N° Ordre……../FHC/UMBB/2024

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA-BOUMERDES

Faculté des Hydrocarbures et de la Chimie

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Présenté par

BENHACINE Souheil Nadjmeddine
Et
NAILI Mohamed Chakib

Filière : Hydrocarbures
Spécialité : Instrumentation dans l'industrie pétrochimique

Thème

Intelligent calibration of level sensors using functional link neural networks with piecewise linear interpolation

Devant le jury :

CHAIB	Ahmed	Prof	UMBB	Président
HARFOUCHI	Fatima	MCB	USTHB	Examinateur
DOGHMANE	M. Zinelabidine	Attaché de Recherche	Sonatrach	Examinateur
HABBI	Hacene	Prof	UMBB	Encadrant

Année Universitaire : 2023/2024

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA-BOUMERDES

Faculté des Hydrocarbures et de la Chimie
Département : Automatisation des procédés et électrification
Filière : Hydrocarbures
Spécialité : Instrumentation dans l'industrie pétrochimique

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Thème

Intelligent calibration of level sensors using functional link neural networks with piecewise linear interpolation

Présenté par :
BENHACINE Souheil Nadjmeddine
NAILI Mohamed Chakib

Avis favorable de l'encadrant :
Nom et prénom : ……
signature :

Avis favorable du Président du jury
Nom et Prénom : ……
Signature :

Cachet et signature

First and foremost, we thank "ALLAH" the Almighty, for the health, willpower, and mercy He has granted us throughout these long years, and for giving us the strength to complete this work.

Our thanks go to our supervisor, Mr. HABBI Hacen. Thank you for your guidance throughout the development of our thesis. Your expertise, availability, and invaluable support have been essential in the realization of our project.

We thank our families: our parents, our siblings, and our friends for their love, irreplaceable and unconditional support.

We would also like to thank all the teachers in the Department of Automatisation et electrification des procedes industriels and Equipment at the Faculty of Hydrocarbons and Chemistry / University of Boumerdès.

We extend our gratitude to the engineers of DP Hassi-Rmel from Sonatrach who guided us step by step in handling the project.

Content

Content

List of figures VII
List of tables IX
List of Abreviations X
Abstract XI
Introduction 1
Chapter.I. Measurement systems and calibration 2
I.1. Introduction 3
I.2. Metrology and related terms 3
I.3. Sensors and transducers 4
I.4. Measurement Chain 5
I.5. Sensor Characteristics 6
I.6. Sensors classification 10
I.7. Non-linearity of sensors 13
I.7.1. Linearization of a sensor's characteristic 14
I.7.2. Sensor linearization methods 15
I.7.2.1. Classical methods 15
I.7.2.2. Calibration methods 15
I.8. Conclusion 19
Chapter.II. Functional Link Artificial Neural Networks 21
II.1. Introduction 22
II.2. Artificial Neural Networks 22
II.2.1. Recurrent Neural Networks (RNNs) 23
II.2.2. Feedforward Neural Networks 24
II.3. Neural Network Learning 24
II.3.1. Unsupervised Learning 25
II.3.2. Supervised Learning 25
II.4. Functional Link Artificial Neural Network 26
II.4.1. Implementation of the FLANN neural network 26
II.4.2. FLANN types 27
II.5. Training algorithms 29
II.5.1. Gradient Descent Algorithm 29

II.5.2. Artificial bee colony (ABC) algorithm 29
II.6. FLANN-based LVDT non-linearity compensation 30
II.6.1. Interpretation of Results 34
II.7. Conclusion 34
Chapter.III. FLANN-based level sensors calibration and implementation 36
III.1. Introduction 37
III.2. Experimental setup presentation 37
III.2.1. Three-Tank-System 38
III.2.1.1. Level sensor overview 39
III.2.1.2. Simulink calibration platform 39
III.2.2. Data collection 41
III.2.3. Intelligent calibration system design 47
III.2.3.1. FLANN model 48
III.3. Simulation results and discussion 50
III.3.1. Gradient-based training of the model 50
III.3.2. FLANN model with piecewise linear interpolation 54
III.3.3. Gradient-based training of the model with PWL interpolation 55
III.4. Real-time implementation 58
III.4.1. FLANN model without piecewise interpolation results 59
III.4.2. FLANN model with piecewise interpolation results 61
III.4.3. Performance evaluation and discussion 63
III.5. Conclusion 63
Conclusion 64
Bibliography 66

List of figures

Figure I-1 Block diagram of a sensor 4
Figure I-2: Typical Diagram of a Measurement Chain 5
Figure I-3 Sensitivity of measurement 7
Figure I-4: Calibration curve, Transfer function, or Output curve = f(Input) 8
Figure I-5: Simplified scheme about measurement errors 9
Figure I-6 Hysteresis phenomenon 9
Figure I-7:Examples of the application of physical effects in the realization of active sensors: (a) thermoelectricity, (b) pyroelectricity, (c) piezoelectricity, (d) electromagnetic induction, (e) photoelectricity, (f) Hall effect. 13
Figure I-8:Look-up table calibration method 16
Figure I-9: Piecewise linear calibration method 17
Figure I-10: Sensor characteristic linearization 18
Figure I-11: Inverse transfer function method for linearizing calibration. 19
Figure II-1: Diagram of a Feedforward Neural Network (Static) 22
Figure II-2 : Diagram of a Recurrent Neural Network (Dynamic) 23
Figure II-3: Recurrent Neural Network 23
Figure II-4:Illustrative Example of a Multilayer Perceptron (MLP) 24
Figure II-5:Supervised Learning of an Artificial Neural Network (ANN) 25
Figure II-6:The structure of a FLANN 27
Figure II-7: Trigonometric expansion 27
Figure II-8:Chebyshev expansion 28
Figure II-9:Legendre Structure 28
Figure II-10:Scheme of a non-linearity compensator of LVDT 31
Figure II-11:Prediction for training points when P=10. 32
Figure II-12:Prediction for training points when P=50. 32
Figure II-13:Prediction for unseen points when P=10. 33
Figure II-14:Prediction for unseen points when P=50. 33
Figure III-1:Closed-loop control System for a 3-tank process with disturbance Injection 38
Figure III-2:Three-Tank-System 39
Figure III-3:Real-Time Control System for a Three-Tank Setup Using Arduino and Simulink 40
Figure III-4:The learning scheme of the intelligent calibration system 47
Figure III-5:The learning scheme of the intelligent calibration system 48
Figure III-6:Sensor 1 response after level calibration using FLANN-GD with training data 51
Figure III-7:Sensor 2 response after level calibration using FLANN-GD with training data 51
Figure III-8:Sensor 3 response after level calibration using FLANN-GD with training data 52
Figure III-9: Sensor 1 response after level calibration using FLANN-GD with testing data. 52
Figure III-10:Sensor 2 response after level calibration using FLANN-GD with testing data 53
Figure III-11:Sensor 3 response after level calibration using FLANN-GD with testing data 54

