Gaussian noise is added to the parameters in order to provide a training data-set whose output parameters include also variations that may affect a sensor. In the figure below are shown the most relevant quantities for the training phase: the percentage of command to the meter-in valve, the volumetric efficiency of the pump and the mean flow at the piston side for the cylinder.



(a) Valve fault. (b) Pump fault. (c) Cylinder fault.

Figure 3.5: Neural network training data-set - Single fault.

The results of the training process are shown in Fig. 3.6. On the horizontal axis is represented the case number, starting from the faulty condition 40 simulation are run for each fault (n.10 for each health level) keeping the other two components in an healthy state. On the vertical axis, both target and training output are plotted and the results are positive: the network has been trained properly according to the target data-set. This result is also given by the minimization algorithm used for the training process, in terms of mean square error (MSE) or root mean square error (RMSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{tar_i} - y_{est_i})^2 = 5.0737 \cdot 10^{-6}$$

$$RMSE = \sqrt{MSE} = 0.0023$$

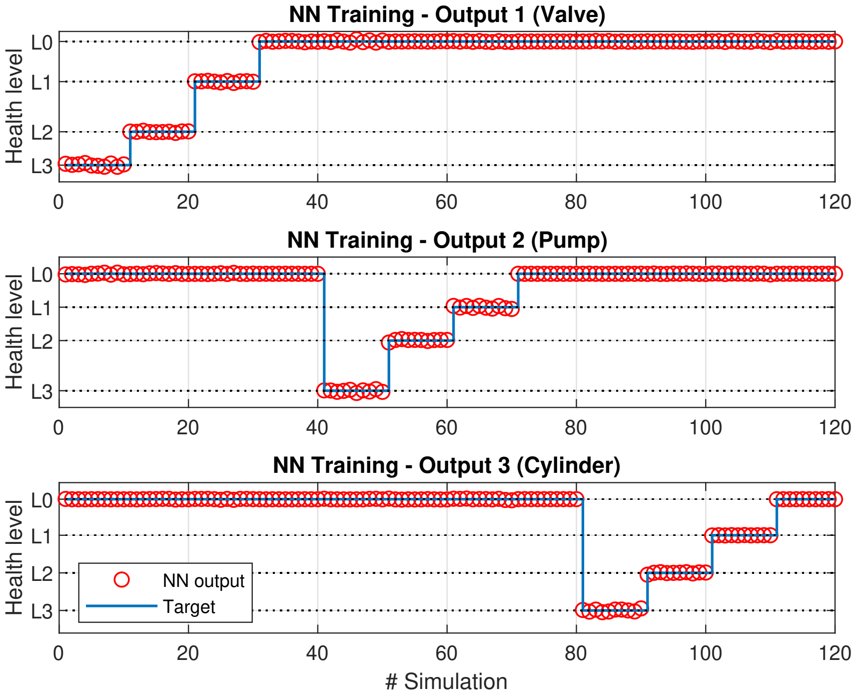
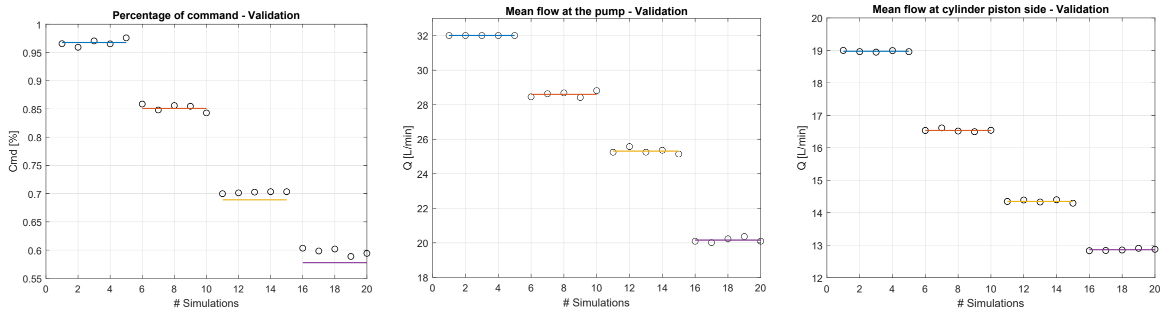


Figure 3.6: Neural network training - Single fault.

3.1.3 Validation and results

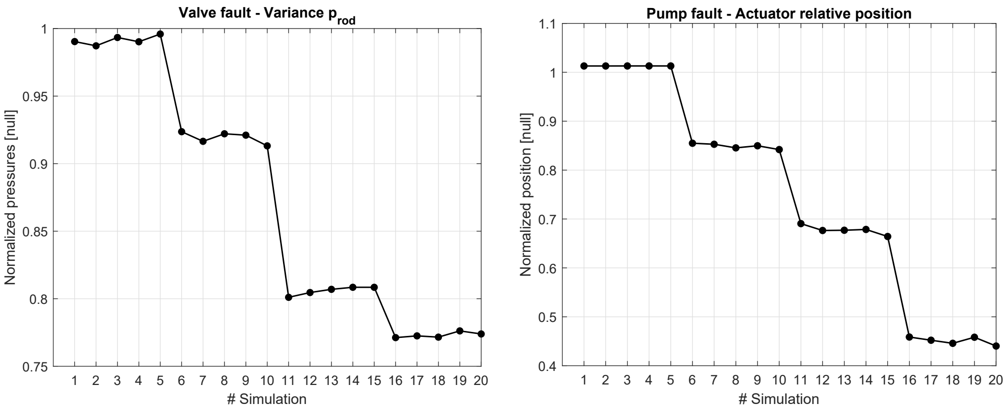
Once exploited the offline training phase, it's necessary to validate its effectiveness by providing a different set of data. As a consequence, in this phase, other fault evolutions for the three components have to be generated, then run the circuit to *acquire* the monitoring parameters to feed the network and verify the output. The idea is equivalent to the definition of the training data-set: adding white-Gaussian noise to the varying parameters, a more realistic scenario is simulated.



(a) Valve fault. (b) Pump fault. (c) Cylinder fault.

Figure 3.7: Neural network validation data-set.

In Fig. 3.7 are shown the most relevant quantities in the validation process. In the same way as for the training phase, the most important quantity for the valve fault is the percentage of command provided to the meter-in valve representing the progressive blockage of the spool; for the cylinder fault, the mean flow at the piston side is considered, based on the assumption that some of the flow provided by the pump is diverted to the rod side and so a loss of useful flow is experienced. Since the pump fault is simulated according to the real test (Fig. 2.1a), the mean flow at the outlet of the pump is considered for the validation as most relevant quantity, since cavitation leads to a reduction of efficiency and flow. The model is run with the new parameters defined for the validation phase and the data acquired from the model are used fed to the previously trained network. In particular, as explained in the *Neural network design* section, the normalized variance of each quantity over the healthy case is performed across the opening time of the meter-in valve, that is to say considering the quantities between second one to six, referring to the input signal in Fig 3.3. The actual shapes of the rod pressure and the cylinder position fed to the network as input are showed in Fig. 3.8: each of the depicted point represents the normalized variance computed within a single simulation; later on, in order to test the network, each of this discrete values have been interpolated to get the overall behaviour of the parameter.



(a) Rod side normalized pressure. (b) Cylinder relative normalized position.

Figure 3.8: Neural network validation input data-set.

Considering the figure above, it's easy to understand that four different levels are reached, that represents the predefined ones explained at the beginning of the chapter in Table 3.1.

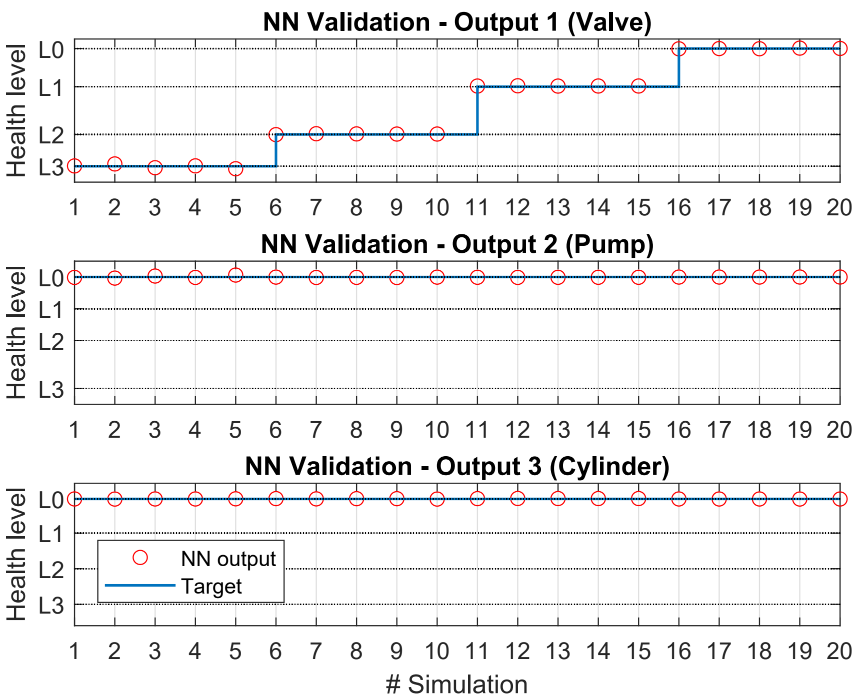


Figure 3.9: Neural network validation - Valve fault.

Applying the validation data-set defined above to the previously trained neural network, it's possible to check if the current approach is working: as can be seen in Fig. 3.9, the validation results for the valve fault are really encouraging: based on the defined target-set, the neural network can actually follow the target, even if the input data-set is different from the one used for the training process. This means that the current approach is validated on this new set of data and eventually tested on the reference machine. During the development of the current approach, the component that gave more trouble to let the algorithm converge to the target set was the pump. One of the reasons is identified in how the validation data-set is generated. Since, as broadly explained in the *Failure modes* section, an orifice upstream the pump is used to reproduce the flow loss, the correct opening area for each of the defined fault levels had to be found in order replicate the same flow loss experienced during the training, done with the variation of the volumetric efficiency parameter in the numerical model. In the latter indeed lies the reason of the troubles undergone by the algorithm since the opening area range of the orifice results to be really small and so the input data-set, during some iterations of the algorithm, couldn't train properly the network, leading to bad results. Eventually, repeating the algorithm and starting from different initial conditions, good results have been reached, as shown in Fig 3.10. The result plot for the cylinder actuator fault is not shown, but the results are

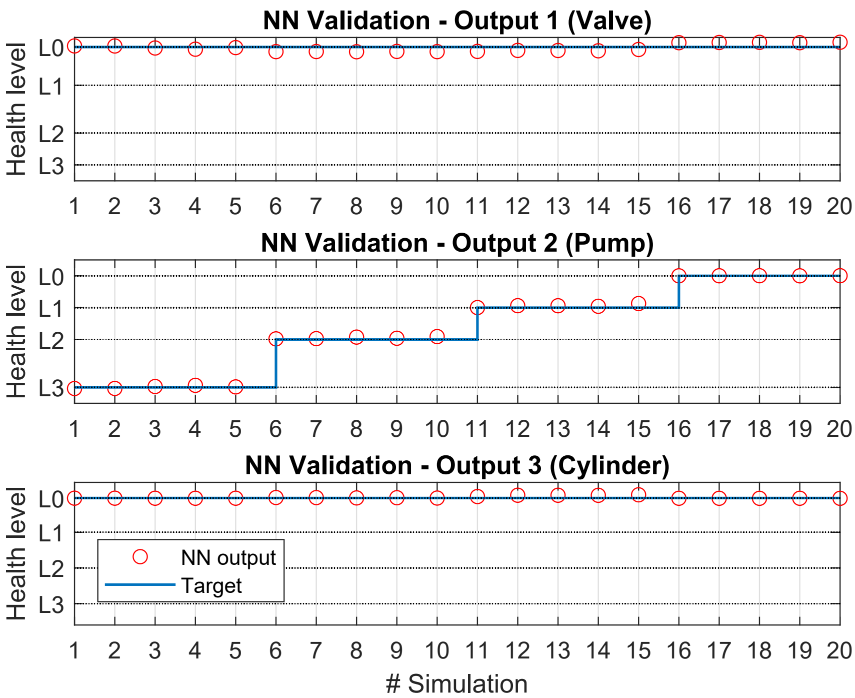


Figure 3.10: Neural network validation - Pump fault.

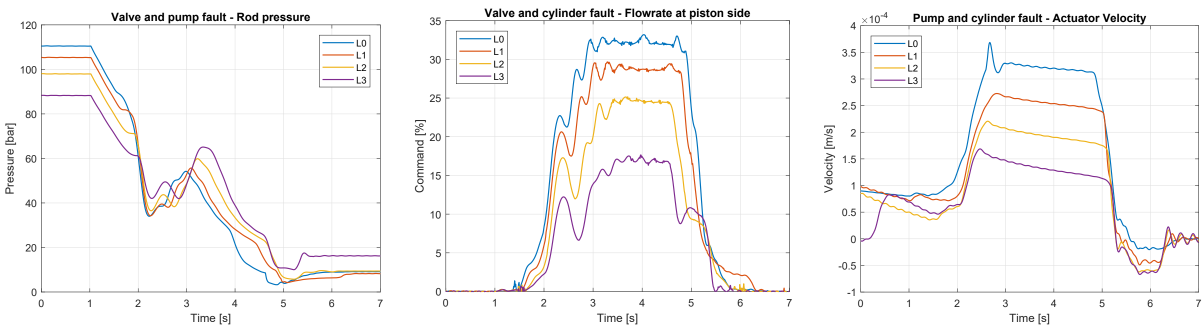
comparable to Fig 3.9 and 3.10: of course, the valve and pump outputs are at the healthy level, while the cylinder output follows the stair shape of the other faults. Analyzing deeply the above plot using numbers, as done for the training phase, the same error can be computed for the neural network for each kind of validation data-set, as displayed in Table 3.4. As can be seen, the neural network fed with pump fault data is the one with highest error, but the result is still acceptable since the trend is well followed. component. Having in mind these results, it is possible to compute a further step by analyzing a harder scenario from the classifier point of view, estimating the health level of the system when two faults are advancing together within the system.

Faulty component	MSE	RMSE
Valve	$5.0463 \cdot 10^{-6}$	0.0022
Pump	$8.5028 \cdot 10^{-5}$	0.0092
Cylinder	$3.2422 \cdot 10^{-6}$	0.0018

Table 3.4: Neural network validation performances.

3.2 Multi fault

In any real system, each part has its own degradation course, which can be either independent or related to other component one. Surely, the complete system health is correlated to the components' different level of degradation and more likely to their interaction. As a result, it is convenient to consider the analyzed faults occurring together to obtain a situation closer to the actual application in the real world. Although reality is much more complex, since the interaction are not limited to just the three considered components, it is a first step to extend the analysis to the overall system. Nevertheless, this kind of approach is still meaningful, based on the considerations done in the *Failure modes* section: the components were chosen based on their influence and importance for the system, and also since they are the components that are more stressed during the normal duty cycle of the machine. As can be seen in the figures at the bottom of the page, comparing with the single fault ones (Fig 3.2), the shape of these curves is different since the system is now experiencing simultaneous failures of two components. In particular, a huge reduction in actuator velocity undergone in Fig 3.11c with respect to the single fault analysis. The possible reason for this behavior can be found in the fact that both pump and cylinder are failing and so at the increasing of the internal leakages, also a reduction of useful flow is faced by the system. Furthermore, for this analysis, it's assumed that the components are failing at the same velocity and so they are every time at the same level of health, based on the Table 3.1. This aspect is crucial to understand the multiple faults analysis: assuming that all the components are new or properly working, they experience degradation in the same way, as it would be expected in a real application. Surely failure can happen in any moment of the life of a component and so this investigation can be applied just with the presence of a model of the system that allows a better understanding during faulty scenarios.



(a) Valve and pump faults. (b) Valve and cylinder faults. (c) Pump and cylinder faults.

Figure 3.11: Parameter curves over time during healthy and faulty situations with multiple faults.

Simulation set-up

Considering the health levels designed for the single fault analysis, as already done, it is necessary to simulate the system in all the different faulty situations. Assuming a constant temperature of 40°C, the circuit is simulated in order to get all the data for training and validation phases. As mentioned, these data-sets are different in order to provide an unbiased evaluation of the algorithm. Repeatability is ensured with a defined cycle that is repeated in every simulation, as shown in Fig. 3.3. Also, the initial conditions (Table 3.2) are ensured in every simulation thanks to the *Planar mechanics* model: indeed, the force applied to the outer boom cylinder is kept constant during the simulation, since the load is fixed and the main boom angle results to be constant. Even if the approach is exactly the same as for the isolated fault, the parameter values used in the models are different in order to let the system experience a simultaneous fault scenario. Moreover, just two degrading components are considered for each simulation: the choice is due the fact that the system is expected to collapse with the failure of two important elements, so adding another one would results in an unrealistic situation. For this reason, all the combinations are exploited to provide a good data-set for the training and validation data-sets. The

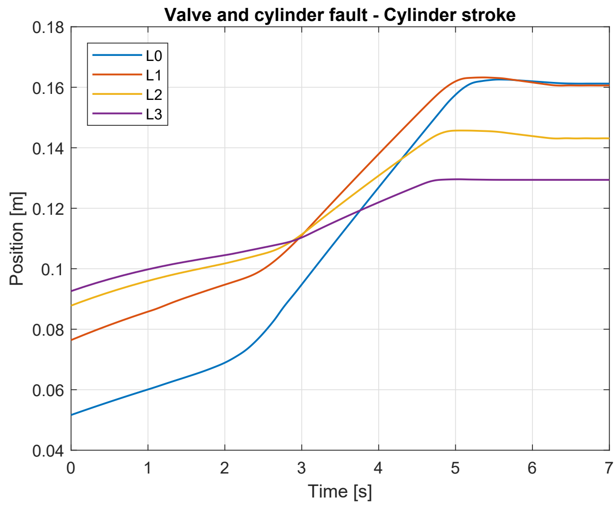


Figure 3.12: Cylinder relative position during the considered cycle.

hydraulic system is the same used before, modelled in the AMESim environment, and a co-simulation between this simulation tool and Matlab/Simulink is arranged to provide values of the faulty parameters to the model and the controller output, and to save all the information regarding the quantities in the circuit. In particular, since the controller has been designed to unfold both main and outer boom, the output of the main boom is kept constant at zero command.

3.2.1 Design

Following what explained in the *State of the art* chapter, the neural network design for diagnostics in multi fault situation is carried out in this section. For this purpose, the *Neural Network Toolbox* in the MATLAB environment was used. A four layers network was designed: the input layer, two hidden layers and the output layer. The choice of a *deep* neural network lies in the attempts done for the training process: using this structure, better results were achieved for both training and validation. The input layer structure is the same as for the single fault: the first two considered parameters are the pressures in the chambers of the outer boom actuator, being them important quantities in hydraulic systems. The third parameter is the relative position of the cylinder with respect to the initial position while the fourth is the command input provided to the meter-out valve during the unfolding of the machine. As soon as the input structure has been chosen, a

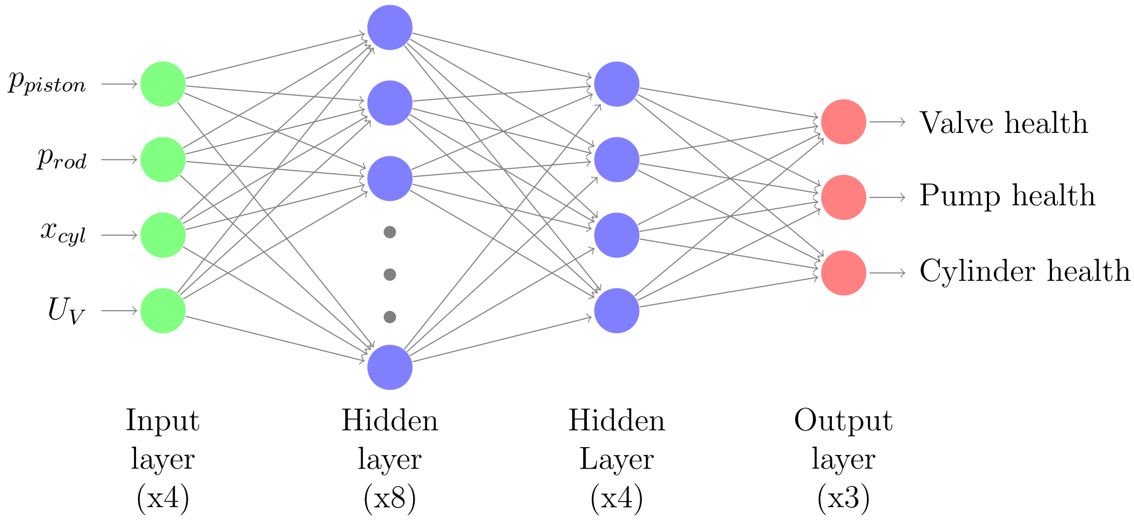


Figure 3.13: Neural network graphical representation - Multi fault

manipulation has been done to enhance the results of the classifier. First of all, since the result of each simulation is an array vector with all the values of each quantity *acquired*, and the neural network works with a single value input, in each simulation, the variance of the quantity is considered for pressures and cylinder position, while the integral is used for the controller command. Moreover, considering the sigmoid transfer function associated to each neuron, as described in the *Artificial Neural Network* section, it resulted more convenient to normalize the input value of the neuron to $[0, 1]$ range: in this way, since the input dynamics of the sigmoid function sweeps the same range of values, the whole scale is used and effectivity is improved. The normalizing value for pressures and cylinder position is given by the variance of the quantity, while for the controller command, the integral is considered, all in healthy condition.