Figure III-12:Piecewise linear (PWL) calibration method 54
Figure III-13:Sensor 1 response after level calibration using FLANN-GD with PWL interpolation with training data 55
Figure III-14:Sensor 2 response after level calibration using FLANN-GD with PWL interpolation with training data 56
Figure III-15:Sensor 3 response after level calibration using FLANN-GD with PWL interpolation with training data 56
Figure III-16:Sensor 1 response after level calibration using FLANN-GD with PWL interpolation with testing data 57
Figure III-17:Sensor 2 response after level calibration using FLANN-GD with PWL interpolation with testing data 57
Figure III-18:Sensor 3 response after level calibration using FLANN-GD with PWL interpolation with testing data 58
Figure III-19:Real-Time Control System for a three-Tank Setup Using Arduino and Simulink with our new model 59
Figure III-20:Sensor1 after level calibration using FLANN-GD without PWL interpolation60
Figure III-21:Sensor 2 after level calibration using FLANN-GD without PWL interpolation 60
Figure III-22:Sensor 3 after level calibration using FLANN-GD without PWL interpolation 61
Figure III-23:Sensor 1 after level calibration using FLANN-GD with PWL interpolation ... 62
Figure III-24:Sensor 2 after level calibration using FLANN-GD with PWL interpolation ... 62
Figure III-25:Sensor 3 after level calibration using FLANN-GD with PWL interpolation ... 63

List of tables

Table I-1 : Some examples of passive sensors 11
Table I-2: Some examples of active sensors and their output 11
Table II-1: Experimental observations of LVDT 31
Table II-2: Prediction of training points 32
Table II-3: Prediction of testing points 33
Table III-1: Input-output dataset of tank 1 43
Table III-2: Input-output dataset of tank 2 45
Table III-3: Input-output dataset of tank 3 47
Table III-4 : Observed simulation results 50
Table III-5 : Results of Gradient-based training of the FLANN model using PWL 55
Table III-6: Results using FLANN-GD without PWL interpolation 59
Table III-7: Results using FLANN-GD with PWL interpolation 61

List of Abreviations

FLANN: Functional Link Artificial Neural Network

PWL: Piecewise Linear

RNN: Recurrent Neural Network

ANN: Artificial Neural Network

LVDT: Linear Variable Differential Transformer

ABC: Artificial Bee Colony

X: Measurement Quantity

[X]: Measurement Unit

{X}: Numerical Value of Measurement Quantity.

Abstract

This study focuses on designing an intelligent calibration model for level sensors by using the Functional Link Artificial Neural Network (FLANN). The FLANN has simple architecture and requires less computational effort compared to other neural networks models. This made it good enough in extending the linearity of many sensors and transducers as reported in recent literature. Despite its advantages, the standard FLANN model has limitations in terms of generalization and accuracy. To overcome this shortcoming, we propose in this study an approach that relies on integrating Piecewise Linear (PWL) interpolation with the FLANN model. This approach aims to improve the overall performance of the calibration process, offering better generalization capabilities. Building on an extensive experimental investigation of the intelligent calibration model on the typical problem of level sensors calibration, this manuscript outlines the methodology, implementation steps, and findings of our study.

ملخص

في أنظمة التحكم الحديثة، يعد معايرة المستشعرات أمرا حاسما لضمان قياسات دقيقة وموثوقة، وهو أساسي لأداء النظام الأمثل. تعمل المعايرة على تصحيح الأخطاء النظامية في قراءات المستشعرات، مما يمكن من التحكم والمراقبة بدقة. ومع ذلك، غالبا ما تواجه نماذج المعايرة التقليدية صعوبة في التكيف والتعميم، مما يؤدي إلى أخطاء كبيرة في البيئات المعقدة أو الديناميكية. تركز هذه الدراسة على تطوير نموذج معايرة مبتكر باستخدام الشبكة العصبية الاصطناعية ذات الروابط الوظيفية (FLANN). تعرف FLANN بهيكلها البسيط وقدرتها الجيدة على تعويض اللاخطية، وقد استخدمت في مهام المعايرة المختلفة. على الرغم من مزاياها، تواجه نموذج التقليدي قيودا من حيث التعميم والدقة. لمعالجة هذه FLANN. تهدف هذه الطريقة إلى تحسين الأداء FLANN مع نموذج (PWL) القيود، نقترح دمج طريقة الاستيفاء الخطي الجزئي العام لعملية المعايرة، مما يوفر قدرات تعميم أفضل. يوضح هذا المستند المنهجية، خطوات التنفيذ، ونتائج دراستنا. نبدأ في هذا السياق. بعد ذلك، نناقش قيود FLANN بوصف الحاجة إلى معايرة المستشعرات في أنظمة التحكم وتطبيق التقليدي ونقدم طريقة الاستيفاء الخطي الجزئي كحل لهذه التحديات. ثم نقدم نتائج تجاربنا التي تقارن FLANN نموذج مع وبدون الاستيفاء الخطي الجزئي، مسلطين الضوء على التحسينات المحققة. تكمن أهمية هذا FLANN أداء نماذج المشروع في قدرته على تعزيز موثوقية وكفاءة معايرة المستشعرات في أنظمة التحكم من خلال دمج الخوارزميات المتقدمة وتقنيات الاستيفاء.