Once the input set is defined, it is possible to address the description of the target set. In order to accurately train the classifier, a target vector is defined according to the levels in Table 3.1. During the training phase, all these quantities are known and given to the algorithm as informations for the input classification. For each training data-set, a target output is built to estimate the health of the considered components.

Input variable	Meaning
p_{piston}	Pressure at piston side of the outer boom cylinder.
p_{rod}	Pressure at rod side of the outer boom cylinder.
x_{cyl}	Position of the outer boom cylinder.
U_V	Input signal to the meter-out valve.

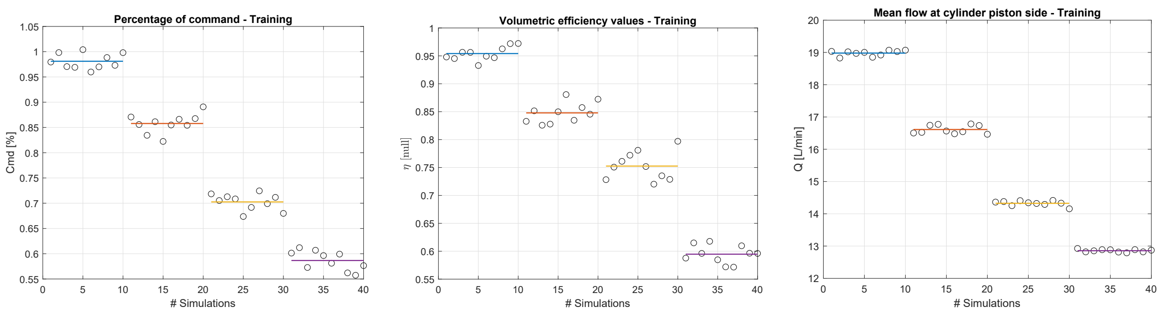
Table 3.5: Neural network input vector.

3.2.2 Training

As broadly described in the *Artificial Neural Network* section, the main goal of the training phase is the tuning of the internal parameters of the network in order to minimize the specified performance index. In details, the values have to be found in the optimization process are the weight matrices of each layer as well as the bias values. Moreover, the objective function to minimize to guarantee performances is the mean squared error (MSE) between the target set and predicted one.

$$\overline{e} = MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{tar_i} - y_{est_i})^2$$

During this training phase, a Bayesian Regularization (BR) based on the Levenberg-Marquadt optimization algorithm is used: a deeper analysis on this training algorithm can be found in the *Artificial Neural Network* section. Furthermore, since an optimization is performed, several attempts were made in order to find the best result: the classifier starts from a different random value every time the algorithm is run, leading to different performances. The training phase starts from the definition of the training data-set that is used to run the simulations and acquire the parameters for the NN. As already explained in the *NN data-set creation* subsection, a white-Gaussian noise is added to the parameters in order to provide a training data-set whose output parameters include also variations that may affect a sensor. In the figure below are shown the most relevant quantities for the training phase: the percentage of command to the meter-in valve, the volumetric efficiency of the pump and the mean flow at the piston side for the cylinder. The results



(a) Valve fault. (b) Pump fault. (c) Cylinder fault.

Figure 3.14: Neural network training data-set - Multi fault.

of the training process are shown in Fig. 3.15. On the horizontal axis is represented the case number, starting from the faulty condition 40 simulation are run for each fault (n.10 for each health level) keeping the other two components in an healthy state. On the vertical axis, both target and training output are plotted and the results are positive: the network has been trained properly according to the target data-set. This result is also given by the minimization algorithm used for the training process, in terms of mean square error (MSE) or root mean square error (RMSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{tar_i} - y_{est_i})^2 = 1.341 \cdot 10^{-5}$$

$$RMSE = \sqrt{MSE} = 0.0037$$

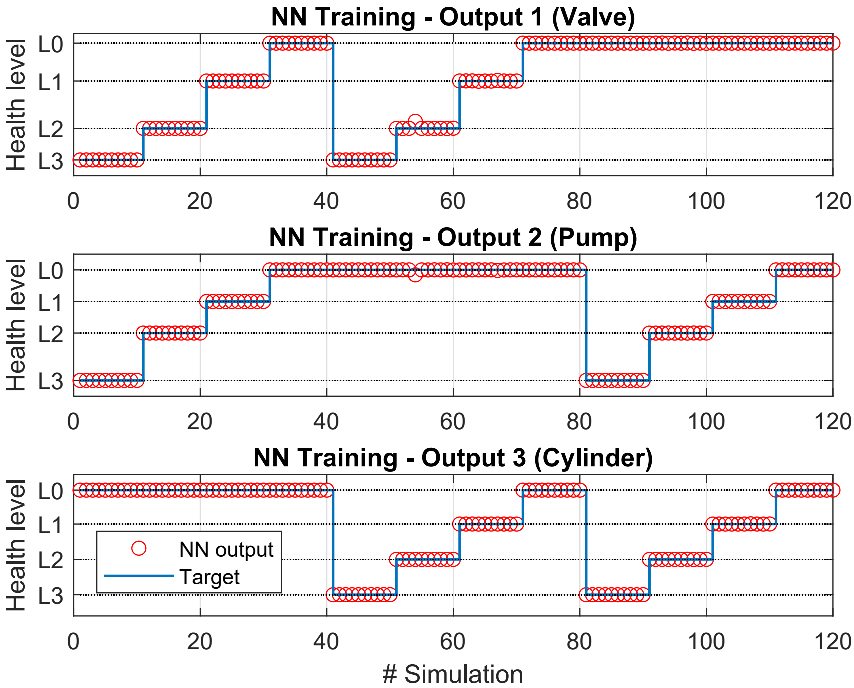
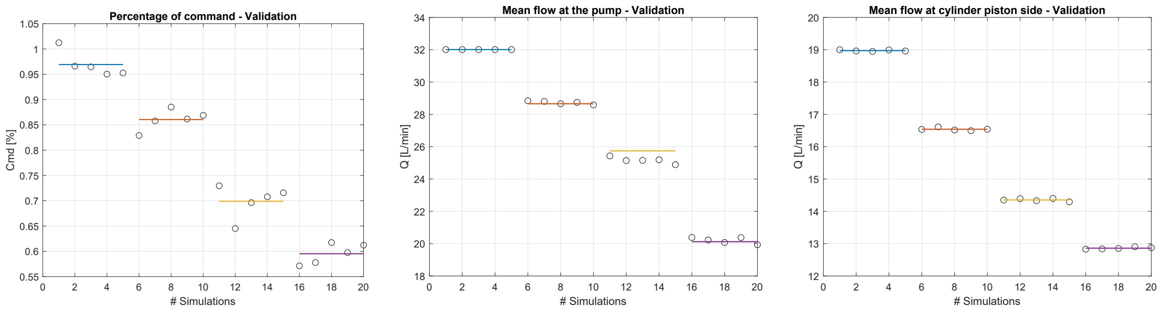


Figure 3.15: Neural network training - Multi fault.

3.2.3 Validation and results

Once exploited the offline training phase, it's necessary to validate its effectiveness by providing a different set of data. As a consequence, in this phase, other fault evolutions for the three components have to be generated, then run the circuit to *acquire* the monitoring parameters to feed the network and verify the output. The idea is equivalent to the definition of the training data-set: adding white-Gaussian noise to the varying parameters, a more realistic scenario is simulated.



(a) Valve fault. (b) Pump fault. (c) Cylinder fault.

Figure 3.16: Neural network validation data-set.

In Fig. 3.16 are shown the most relevant quantities in the validation process. In the same way as for the training phase, the most important quantity for the valve fault is the percentage of command provided to the meter-in valve representing the progressive blockage of the spool; for the cylinder fault, the mean flow at the piston side is considered, based on the assumption that some of the flow provided by the pump is diverted to the rod side and so a loss of useful flow is experienced. Since the pump fault is simulated according to the real test (Fig. 2.1a), the mean flow at the outlet of the pump is considered for the validation as most relevant quantity, since cavitation leads to a reduction of efficiency and flow. The model is run with the new parameters defined for the validation phase and the data acquired from the model are used fed to the previously trained network. In particular, as explained in the *Neural network design* section, the normalized variance of each quantity over the healthy case is performed across the opening time of the meter-in valve, that is to say considering the quantities between second one to six, referring to the input signal in Fig 3.3. Applying the validation data-set defined above to the previously

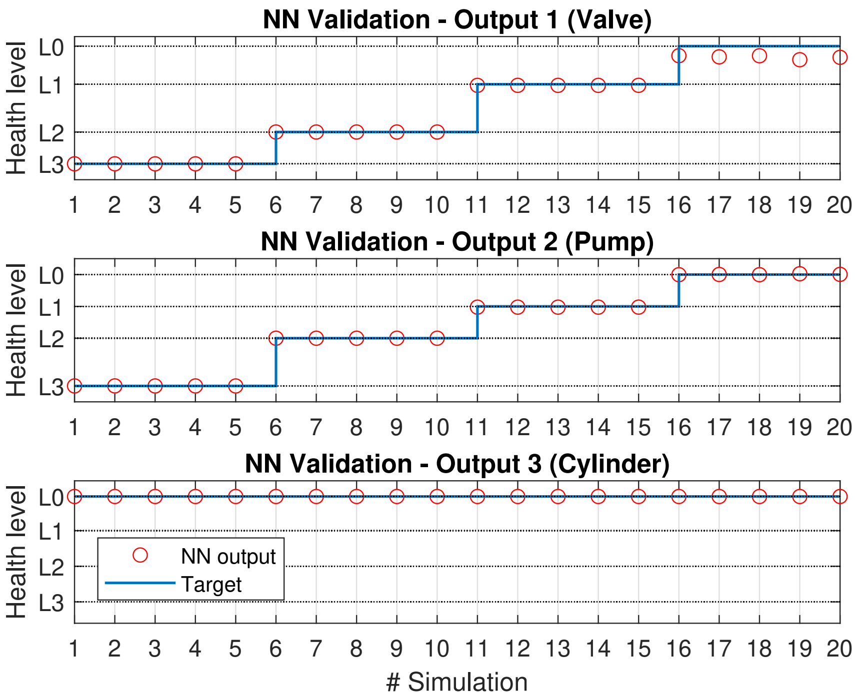


Figure 3.17: Neural network validation - Valve and pump fault.

trained neural network, it's possible to check if the current approach is working: as can be seen in Fig. 3.17, the validation results for the valve and pump fault are really encouraging: based on the defined target-set, the neural network can actually follow the target, even if the input data-set is different from the one used for the training process. This means that the current approach is validated on this new set of data and eventually tested on the reference machine.