Résumé

Cette étude se concentre sur la conception d'un modèle de calibration intelligent pour les capteurs de niveau en utilisant le réseau neuronal artificiel à liens fonctionnels (FLANN). Le FLANN possède une architecture simple et nécessite moins d'effort de calcul par rapport à d'autres modèles de réseaux neuronaux. Cela le rend suffisamment performant pour étendre la linéarité de nombreux capteurs et transducteurs, comme rapporté dans la littérature récente. Malgré ses avantages, le modèle FLANN standard présente des limitations en termes de généralisation et de précision. Pour surmonter cette lacune, nous proposons dans cette étude une approche qui repose sur l'intégration de l'interpolation linéaire par morceaux (PWL) avec

le modèle FLANN. Cette approche vise à améliorer la performance globale du processus de calibration, offrant de meilleures capacités de généralisation. En s'appuyant sur une investigation expérimentale approfondie du modèle de calibration intelligent sur le problème typique de calibration des capteurs de niveau, ce manuscrit décrit la méthodologie, les étapes de mise en œuvre et les résultats de notre étude.

Introduction

Introduction

In modern control systems, sensor calibration is crucial for ensuring accurate and reliable measurements, which are fundamental for optimal system performance. Calibration corrects systematic errors in sensor readings, enabling precise control and monitoring. However, traditional calibration models often struggle with adaptability and generalization, which can lead to significant inaccuracies in complex or dynamic environments. This study focuses on the development of an innovative calibration model utilizing the Functional Link Artificial Neural Network (FLANN). FLANN is known for its simple architecture with good non-linearities compensation and has been employed in various calibration tasks. Despite its advantages, the standard FLANN model has limitations in terms of generalization and accuracy.

To address these limitations, we propose the integration of Piecewise Linear (PWL) interpolation with the FLANN model. This approach aims to improve the overall performance of the calibration process, offering better generalization capabilities. This document outlines the methodology, implementation steps, and findings of our study. We begin by describing the need for sensor calibration in control systems and the application of FLANN in this context. Next, we discuss the limitations of the traditional FLANN model and introduce the PWL interpolation method as a solution to these challenges. The results of our experiments, comparing the performance of FLANN models with and without PWL interpolation, are then presented, highlighting the improvements achieved. The significance of this project lies in its potential to enhance the reliability and efficiency of sensor calibration in control systems by integrating advanced algorithms and interpolation techniques.

Chapter.I. Measurement systems and calibration

I.1. Introduction

The renowned chemist D. Mendelev famously proclaimed, "Science begins where measurement begins". [1]

This profound statement underscores a fundamental truth: without measurement, science itself ceases to exist. Measurement is not merely a passive act of recording observations; it's a cornerstone of scientific inquiry, a powerful tool that transforms qualitative experiences into quantitative data.

Through the process of measurement, we bridge the gap between the physical world and our understanding of it. By engaging in physical experiments and observations, we can assign a numerical value to a particular property or characteristic. This value, known as a quantity, allows us to compare different objects, phenomena, or events in a precise and objective manner.

The essence of measurement lies in establishing a relationship between the quantity we're interested in (the measurable) and a pre-defined unit of the same nature. This unit serves as a reference point, a standard against which we can calibrate our measurements.

The following sections will delve deeper into the intricacies of measurement, exploring various techniques, tools, and the fundamental principles that govern this critical scientific endeavor.

I.2. Metrology and related terms

The word "metrology" comes from the Greek words "metron" (measure) and "logos" (study), so it literally means "the science of measurement." [2] In the everyday language of metrologists, you often hear the saying "to measure is to compare!" [3]

Measurement results are used to make decisions like for instance :

- Acceptance of a product (measurement of characteristics, performance, conformity to a requirement);
- Adjustment of a measuring instrument, validation of a procedure;
- Adjustment of a parameter within the framework of the control of a manufacturing process;
- Validation of a hypothesis;
- Definition of the safety conditions of a product or system.
- A measurement result is written in the form: $X = \{X\}\ [X]$

where X is the name of the physical quantity, [X] represents the unit, and {X} is the numerical value of the quantity expressed in the chosen unit. [4]

We give in the following some metrology terms used to describe key quantities of measurement systems.

Measurable quantity: Defined as an attribute of a phenomenon, body, or substance that can be qualitatively distinguished and quantitatively determined. [5]

Unit of measurement: It is a particular quantity, defined and adopted by convention, to which other quantities of the same nature are compared to express them quantitatively.[6]

Measurement: It is the set of operations that aim to determine a value of a quantity. [2]
Mesurand: A particular quantity subjected to measurement. [7]

Measurement uncertainty: This term refers to a parameter associated with the result of a measurement that characterizes the dispersion of values that could reasonably be attributed to the measurand. In other words, it represents the range within which the true value of the measurand is likely to lie. [8]

Measurement standard: In metrology, a measurement standard is a device that is used as a reference to control the accuracy of the results provided by a measuring instrument. Measurement standards are rigorously calibrated and traceable to higher-order standards, ensuring a consistent and reliable basis for measurement across different laboratories and institutions. [9]

I.3. Sensors and transducers

A sensor is a device that consists of a sensing element that is sensitive to the physical quantity to be measured and a transducer that converts it into an electrical quantity, generally analog [10]. A schematic block-diagram of a sensor is depicted in Figure I-1.

The signal conditioning circuit converts the electrical quantity into a quantity suitable for the information processing system (PLC, computer, servo system, etc.). [11]

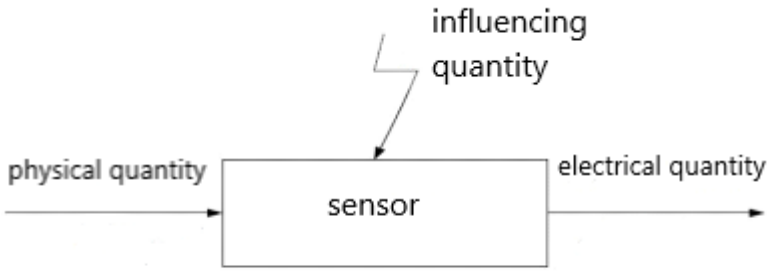

Figure I-1 Block diagram of a sensor [11]

The sensor is characterized by its function: s = f(m), called characteristic function where s is the output quantity or response of the sensor and m is the physical quantity to be measured (measurand, example temperature). [12]

I.4. Measurement Chain

In general, the quantity to be measured, called the measurand, is not directly accessible, and the measurement methods used involve different physical laws and material properties. [10] A measurement chain is generally made up of the following elements, as shown in Figure I-2. [11]