During the development of the current approach, the component that gave more trouble to let the algorithm converge to the target set was the pump when combined to others components. One of the reasons is identified in how the validation data-set is generated. Since, as broadly explained in the *Failure modes* section, an orifice upstream the pump is used to reproduce the flow loss, the correct opening area for each of the defined fault levels had to be found in order replicate the same flow loss experienced during the training, done with the variation of the volumetric efficiency parameter in the numerical model. In the latter indeed lies the reason of the troubles undergone by the algorithm since the opening area range of the orifice results to be really small and so the input data-set, during some iterations of the algorithm, couldn't train properly the network, leading to bad results. Eventually, repeating the algorithm and starting from different initial conditions, good results have been reached, as shown in Fig 3.18, along with the cylinder fault. The result plot for the cylinder actuator fault is not shown, but the results are comparable to Fig 3.17 and 3.18: of course, the valve and pump outputs are at the healthy level, while the cylinder output follows the stair shape of the other faults. Analyzing deeply the above

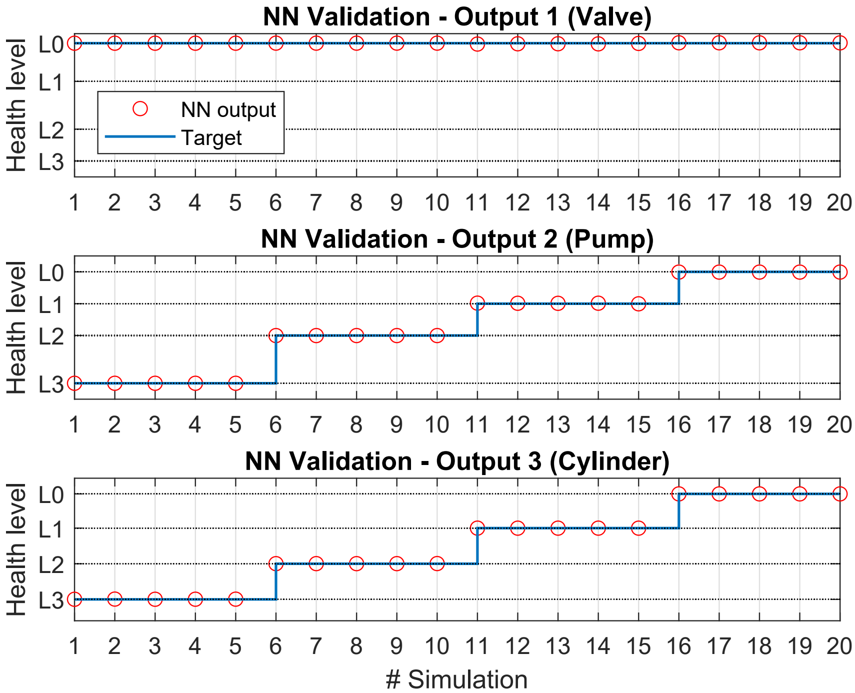


Figure 3.18: Neural network validation - Pump and cylinder fault.

plot using numbers, as done for the training phase, the same error can be computed for the neural network for each kind of validation data-set, as displayed in Table 3.6. As can be seen, the neural network fed with pump fault data is the one with highest error, but the result is still acceptable since the trend is well followed. component.

Faulty component	MSE	RMSE
Valve and pump	$1.0207 \cdot 10^{-4}$	0.0101
Valve and cylinder	0.0331	0.1819
Pump and cylinder	0.0330	0.1817

Table 3.6: Neural network validation performances - Multi fault.

Chapter 4

Prognostic algorithm

To enable the benefits of a truly condition-based maintenance philosophy, real predictive prognostic capabilities are required. Such capabilities are designed to provide maintenance staff with prior notice of pending equipment failure and provide sufficient time to schedule a replacement, thus minimizing both downtime and costs. Real predictive prognostics is understood to be the generation of long-term predictions, describing the evolution of an indicator, for the purpose of estimating the remaining useful life (RUL) of a failing system or component [3]. The primary difficulty encountered in the development of prognostic technologies is the significant uncertainty associated with the generation of long-term predictions of equipment health, and to do so, statistical and historical data are taken into account. The remaining useful life (RUL) is so defined:

$$RUL = 1 - P_i = \left(1 - \frac{t_i}{T_f}\right) \cdot 100 \quad [\%]$$

In this formulation P_i represents the percentage of life related to the i-th inspection point normalized to 1, while t_i represents the age in the same time instant; T_f stands for the failure time of the component, that is the age expressed in number of working hours when the breaking occurs for the analyzed unit. In this chapter, the prognostic approach is defined and developed based on the previously studied case study. In particular, the isolated fault scenario is carried out, in which just one component fails at a certain time.

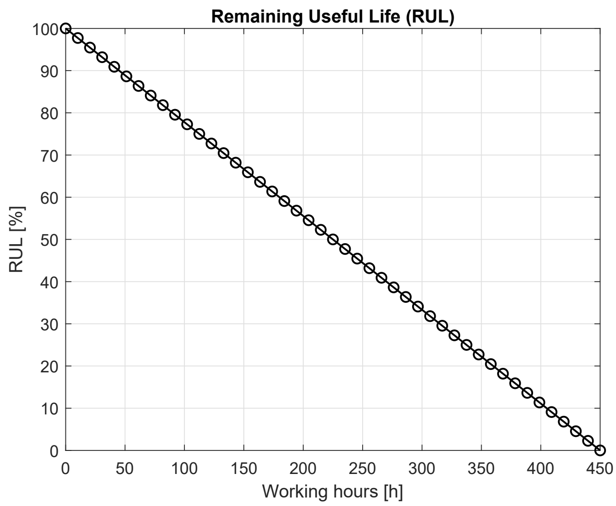


Figure 4.1: Remaining useful life over working hours.

The choice is made due to the fact that, for the prognostic algorithm, the goal of this dissertation is a simple validation of the approach.

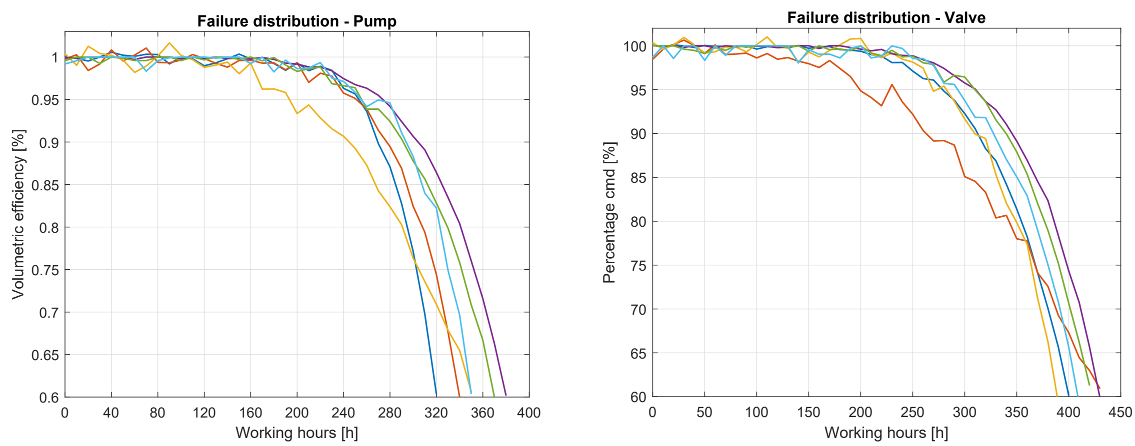
4.1 Single Fault

As already stated in the *Failure mode* section, the faults that have been considered in this research are the blockage of the spool of the valve and the decrease of the volumetric efficiency of the pump, along with the reduction of velocity of the actuator. Based on the fact that pump and valve failure distribution for the considered faults are easier to find, this algorithm is extended just to the meter-in valve and the fixed-displacement pump.In particular, two different neural networks are designed and validated. The reason the choice is that with this configuration, each of the two analyzed units has its own supervisor that can provide a better estimation of the remaining useful life [17].

The most important assumption before getting into the details of the algorithm is the definition of the failure distribution used for the degradation of the components. Based on what said in the *Background* chapter, a reasonable choice for the failure distribution is the Weibull function, or in another formulation, the *universal failure rate function*

$$\hat{z}(t) = Y + K\frac{\beta}{\alpha^{\beta}}t^{\beta-1}$$

Changing the scale (K) and bias (Y) parameters, all the possible failure distributions based on the Weibull function can be modelled. Based on this assumption, it is possible to have a graphical representation of the evolution of different faults over time, both for pump and valve case. Looking at the two plots in Fig. 4.2, it results immediately clear that more then one scenario is considered for the components: since there's not a predefined schedule for a component to fail, some situations near the considered failing time are considered. Furthermore, the used data-sets represent a simulation of an accelerated test, since the durations of the life of the two components are not compliant with the ones of a real unit. This is due the fact that there's no actual advantage to consider the real life duration of the components, since the model run for the simulation does not include the ageing of the system, and so just the parameters value can give interesting results. Additive noise is considered to provide a different situation at each iteration.



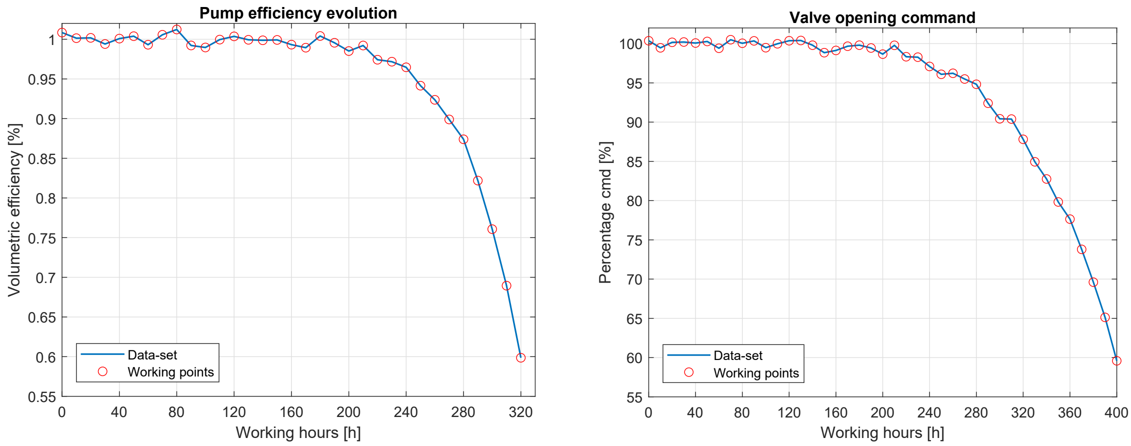
(a) Volumetric efficiency distribution.