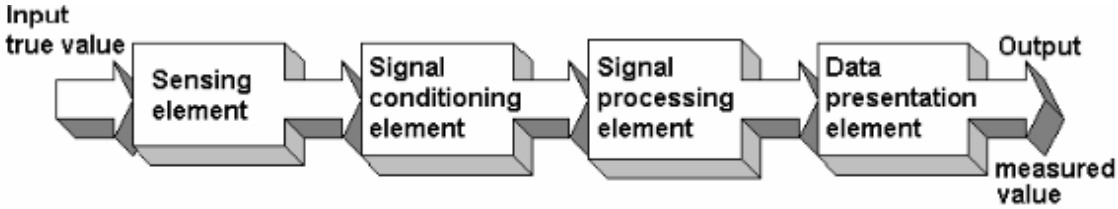

Figure I-2: Typical Diagram of a Measurement Chain [11]

Transducer:

A transducer is the fundamental element of the device, based on the use of a particular physical law. It makes a value Gs of another quantity, generally electrical, called the output quantity, correspond to a value Ge of the quantity to be measured. Transducers are generally sought such that the relationship between the variation of the measurand and the variation of the output signal of the transducer is linear, or at least to use the linear part of this relationship if it is more complex. [12]

Signal conditioner:

A signal conditioner is an electrical or electronic circuit that converts, compensates, or modifies the output signal of the transducer in order to transform it into a standard electrical signal. The conditioner is often physically inseparable from the transducer. The Wheatstone bridge or potentiometric assembly, which are conditioners for passive sensors, thus make it possible to transform the variation in resistance of the transducer into a variation in voltage across the bridge terminals. [11]

Amplifier:

An amplifier is an essential element when the output signal of the conditioner is weak, it is very often necessary to amplify them in ratios of 10 to 1000, or more. After amplification, voltages are generally reached between 0 and 5 or 10V. [10]

Display/recorder:

A display/recorder is an element that measures the signal (current or voltage) coming out of the amplifier to restore it in a readable and interpretable form for the user. [13]

The processor:

This is an element present in all measurement devices that displays and/or delivers a digital signal. It is typically an analog-to-digital converter (ADC). [14]

In practice, the term "sensor" can refer to different things depending on the authors and speakers:

- The transducer itself;
- The transducer + conditioner assembly;
- The entire measurement chain represented in the figure above.

The distinctions can be difficult because increasingly, transducers are physically associated with conditioners and amplifiers, thanks to advancements in miniaturization that have significantly reduced the size of these components. The main advantage of this hardware integration is the reduction of disturbances to the transducer's output signal (interferences, noise, energy and signal losses, etc.) before its processing by subsequent elements. [15]

We will refer to the sensor as the part of the measurement chain in contact with the environment where the measurement takes place, and the transmitter as the rest of the elements in the measurement chain. [16]

I.5. Sensor Characteristics

Sensors are designed to reproduce variations in the physical quantity being measured with high quality. They must exhibit stable, reproducible, and reliable behavior. Common sensor characteristics are:

– **Measurement range and operating limits:**

The measurement range is the difference between the extreme (minimum and maximum) values that can be measured by the measurement chain. It is expressed as:

Measurement Range = Maximum Measurable Value - Minimum Measurable Value

EM = Vmax - Vmin

Operating limits are the extreme (lower and upper) limits of the physical quantity that can be reproduced without deteriorating or altering the metrological characteristics of the sensor. [10]

Example: Temperature sensor: -20°C to 70°C EM = Tmax - Tmin = 70 - (-20) = 90°C

– **Absolute and Relative Error :**

Absolute error is the value of the error directly related to the measurement. It represents the difference between the measured value and the true value. [12]

Example: For a nominal value of 100°C, an absolute error of ±0.2°C indicates that the measured value could be anywhere between 99.8°C and 100.2°C.

Relative error is the percentage ratio between the absolute error and the measurement result. It expresses the error as a proportion of the true value, providing a more meaningful comparison for different measurement scales. It is calculated as:

Relative Error = (Absolute Error / True Value) × 100%

Example: Using the same example from absolute error, the relative error would be:

Relative Error = (0.2°C / 100°C) × 100% = 0.2%

A relative error of 0.2% indicates that the measured value deviates from the true value by 0.2% of the true value. This provides a more contextual understanding of the error's significance.

- **Resolution in Measurement Systems**

Resolution is a fundamental concept in measurement systems, representing the smallest discernible change in the measurand (the quantity being measured) that can be detected and distinguished by the sensor or measurement device. It is often expressed in units of the measurand, such as meters, volts, or degrees. [11]

- **Sensitivity**

It represents the ratio between the change in output signal and the change in input signal at a given point of the transfer function (Input/Output relationship).

The function of a sensor is to deliver an electrical output signal S that is a function of the measurand m. The sensor provides us with the relation S = f(m).

Sensitivity determines how the output quantity changes with respect to the input quantity at a given point. It corresponds to the slope of the tangent to the curve derived from the sensor's characteristic.

In the case of a linear sensor, the sensitivity of the sensor is constant. [13]

We can express it as: Sensitivity: (d'O)/(d'I) = (d'O/dm)

In differential form, Sensitivity at a given point x0 is given by: S = dy/dx at x0;

where y is the output and x is the input.

It should be noted that the sensitivity of a sensor can be influenced by the conditioner with which it is associated.

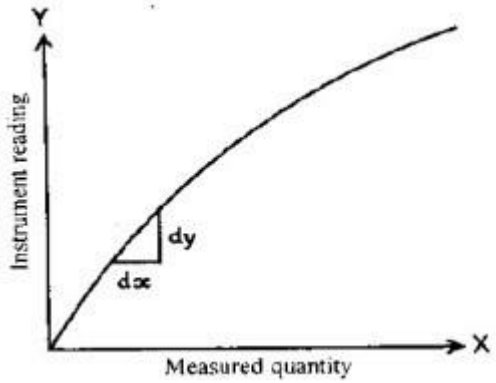

Figure I-3 Sensitivity of measurement [13]

- **Linearity:**

Linearity is expressed as a percentage, representing the maximum relative error between the regression line and the actual characteristic. [14]

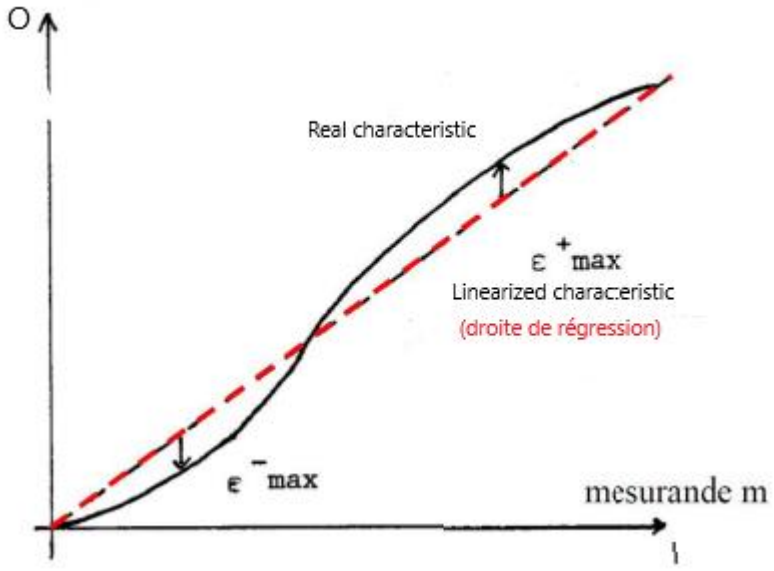

Figure I-4: Calibration curve, Transfer function, or Output curve = f(Input) [14]

If a sensor is theoretically linear, its sensitivity S is determined by the full-scale range.

For example, consider a temperature sensor with a range from -10°C to 240°C:

- Setting: Minimum output = 0% or 4mA for Tmin = -10°C
- Maximum output = 100% or 20mA for Tmax = 240°C
- Full-scale range = 250°C

The sensitivity of the sensor is thus $S=\frac{16\ mA}{250\ ^\circ\ \mathrm{C}}$ =0.064 mA per °C

Measurement Errors

Measurement errors can be categorized into the following types of error:

- **Systematic Errors:** Systematic errors are consistent, repeatable errors caused by faulty equipment or flawed experimental design. Examples include calibration errors, environmental influences, and instrument drift. These errors can often be identified and corrected. Examples: Drift, aging, improper usage, etc. [16]
- **Random (Accidental) Errors:** Random (accidental) errors are unpredictable variations in measurements due to unknown and uncontrollable factors. These errors affect precision and can be reduced by averaging multiple measurements. Examples: Noise, interference, parallax error, etc. [15]

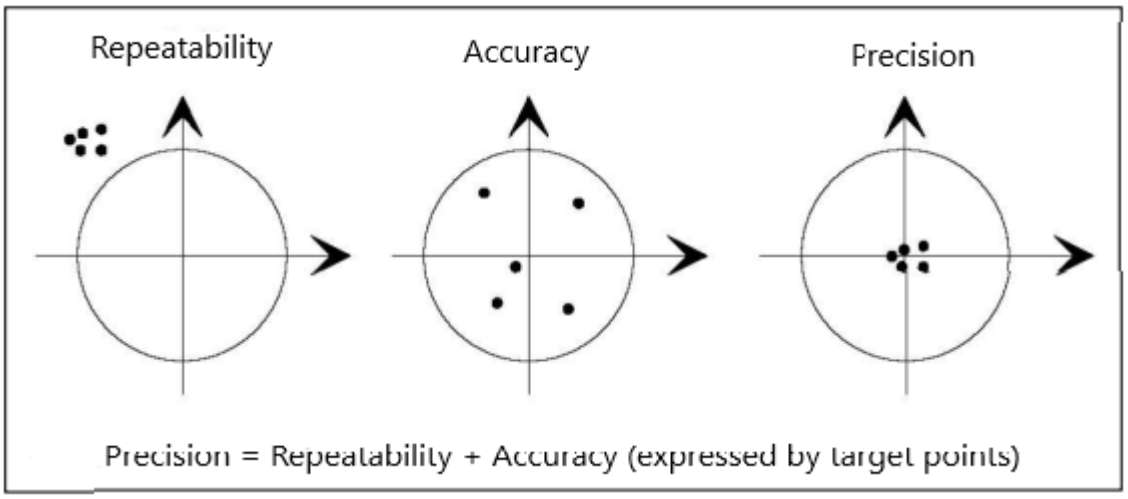

Figure I-5: Simplified scheme about measurement errors

Repeatability

It is the ability of a sensor to reproduce the same output signal when subjected to the same magnitude. It is expressed as a percentage of the measurement range. [2]

Accuracy

Refers to how close a measurement is to the actual or accepted value. It reflects how well a measurement system hits the bullseye of the target. [3]

Precision

Refers to how repeatable or reproducible a measurement is under unchanged conditions. It reflects how tightly grouped your measurements are, even if they might not be centered on the bullseye. [7]

Hysteresis

When the transfer curves of the sensor for an increasing and decreasing variation of the magnitude physical are not identical, we are talking about an error due to the hysteresis of the sensor. This error is expressed as a percentage of the measurement range, see Figure I-5. [8]

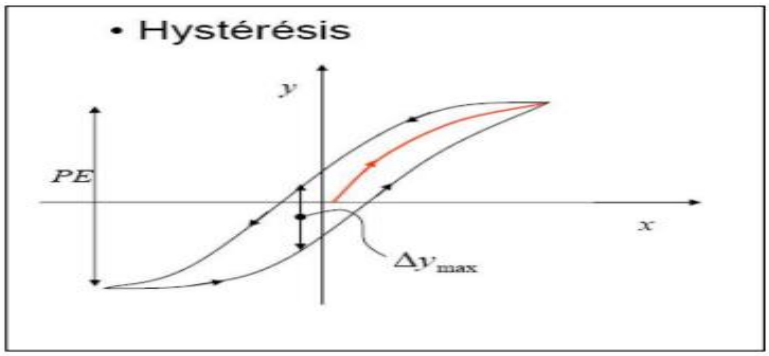

Figure I-6 Hysteresis phenomenon [8]

Finess

It is the quality expressing the ability of a sensor to give the value of the quantity to be measured without changing this one by its presence. [9]

Speed

Speed indicates the ability of a sensor to track changes in size over time.

This is because it always takes some time for a change in the input signal to be perceived at the output. It is expressed in one of three ways:

- Response time (or time constant);
- Sensor bandwidth;
- The cut-off frequency (or clean frequency). [10]

I.6. Sensors classification

Sensors are the building blocks of data acquisition systems. Typically, we get output size of electrical type. It can be either: A charge, voltage, current and impedance (R, L, C). Depending on the output quantities, there are two types of sensors: active and passive sensors. This classification has an influence on the conditioner associated with it. Thus, we distinguish between conditioners for active sensors (transistors, operational amplifiers, …); conditioners for passive sensors (Divider bridge, Wheatstone bridge, …). Depending on the measurand, several effects are used to perform the measurement.