(b) Valve opening command.

Figure 4.2: Failure distribution of the components over time.

Simulation set-up

As done for the diagnostic algorithm, once defined the value of the varying parameters, the simulations are run to acquire all the informations about the faulty scenarios to train and validate the neural network and, eventually, perform the RUL estimation. Anyway, it is not possible or feasible to make the fault index vary continuously within a single unique simulation: since each simulation represents a generic extension of the outer-boom actuator, a varying index would mean that the life of the component is degrading in one simple case study. As a result, the only possible solution consists in defining several working points, imposing a faulty level for each of these and, finally, in the acquisition of the related data. Realizing this kind of simulation scheme, it is like monitoring the system with an imposed sampling time and increasing the fault with a discrete step size. As shown in Fig. 4.3, each red circle represents a different level of fault, in which the system is analyzed, leading to a simulation of the system. In particular, a constant step size to fulfill monitoring operations has been chosen and, as a result, all the points are equally distant from each other. The choice of the step size is related to the considered life of the components: this kind of algorithm is not meant to be estimate the life of a component with high resolution, so a $10h$ advance prediction looks reasonable.



(a) Volumetric efficiency distribution. (b) Valve opening command.

Figure 4.3: Working points used for the simulations.

Since the defect evolutions have been described, it is possible to introduce the actual steps employed in the simulation. Each of the circled point in red, in fact, represents the value of an internal variable of the faulty component within the model in a certain simulation. As already specified, the system model is developed in the AMESim environment and the two considered faults are the loss of volumetric efficiency of the pump and the spool blockage of the valve. As said in the *Failure modes* section, the first condition can be easily implemented in the system by changing the efficiency parameter in the pump component: imposing a progressive variation of such parameter, it is possible to simulate the component across its entire life; moreover, the pump fault can be replicated using the upstream orifice, letting the pump work in cavitation and so performing the flow reduction. The values of the parameter are imposed through Matlab/Simulink, where the AMESim interface-block is located and where global variables can be set between the two softwares. Similar considerations can be carried on for the valve fault: adding a decreasing gain to the signal that controls the opening of the meter-in valve, it is possible to replicate a progressive spool blockage since less useful area is opened. As for the diagnostic algorithm, a fixed input command to the controller to extend the actuator, along with the absence of payload, are considered to ensure repeatability and validate the experiment. The input command is shown in Fig. 4.4. The temperature of 40 °C is considered for all the considered cases.

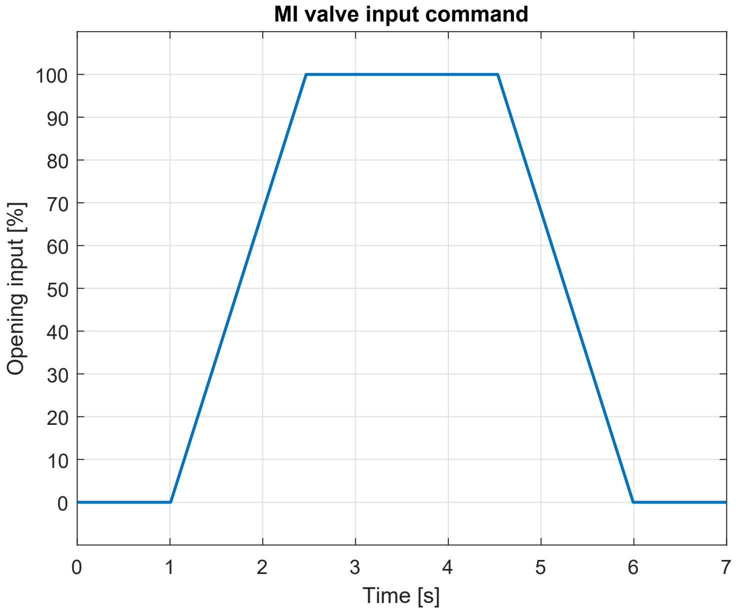


Figure 4.4: Valve input command over time.

4.1.1 Design

Following what explained in the *State of the art* chapter, the design of the neural network for the prognostic algorithm is carried out in this section. For this purpose, the *Neural Network Toolbox* in the MATLAB environment was used. Considering the basic structure, as used for the diagnostic algorithm for the isolated fault, three layers were designed: the input layer, the hidden layer and the output layer. During the first part of the research, a deep analysis on the circuit was done in order to choose the smallest set of monitored parameters to feed the neural network on, following a cost-saving policy.

The results of this analysis show that at least five parameters have to be considered as input of the neural network to provide enough information to the NN to be successfully trained, as shown in Fig. 4.5. The first two considered parameters are the pressures in the chambers of the outer boom actuator. The choice is related to the fact that pressure is a crucial quantity in hydraulic systems. The third parameter is the relative position of the cylinder with respect to the initial position: this quantity has been chosen because since all the considered faults slow down the actuator due to a reduced flow at the piston side and for this reason it's a fundamental parameter. Another easier choice could have been the flow delivered at the actuator: flow rate sensors are expensive and, even if one is present on the outer boom at the rod side of the cylinder, the algorithm is challenged to understand the scenarios with different quantities and cheaper sensors. The fourth monitored parameter is the command input provided to the meter-out valve for the unfolding of the outer boom; such parameter is crucial to include the controller in the diagnostic process since the command is provided by the controller itself. In addition to the parameters mentioned above it is possible to consider also another fundamental variable which is necessary to provide the network central informations about the life of the components, that is to say the age from the installation time. As a result, the overall number of input elements for the neural network is five. This number, anyway, has to be doubled because the informations related to the previous values in previous time instant are crucial to understand the the evolution of the fault. So the real input-set is constituted by ten components, as shown in the scheme below. The ten defined input, anyway, have to be manipulated before being provided to the network, in order to enhance the results of the classifier. In details, the mean value of the quantity is considered over the time the input signal is applied. Furthermore, considering the structure of the transfer function associated to each neuron, as described in *Artificial neural network* section, it is convenient to normalize the input value in order to impose a working range between zero and one. To do so, the mean value of the variable is computed considering the component as completely healthy: in other words the neural network input is normalized to the value at nominal conditions.

$$\bar{x} = E[x] = \frac{1}{n}\sum_{i=0}^{n} x_i = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{n}$$

Input variable	Meaning
t_i	Age of the component at current inspection time.
t_{i-1}	Age of the component at previous inspection time.
p_{piston_i}	Pressure at piston side at current inspection time.
$p_{piston_{i-1}}$	Pressure at piston side at previous inspection time.
p_{rod_i}	Pressure at rod side at current inspection time.
$p_{rod_{i-1}}$	Pressure at rod side at previous inspection time.
x_{cyl_i}	Position of the outer boom cylinder at current inspection time.
$x_{cyl_{i-1}}$	Position of the outer boom cylinder at previous inspection time.
$U_{V\,i}$	Input signal to the meter-out valve at current inspection time.
$U_{V\,i-1}$	Input signal to the meter-out valve at previous inspection time.

Table 4.1: Neural network input vector.

Once the input set is defined, it is possible to address the description of the target set. In order to accurately train the classifier, a target vector is defined according to the plot in Fig. 4.1. During the training phase, all these quantities are known and given to the algorithm as informations for the input classification. For each training data-set, a target output is built to estimate the health of the considered components. In details, three training-sets for each component are considered, as shown in Table. 4.2.
The shape in the plot above is exactly a straight line with the point on it equally distant,

Pump data-set length [h]	Valve data-set length [h]
320	400
340	430
350	390

Table 4.2: Neural network training data-set.

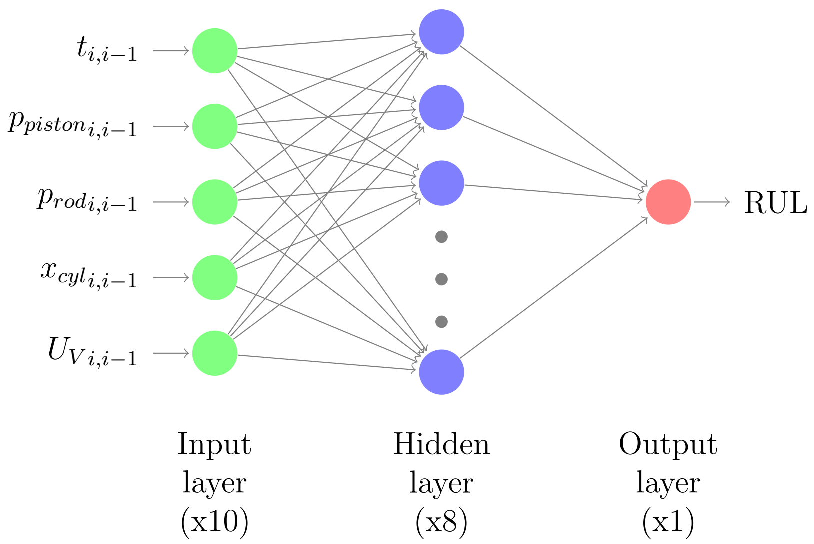


Figure 4.5: Neural network graphical representation.

which as natural consequence of the RUL formulation. By monitoring the system with a constant time step equal to ten hours, the remaining useful life moves linearly from 100% (new) to 0% (complete failure). However, the linear behavior doesn't imply that the components are failing in a linear way: as specified before, in fact, the evolution of the life follows the shape of the Weibull function. The linear behavior depends only on the formulation of the RUL and its true as long as the failure time is known. Basically every discrete working point in Fig. 4.3 is mapped on Fig. 4.1.

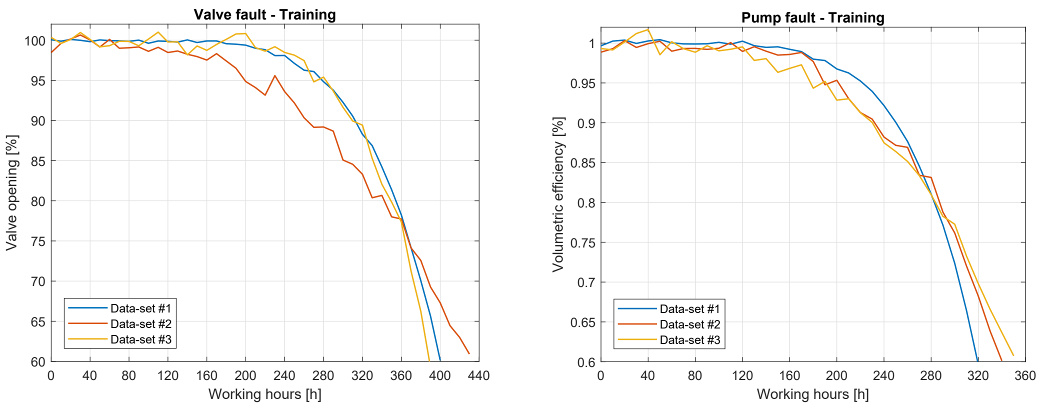
Once input and target data-sets are defined, it's possible to move to design the structure of the NN. A neuron has to be considered for each input vector, so four neurons are considered for the input layer. There is no general rule to choose the number of neurons of the hidden layer, therefore a cross-validation approach is used: after a certain number of added hidden neurons, the solver start over fitting the data and give bad estimates on the test set. The result of this analysis showed that eight neurons in the hidden layer provide the best performances. In details, the values that are modified during the optimization process are the weight matrices related to each layer. The output layer is given by three neurons since the goal of this research is to monitor the faults in the meter-in valve, pump and cylinder: the output is a number in the range [0,1] according to the normalization of the inputs. A graphical representation of the designed NN is shown in Fig. 4.5. Therefore, obtained the monitoring parameters for all the defined conditions and normalizing them as stated, a sufficient set of input data is available and a suitable neural network can be trained. During the design phase of the network several parameters can be customized in order to obtain the best results related to the analyses case, as mentioned in the previous subsection. Apart from the number of neurons, which is imposed equal to eight in this case, the starting weights and biases of each neuron are set up as randomly chosen at the beginning of the optimization procedure. Similarly, the amount of data that are assigned to each group during the training phase are split casually: even if the percentage of data assigned to training, validation and test set is well specified, the input/output pairs that will belong to each of them are randomly chosen. Moreover, promising results have been obtained by assigning 70% of data to training class and splitting the remaining part 15% in validation and 15% in test group.

4.1.2 Training

As broadly described in the *Artificial Neural Network* section, the main goal of the training phase is the tuning of the internal parameters of the network in order to minimize the specified performance index. In details, the values have to be found in the optimization process are the weight matrices of each layer as well as the bias values. Moreover, the objective function to minimize to guarantee performances is the mean squared error (MSE) between the target set and predicted one.

$$\bar{e} = MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{tar_i} - y_{est_i})^2$$

During this training phase, a Bayesian Regularization (BR) based on the Levenberg-Marquadt optimization algorithm is used: a deeper analysis on this training algorithm can be found in the *Artificial Neural Network* section. Furthermore, since an optimization is performed, several attempts were made in order to find the best result: the classifier starts from a different random value every time the algorithm is run, leading to different performances. The training phase starts from the definition of the training data-set that is used to run the simulations and acquire the parameters for the NN. As already explained in the *NN data-set creation* subsection, a white-Gaussian noise is added to the parameters in order to provide a training data-set whose output parameters include also variations that may affect a sensor. In the figure below are shown the most relevant quantities for the training phase: the meter-in valve area opening and the volumetric efficiency of the pump, according to Table 4.2. The shape of the training data-sets, as can be seen, recalls the Weibull function distribution and the P-F curve.



(a) Valve fault.

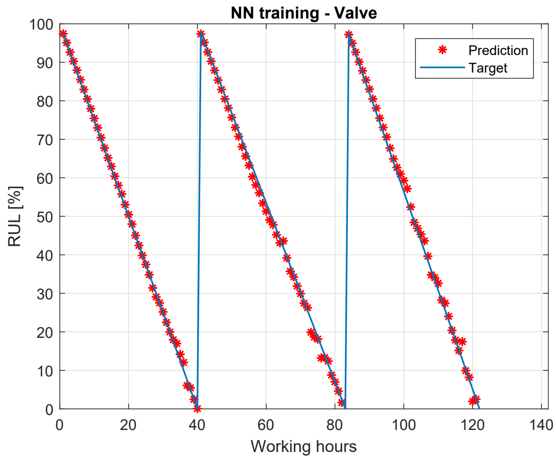
(b) Pump fault.

Figure 4.6: Neural network training.

The results of the training process are shown in Fig. 4.7. On the horizontal axis, the working hours are represented, while on the vertical axis, both target and training output are plotted and the results are positive: the network has been trained properly according to the target data-sets. This result is also given by the minimization algorithm used for the training process, in terms of mean square error (MSE) or root mean square error (RMSE). Recalling what said for diagnostic algorithm results, also for the prognostic one the pump gives more problems to the classifier, leading to a higher error.

Faulty component	MSE	RMSE
Pump	$6.7008 \cdot 10^{-4}$	0.0259
Valve	$1.6062 \cdot 10^{-4}$	0.0127

Table 4.3: Neural network training data-set.

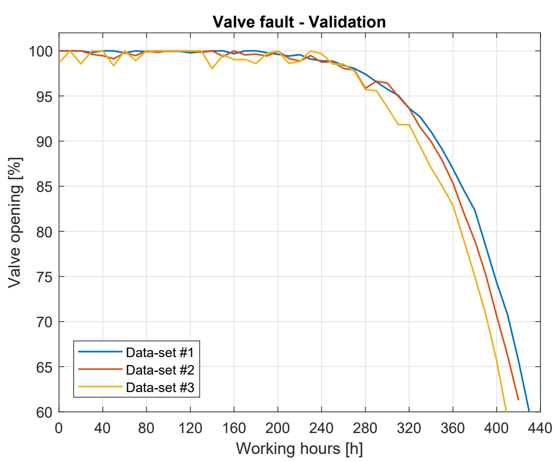


(a) Valve fault. (b) Pump fault.

Figure 4.7: Neural network training.

4.1.3 Validation and results

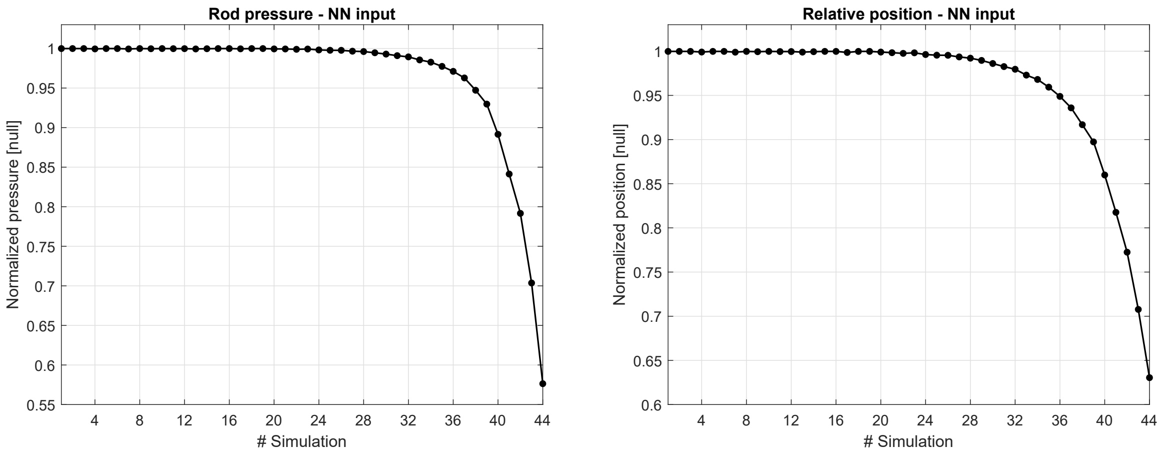
Once exploited the offline training phase, it's necessary to validate its effectiveness by providing a different set of data. As a consequence, in this phase, other fault evolutions for the three components have to be generated, then run the circuit to *acquire* the monitoring parameters to feed the network and verify the output. The idea is equivalent to the definition of the training data-set: adding white-Gaussian noise to the varying parameters, a more realistic scenario is simulated.
In Fig. 4.8 are shown the most relevant quantities in the validation process. In the same way as for the training phase, the most important quantity for the valve fault is the per-



(a) Valve fault. (b) Pump fault.

Figure 4.8: Neural network validation data-set.

centage of command provided to the meter-in valve representing the progressive blockage of the spool; the pump fault is simulated according to the real test (Fig. 2.1a), the mean flow at the cylinder is considered for the validation as most relevant quantity, since cavitation leads to a reduction of efficiency and flow. The model is run with the new parameters defined for the validation phase and the data acquired from the model are used fed to the previously trained network. In particular, as explained in the *Design* section, the mean value of each quantity normalized over the healthy case is performed across the opening time of the meter-in valve, that is to say considering the quantities between second one to six, referring to the input signal in Fig 4.4. The actual shapes of the pressures inside the chambers of the actuator and the cylinder position fed to the network as input are showed in Fig. 4.9: each of the depicted point represents the normalized mean value computed within a single simulation; later on, in order to test the network, each of this discrete values have been interpolated to get the overall behavior of the parameter. Considering the figures below, it's easy to understand that also the parameters fed to the network follow the Weibull function shape.



(a) Rod side normalized pressure.

(b) Cylinder relative normalized position.

Figure 4.9: Neural network validation input data-set.

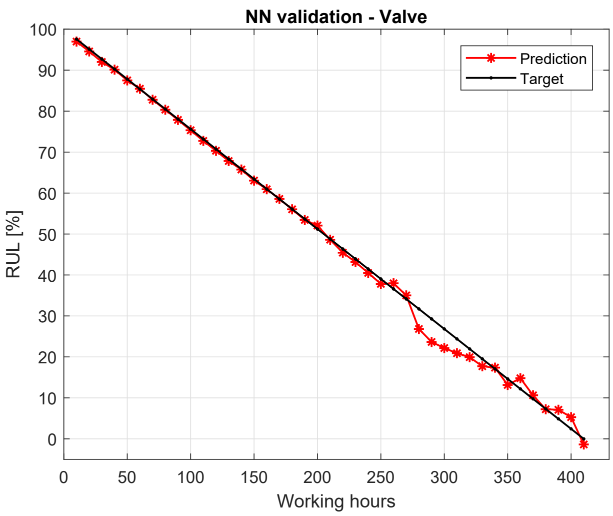


Figure 4.10: Neural network validation - Valve fault.