Passive sensors

In most cases, passive sensors need outside energy to operate (such as in the case of strain gauges, thermistors, etc.), they are often modelled by an impedance (R, L,C). A variation of the physical phenomenon studied (measured) generates a variation of the impedance. They must be apply voltage to obtain an output signal. The sensor behaves in output as a passive dipole which can be resistive, capacitive or inductive (R, L,C). Examples of passive sensors are given in Table I-1.

Measured quantity	Electrical characteristic	type of materials used
Temperature	Resistivity	Metals: platinum, nickel, copper...
Very low temperature	Dielectric constant	Glass
Optical radiation flux	Resistivity	Semiconductor
Deformation	Resistivity Permeability	Nickel alloy, doped silicon Ferromagnetic alloy

Position (magnetic)	Resistivity	Magneto-resistive materials: bismuth, indium antimonide
Humidity	Resistivity	Lithium chloride

Table I-2 Some examples of passive sensors

Active sensors

When the physical phenomenon used to determine the measurand directly performs the transformation into an electrical quantity, we are dealing with an active sensor. It is the physical law itself that links the measurand to the electrical output quantity. The sensor output is then assimilated to a generator. It is an active dipole that can be of the current, voltage, or electric charge (Q in coulombs) type. Certain physical principles can be employed. Typical examples of active sensors are shown in Table I-2.

Measured Physical Quantity	Utilized Effect	Output Quantity
Temperature	Thermoelectricity	Voltage
Current	Photoemission	Optical radiation flux
Optical radiation flux	Photovoltaic effect	Voltage
Optical radiation flux	Photoelectric effect	Voltage
Force	Piezoelectricity	Electric charge
Pressure	Piezoelectricity	Electric charge
Acceleration	Piezoelectricity	Electric charge
Speed	Electromagnetic induction	Voltage
Position (magnet)	Electromagnetic induction	Voltage
Current	Hall effect	Voltage

Table I-3: Some examples of active sensors and their output

As shown in Figure I-7, the effects used in active sensors are categorized according to their physical nature. We can distinguish the following types of effects [11]

Thermoelectric effect: A circuit formed by two conductors of different chemical natures, with junctions at temperatures T1 and T2, generates an electromotive force ($T1,2$). Application: Determining an unknown temperature T1 from the measurement of ee when T2 (e.g., 0°C) is known.

Pyroelectric effect: Certain crystals, such as triglycine sulfate, have a spontaneous electrical polarization dependent on their temperature. Application: Absorbed radiation flux raises the temperature of a pyroelectric crystal, altering its polarization, which can be measured by the voltage variation across an associated capacitor.

Piezoelectric effect: Applying a force or mechanical stress to piezoelectric materials, such as quartz, induces a deformation that generates electrical charges of equal and opposite signs. Application: Measuring forces or related quantities (pressure, acceleration) based on the voltage changes across a capacitor linked to the piezoelectric element.

Electromagnetic induction effect: A conductor moving within a fixed induction field generates an electromotive force proportional to its velocity. Application: Measuring the induced electromotive force to determine the speed of the displacement causing it.

Photoelectric effects: Several types, all originating from the liberation of electrical charges in a material under electromagnetic radiation with a wavelength below a threshold characteristic of the material.

Photoemissive effect: Liberated electrons are emitted from the illuminated target, forming a current collected by an applied electric field.

Photovoltaic effect: Electrons and holes are liberated near an illuminated P-N semiconductor junction, and their movement within the junction's electric field alters the voltage across it.

Photoelectromagnetic effect: An applied magnetic field perpendicular to the radiation causes a voltage in the illuminated material in a direction normal to both the field and the radiation. Applications: Measuring photometric quantities and converting light-borne information into electrical signals.

Hall effect: A semiconductor material, typically in wafer form, subjected to a current I and an induction B at an angle T, produces a transverse voltage vH given by:

$$vH = KH \cdot I \cdot B \cdot \sin T$$

where KH depends on the material and wafer dimensions. Application: An attached magnet's values of B and T relative to the wafer translate the object's position into an electrical signal via vH.