Applying the validation data-set defined above to the previously trained neural network, it's possible to check if the current approach is working: as can be seen in Fig. 4.10, the validation results for the valve fault are really encouraging: based on the defined target-set, the neural network can actually follow the target, even if the input data-set is different from the one used for the training process. This means that the current approach is validated on this new set of data and can be eventually tested on the reference machine. During the development of the current approach, the component that gave more trouble to let the algorithm converge to the target set was the pump. One of the reasons is identified in how the validation data-set is generated. Since, as broadly explained in the *Failure modes* section, an orifice upstream the pump is used to reproduce the flow loss, the correct opening area for each of the defined fault levels had to be found in order replicate the same flow loss experienced during the training, done with the variation of the volumetric efficiency parameter in the numerical model. In the latter indeed lies the reason of the troubles undergone by the algorithm since the opening area range of the orifice results to be really small and so the input data-set, during some iterations of the algorithm, couldn't train properly the network, leading to bad results. Eventually, repeating the algorithm and starting from different initial conditions, good results have been reached, as shown in Fig 4.11. Analyzing deeply the above plot using numbers, as

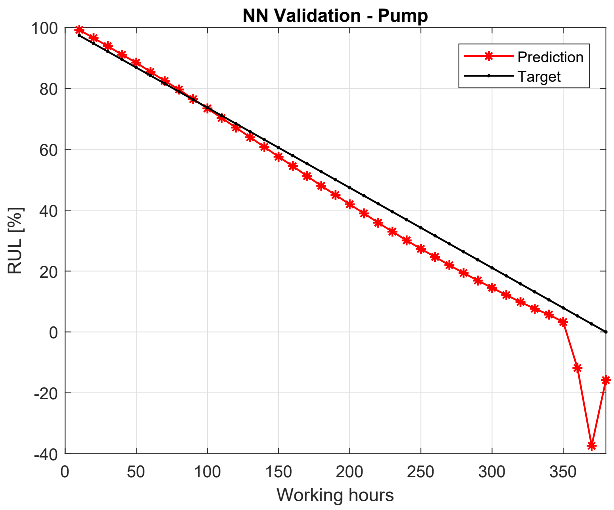


Figure 4.11: Neural network validation - Pump fault.

done for the training phase, the same error can be computed for the neural network for each kind of validation data-set, as displayed in Table 4.4. As can be seen, the neural network fed with pump fault data is the one with highest error, but the result is still acceptable since the trend is well followed.

Faulty component	MSE	RMSE
Valve	$3.345 \cdot 10^3$	57.8327
Pump	$3.908 \cdot 10^3$	62.5144

Table 4.4: Neural network validation performances.

Chapter 5

Experimental setup

After the description of the operations performed in simulated environment and all the various achieved results, it is possible to address the implementation on the real machine. The main purpose of this chapter is the description of the preparatory phase related to the actual experimental tests, whose aim is the validation of the proposed approach. Unlikely, in this chapter, some results are shown regarding the response of the system but due to some differences between the model and the actual machine, the algorithm has not been tested. The goal of this chapter is then to describe the designed acquisition system of the machine along with the response of the system during the case study: as will be explained in the next chapter, tuning more accurately the parameters of the model, the algorithm can be tested and eventually validated.

5.1 Acquisition system

The acquisition system is a fundamental part of the machine since it's not only related to the monitoring of the quantities of the system but also to the controller implementation. The environment used for the monitoring system is the graphical programming language *LabVIEW™* by National Instrument™, leader company in the market for data acquisition solutions and software platforms. In particular, the device used is a NI CompactRIO™(cRio™), that is a real-time embedded controller that combines reconfigurable IO modules along with FPGA and Ethernet interface. Thanks to this system, it is possible to acquire multiple signals coming from sensors and provide output signals to actuate

Figure 5.1: NI cRio™.

Sensor name	Meaning
P1	Pump line pressure
P2	Tank pressure
P7	Main boom rod side pressure
P8	Main boom piston side pressure
P9	Outer boom rod side pressure
P10	Outer boom piston side pressure
P12	LS line pressure
Angle 1	Main boom angle
Angle 2	Outer boom angle

Table 5.1: Main sensor set.

the machine, according to the defined control strategy. The FPGA system is used to map the pins of the data acquisition system to the sensors of the machine: in this phase all the signals are raw and some processing, such as calibration and saturation, has to be done in order to have feasible informations. ¡In details, twelve pressure sensors are mounted on the machine, covering the actuator lines and also the pump, tank and LS ones. Then two angular sensors are mounted to get the position in the space of the actuators and also the relative position from the initial state: through some geometric calculations is possible to get the strokes of both main and outer booms. Based on the design of the input layers of both proposed algorithms, no other information is actually needed to be *sensed* in the system. In fact, the forth variable that is used by the AI is the command provided by the controller, that is computed real-time during the operation of the machine: for this reason, this signal is also related as *virtual sensor*, since it's not a quantity sensed by the acquisition system but a signal produced by the controller itself and fed again as output. The output of the controller, after some processing and conversions, is directly used to actuate the solenoid and so the related valve. The most important sensors mounted on the machine are shown in Table 5.1 A *case* structure (if-else) has been implemented to provide the same signal to each test and to ensure the repeatability, along to be compliant to the simulation environment. As shown in Fig 5.2, since the sampling time is 50ms,

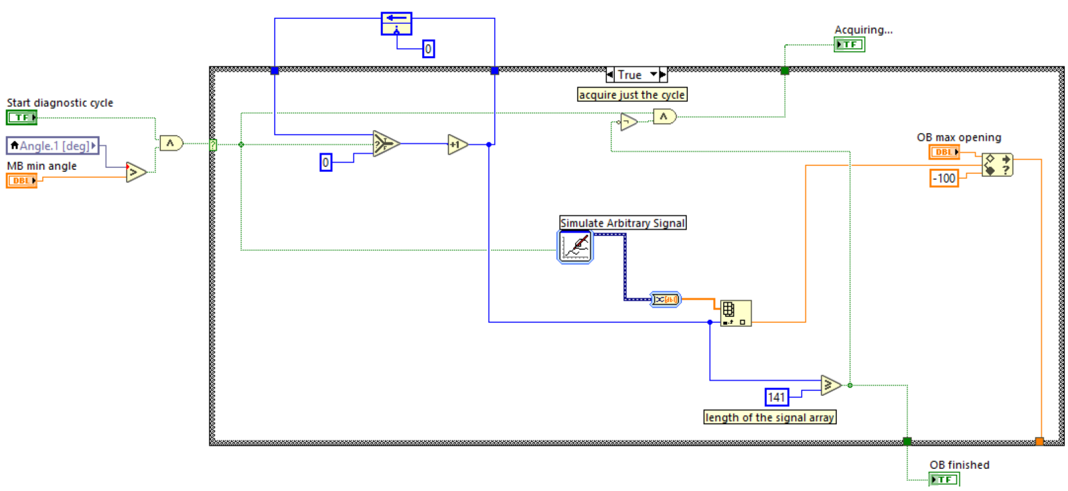


Figure 5.2: Diagnostic cycle definition in LabVIEW.

n.141 points are used to reproduce the defined signal lasting for five seconds, like in the simulation. Also, before the actuation, a check on the opening angle of the main boom is done in order to keep the machine safe: indeed, if by mistake the operator starts the diagnostic cycle when the main boom is not safely opened, no damage is done. Another interesting detail that is worth to mention is the file saving process: as soon as the diagnostic cycle begins, a file is created, opened and the writing of the acquiring data is done till the cycle is not over; this helps for the post-processing phase since only the cycle is considered. Eventually, the signal is saturated in order to provide always a reasonable signal at the valves, preventing them to be damaged by an excessive voltage.

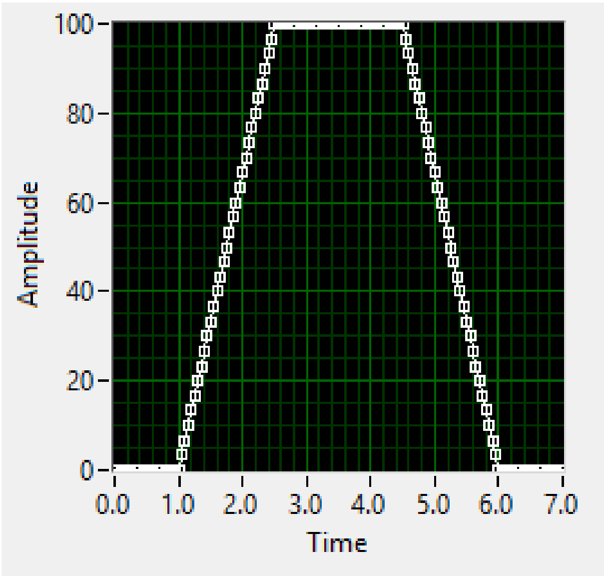


Figure 5.3: Arbitrary input signal.

5.1.1 Controller implementation

As already explained in the *Introduction* chapter, a controller is implemented in the system and on the acquisition system. The reasons behind the controller are that in the considered case study both overrunning and resistive load conditions are experienced. The basic logic for the computation is shown in Fig. 5.4. Two blocks are used for the definition of the output command during overrunning: a PI controller and a logic block for the definition of the command.

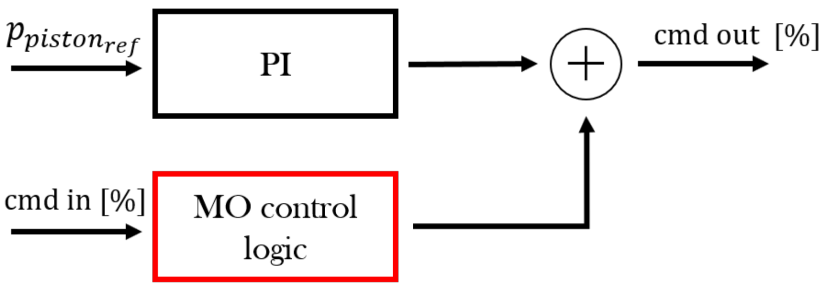
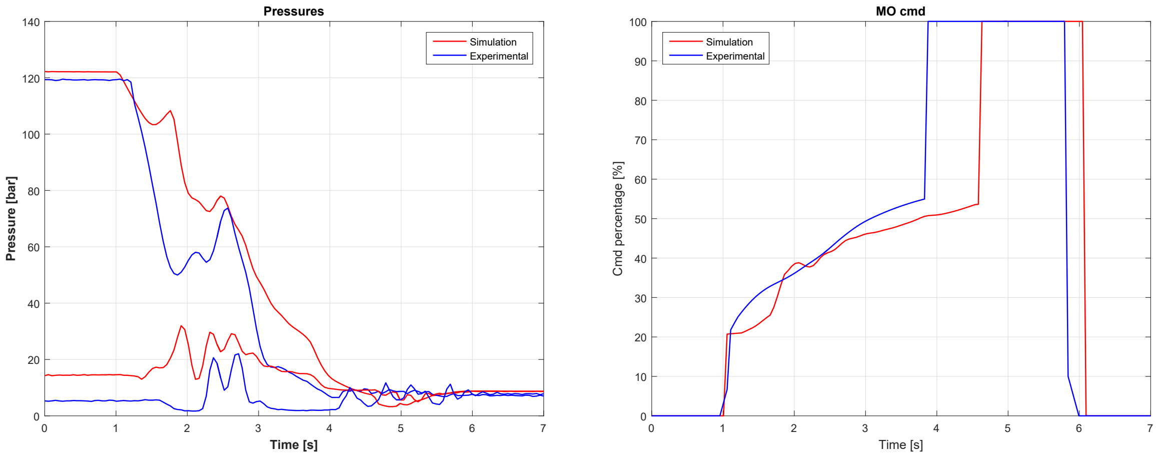


Figure 5.4: Output command computation.