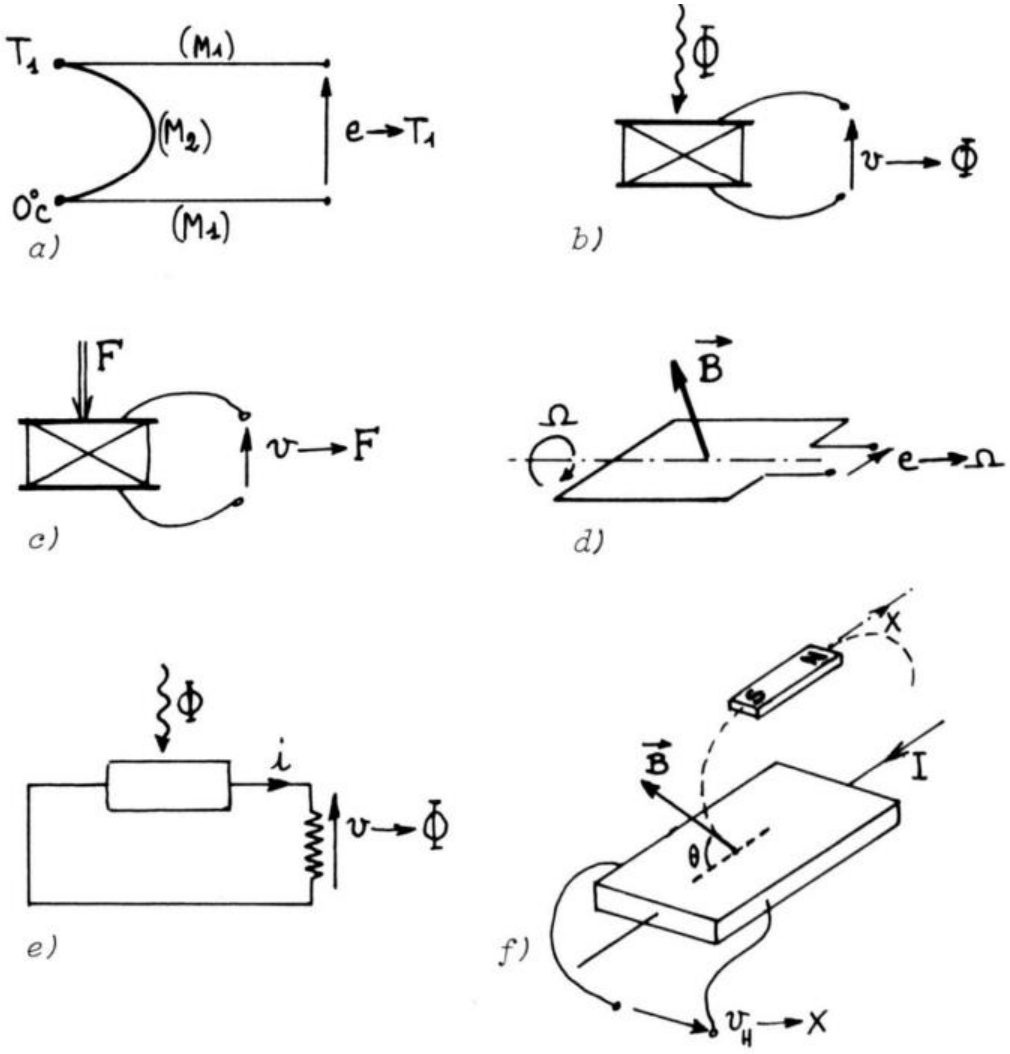

Figure I-7:Examples of the application of physical effects in the realization of active sensors: (a) thermoelectricity, (b) pyroelectricity, (c) piezoelectricity, (d) electromagnetic induction, (e) photoelectricity, (f) Hall effect. [11]

I.7. Non-linearity of sensors

The measurement operation plays a crucial role in ensuring the proper functioning of physical processes. Industrial activities have rapidly evolved, thanks to measurement technologies, with robotics being a significant example. In automatic regulation, good measurement is directly linked to the quality of control desired, thereby ensuring reliable management of the physical quantities associated with the system. The tools provided by electronics have enabled the translation of any physical quantity to be evaluated into an electrical quantity, which is the role of the sensor [17].

Some sensors have nonlinear response, this sensors stand out as they are widely used in chemical processes, the food industry, environmental monitoring, and biomedical applications, etc.. Nonlinearities can be due to the characteristic of the sensor or because of influencing quantities effect.

Interpolation methods are commonly used to correct these nonlinearities. The drop in microprocessor prices has facilitated the replacement of analog linearization techniques with digital techniques. In this context, artificial intelligence based systems artificial neural networks have been suggested as tools for sensor calibration including linearization or non-linearity compensation. [18]

Let us consider a sensor described by a general model of the form:

$y=(x,g)$ (I.1)

where y is an electrical signal, x is the quantity to be measured and g is an influencing quantity. In the case of a passive sensor, where there is a variation in impedance, this is translated into an electrical signal using a conditioner [17]

The variation in the measurement y is given by the Taylor series expansion around the point (x0,0) up to the second order:

$$\Delta y = \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial g}\Delta g + \frac{\partial^2 f}{2\partial x^2}(\Delta x)^2 + \frac{\partial^2 f}{2\partial g^2}(\Delta g)^2 + \frac{\partial^2 f}{2\partial x \partial g}\Delta x \Delta g + \frac{\partial^2 f}{2\partial g \partial x}\Delta g \Delta x \qquad \text{(I.2)}$$

The term $S = \frac{\partial f}{\partial x}$ is called the static sensitivity of the sensor.

The term $S_g = \frac{\partial f}{\partial g}$ is called noise sensitivity.

The term $S_d = \frac{\partial^2 f}{\partial x^2}$ is called the sensitivity drift with respect to x.

A sensor is said to be nonlinear if its sensitivity to the measurand x depends on x, i.e.,

$$\frac{\partial^2 f}{\partial x^2} \neq 0 \qquad \text{(I.3)}$$

In the dynamic regime, a sensor is considered nonlinear if it is governed by a nonlinear differential equation. In this case, we refer to dynamic sensitivity, which depends on the frequency of the measurand, or the transfer function. Examples of nonlinear sensors include thermal resistances, air flow sensors, gas composition sensors and ionic sensors.

I.7.1. Linearization of a sensor's characteristic

This is the question that a user asks every time they need to use a sensor. It's true that it all depends on the application to which the sensor will be applied. Sometimes, there's no need to linearize for reasons deemed valid and essential for the proper functioning of the instrument. However, more often than not, this isn't the case, and adequate linearization becomes essential.

When a sensor's characteristic is linear, the output signal can exceed the visualization range, but this problem can be solved by simply scaling the quantities. [19] However, if the sensor is nonlinear, it presents other types of problems that may be more difficult to resolve. Moreover,

linearization has the advantage of increasing the precision of sensors and their measurement range.

Generally, we approximate the nonlinear parts of the sensor's characteristic with line segments around operating points. This method has the disadvantage of having a narrow measurement range, which decreases even more if the sensor is highly nonlinear. In control systems, linearizing a nonlinear block simplifies the synthesis of the control law by using linear system theory.

I.7.2. Sensor linearization methods

I.7.2.1. Classical methods

In instrumentation, there are two main methods for linearizing nonlinear sensors: analog linearization and digital linearization [19].

The first method uses nonlinear analog circuits placed after the sensor conditioner. These circuits take the sensor's measurement signal as input and produce a linear output signal based on the measurand. They can be designed using nonlinear electronic components such as diodes and transistors, aiming to approximate the inverse function of the sensor's characteristic obtained through calibration. This method is used when designing low-cost and high-speed linearizing circuits. However, it is limited because it is only operational if the measurement signal depends solely on the measurand, or when disturbance quantities are time-invariant. In other words, if the sensor is highly sensitive to noise, it will be difficult to linearize it using an analog circuit.

The second method is based on implementing a program in a ROM memory in the form of a table of measurements taken on the sensor, which provides us with the value of the measurand each time the measurement value is given. However, the sensor's output signal must be conditioned and digitized through an A/D conditioner. The advantage of this method is significant because it allows linearization while considering the influence of disturbances, which is not possible with analog linearization. In this way, the measurements become undoubtedly more accurate. The principle of the method simply involves performing linear or quadratic interpolation from the data in the table containing the sensor's inputs/outputs.