5.2 Crane response

In order to better understand the behavior of the system, a comparison between the model response and the experimental one for the most important quantities is given. As can be seen in Fig. 5.5a, there are some differences between the two responses: in the experimental data, the pressures drop faster, meaning that the meter-out control is provided in a different way with respect to the simulator. Indeed, as shown in Fig 5.5b, even if the shape of the commands matches for the first seconds, the overrunning-resistive threshold is crossed in a different time. As already explained in the *Control strategy*



(a) Piston and rod pressures.

(b) Meter-out command.

Figure 5.5: Machine and model comparison.

subsection, the meter-out command is provided based on the force balance at the outer-boom actuator. In Fig. 5.6, the force balance is plotted: as can be seen here, the force curve taken from the machine lies between the simulated one and the filtered one on the Matlab environment. But this kind of signal is extremely noisy, leading to a problem when setting the threshold for the force sign switching: to solve this, a low-pass filter (LPF) is installed to clean the signal. On the LABView code, the LPF is implemented in a different way with respect to the MATLAB code, changing the shape of the force curve and so a different command to the meter-out valve.

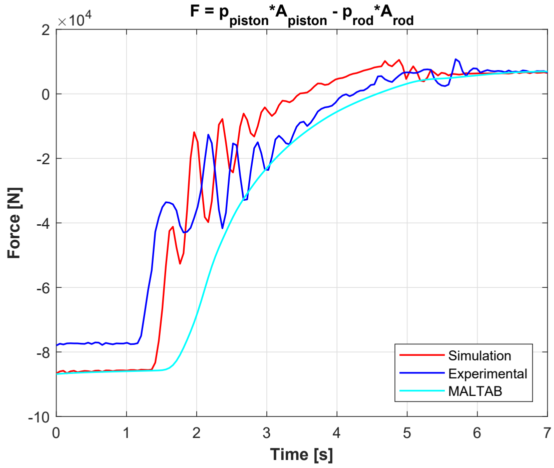
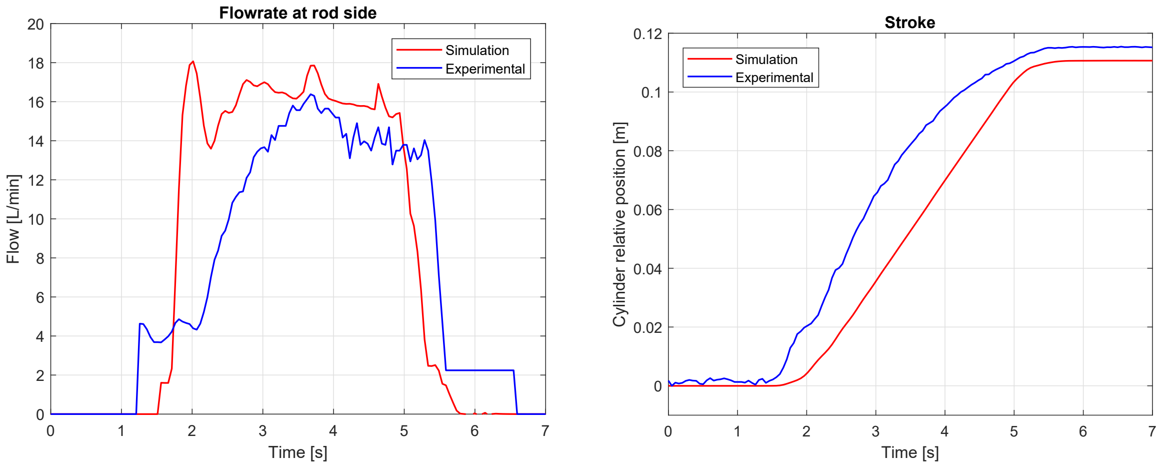


Figure 5.6: Force balance computation.

The last two relevant quantities provided by the experiment are the flow at the rod side and the relative stroke of the actuator, shown in Fig. 5.7. As can be seen, the stroke



(a) Flowrate at rod side. (b) Cylinder relative position.

Figure 5.7: Machine and model comparison.

variable are close to the simulation data, while the is a little bit different. Even if the variables are close enough to believe that the algorithm can work properly, instead these small differences are enough to make it not converging to the right solution. The main problem is that the ranges for these faults are pretty short, giving hard time to the classifier and leading to the wrong solution. A further step would be to increase the training set to new scenarios and trying to better tune the numerical model to match the machine at least during the case study.

Chapter 6

Conclusions and future work

After the entire dissertation has been covered, it is possible to summarize the main concepts explored and try to carry out some considerations. At the beginning of the research, a broad overview about maintenance strategies is performed, giving wide space to the condition-based maintenance method. In particular, the difference between diagnostics and prognostics is given, defining the range of this approaches during the life of a component or a system. Later on, a deep analysis on the reference machine is done to better understand the case study of the research and the relative machine on which the algorithms are based. After this, some studies about the available literature related to health management and fault detection are fulfilled, going from the analysis of the considered failing component within the system, along with the reasons behind the choices made; moreover an explanation on the data-set creation is carried out to understand the difference between training and validation for the developed algorithms. Afterwards, an overview on the artificial neural network structure and advantages is done to provide nomenclature and motivations for the developing of the algorithms. Although the original idea consisted in the employment of an ANN trained on the information coming from actual experiments, the lack of available run-to-failure data set related to hydraulic units has led to the necessity to employ a detailed model to obtain the necessary information. Concluding the theoretical background, the Weibull functions is explained along with all the derivations that are broadly used in the literature, helping the reader to understand how the components are considered failing in time for the prognostics algorithm.

Later on, once all the needed background has been over viewed, the diagnostic algorithm has been designed and then applied to the hydraulic model of the reference machine: the obtained results demonstrate the robustness and the strength of the proposed methodology. During this phase, moreover, some considerations about the optimal set of acquired data have been performed, defining the most suitable group of input that not only can ensure high performances in terms of prediction, but results also to be feasible for its implementation on a real system. Eventually, both isolated and multiple faults situations are considered, challenging the algorithm on a more difficult detection scenario.

Afterward, the prognostic algorithm is exploited, explaining in details the design and the choices behind that for the estimation of the Remaining Useful Life (RUL): the obtained results are promising and demonstrate that this kind of approach results really powerful for this type of prediction. During this part, some observations and assumptions are made on the way the considered components fail during time: the choice of the Weibull function for the model of the fault of the system is reasonable since a lot of components follow that path to failure.

Eventually, a validation experiment has been set up in such way that the considered condition was reproducible and the post-processing analysis was easy. In particular, the case study is safely reproduced on the acquisition and control system, designing a structure that allows the saving of all the necessary sensor data and ensuring to be compliant with the training case on the simulator.

After the conclusion of the research, some reasoning about the of the diagnostic and prognostic method can be performed. In particular, the results for the diagnostic algorithm are interesting since with a small number of inputs and a small network, great results are achieved and no multiple interacting systems is required. The multi fault analysis results to be really interesting being the actual application a combination of components that interact between each other. A temperature analysis would be a good improvement on the developed algorithm, adding new situations and helping the user to get the best performances out of this tool.

Further steps can be done for the prognostic algorithm, trying to include a classifier for the cylinder fault and including the temperature variable, including other scenarios and allowing the predicting algorithm to be as general and reliable as possible. Eventually, with a deeper analysis on the components and a more detailed classifier can be trained, complicating the structure adding a fuzzy logic system that combines the networks providing good and clean results. This kind of approach can be regarded as the best one to solve all kind of discrepancies between actual and estimated curves, keeping in mind that the network can learn how to manage new kind of situations through the training experience. Although it represents the best possible solution, however, it is not yet feasible in reality, considering the complete lack of monitoring data coming from a real system where the faults are advancing. As specified, the final goal of the project is the implementation a diagnostic and prognostic estimator, based on real data coming from run-to-failure tests performed on the actual components. Based on this premises, it appears clear that a a possible improvement can been implemented, trying to enhance the performances introducing a fuzzy logic system.

In conclusion, the results obtained are really satisfactory and demonstrate the effectiveness of the proposed methodology. Looking at the research, however, it's clear that further steps have to be done on this pattern, especially on the validation of the current approach on the machine: to do so, the current model has to be finely adjusted in order to match more the actual system and provide results that could not be discussed in this dissertation. Eventually, different kind of faults can be added and different types of external influencing variables can be taken into account, to further challenge the capabilities of the proposed method. Although all these considerations, the designed and implemented prognostic strategy has revealed many of the initially hidden potentiality, demonstrating flexibility and robustness in all the handled situations.

Bibliography

[1] Andrew K.S. Jardine, Daming Lin, and Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510, 2006.

[2] John Watton. *Modelling, monitoring and diagnostic techniques for fluid power systems.* Springer-Verlag London Limited, Cardiff, UK, 2007.

[3] Shane Butler. *Prognostic Algorithms for Condition Monitoring and Remaining Useful Life Estimation.* PhD thesis, National University of Ireland, Maynooth, 2012.

[4] Mohammed Ben-Daya, Uday Kumar, and D.N. Prabhakar Murthy. *Condition-Based Maintenance*, pages 23–42. 2016.

[5] Janos Gertler. Fault detection and diagnosis. *Encyclopedia of Systems and Control*, 2014.

[6] Diego A. Tobon-Mejia, Kamal Medjaher, and Noureddine Zerhouni. The iso 13381-1 standard's failure prognostics process through an example. *Prognostics and System Health Management Conference, Macau*, pages 1–12.

[7] R. Bianchi, A. Vacca, and F. Campanini. Combining control and monitoring in mobile machines: the case of an hydraulic crane. *The 11th International Fluid Power Conference, 11. IFK, Aachen, Germany*, 2018.

[8] Martin T. Hagan, Howard B. Demuth, and Mark H. Beale. *Neural network design.* Martin Hagan, 2014.

[9] Krzysztof Patan. *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes.* Springer-Verlag Berlin Heidelberg, 2008.

[10] W. McCulloch and W. Pitts. *A logical calculus of ideas immanent in nervous activity.* Bulletin of Mathematical Biophysics, 1943.

[11] Laurene V. Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications.* Prentice-Hall, 1994.

[12] Frank Burden and Dave Winkler. *Bayesian Regularization of Neural Networks*, pages 355–387. Humana Press, 2009.

[13] Anselm Lorenzoni and Michael Kempf. Degradation processes modelled with dynamic bayesian networks. *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 1694–1699.

[14] Wayne B. Nelson. *Accelerated testing: statistical models, test plans and data analysis*, volume 344.

[15] Weibull.com. Characteristics of the weibull distribution, urldate: 2018-11.

[16] John Moubray. *Reliability-centred Maintenance*. Industrial Press Inc., 1997.

[17] Yuri Ghini and Andrea Vacca. A method to perform prognostics in electro-hydraulic machines: the case of an independent metering controlled hydraulic crane. *Int. J. of Hydromechatronics, 2018*, 1(2).