I.7.2.2. Calibration methods

A certain number of measurements need to be taken in order to determine and correct a sensor’s nonlinearity. The number of measurements necessary to reduce the linearity error depends on the linearising calibration method used and to reduce the costs of calibration it is important to minimize the number of measurements costs refer to the expense of processing power and time. This is an important criterion in the selection of an appropriate linearising calibration method for sensor calibration

All of the following methods are based on the use of calibration measurements [20]:

- Look-up table,

- Piecewise linear interpolation,
- Piecewise polynomial or spline interpolation,
- Error minimization,
- Sensor characteristic linearization,
- Curve fitting,
- Progressive polynomial calibration

A brief description of each of these is given below after which then follows more detailed approach to methods which were chosen for further investigation.

Look-up table

In this method, the sensor output is measured for a large number of known input signals and the values, so obtained, are stored in a table and used as the ideal inverse sensor transfer function (Figure I-8). For a particular sensor output, the corrected sensor output value can be obtained by looking it up in the table. During calibration, an ADC is used to bring the sensor output signal into a digital form and this can be used as an address for the memory location holding the corrected digital value for the ideal transfer curve. The advantages of this method are that calibration can be realised in a simple manner and, once the lookup table is generated, the sensor signal correction is fast and does not require any signal processing. The disadvantages are that a large memory is required and a large number of calibration measurements are needed.

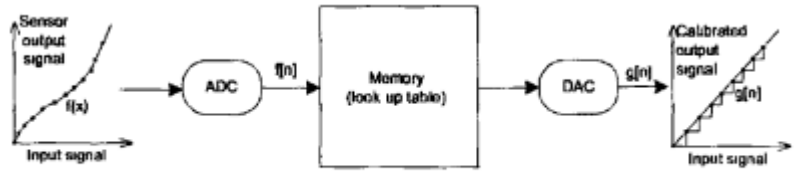

Figure I-8:Look-up table calibration method

Piecewise linear (PWL) interpolation

In this method, the sensor output is measured for a small set of known input signals. These calibration points become anchor points or “break” points on the ideal inverse transfer curve. The inverse transfer curve is completed by joining the “break” points with straight lines y = an + bnx. These lines form a piecewise linear interpolation of the ideal inverse sensor transfer curve (Figure I-9). The parameters that need to be stored in memory are the subrange limits and the required offset and gam corrections for each subrange. The advantages are a small number of calibration points, a small memory and signal processing that is relatively simple. The disadvantages are that error correction is modest, a large number of calibration points are needed for highly nonlinear transfer curves and separate calculations of offset and gam for each subsection are needed.

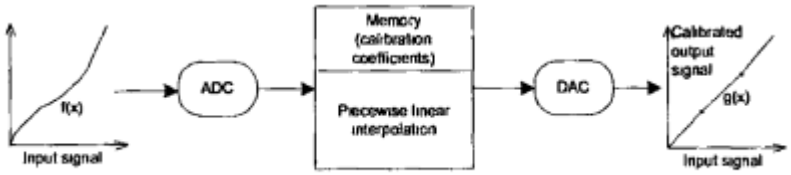

Figure I-9: Piecewise linear calibration method

Piecewise polynomial or spline interpolation

The sensor output is measured for a small set of known input signals. These calibration points become anchor points or "break" points on the ideal inverse transfer curve. The transfer curve is completed by adding polynomial curves (y = Pn(x)) between the "break" points where these curves are derived by polynomial interpolation between three or more "break" points. These interpolations can also be expressed mathematically as splines. These interpolations or splines form a piecewise polynomial or spline function representing the ideal inverse transfer curve over the complete input range. The advantages are fewer calibration points than the PWL method, good linearity that can be obtained with few calibration points and recursive equations suitable for software implementation can be used. The disadvantages are that the method requires advanced computations and storing different correction functions with different coefficients for each subsection of the transfer curve.

Error minimization

The error minimization is based on constructing the curve that does not exactly go through calibration measurements. The coefficients of the approximation curve are adjusted in way that the errors at the calibration points are minimized. Usually the mean square error ("Least Squares") is taken as the error function that should be minimized. The advantages are that good linearization can be achieved, but this is only the case of a large number of calibration measurements. The storage of many calibration measurements is needed because they are used in the error function optimization. In addition, there are problems of finding the optimum order for the linearization curve (number of coefficients), determining a good error function and how to minimize it. Usually the choice of a linearization function is based on some "knowledge" about the function. These are complex issues, especially for high order linearization and high order calibration (compensation) which makes this method not very suitable for realization. In addition to these disadvantages this method employs complex calculations of calibration coefficients.

Sensor characteristic linearization

Sensor characteristic linearization is based on function transformation (Figure I-10) but it is inappropriate when it comes to generic methods as it requires a pnort knowledge of a sensor characteristic.

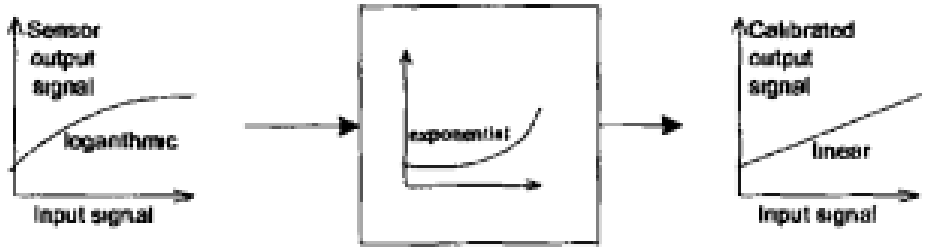

Figure I-10: Sensor characteristic linearization

If, for example a sensor transfer function is described by $v = ax^b$ then it is possible to determine a and b by transforming for instance the expression to $V = bX + n$

where $V = \log(v)$, $X = \log x$ and $n = \log a$ and using least squares estimates. Knowing a and b, the inverse function can be easily determined as : ($x = \sqrt[a/b]{\frac{v}{a}}$)

in order to linearize the sensor transfer function .The only advantage is that it is relatively easy to do it in software. The main drawback of the method is that the nonlinearity has to be well known. The other disadvantage is that the transformation of the nonlinear function is usually not good enough to reduce the nonlinearity.

Curve fitting

When the sensor output is measured for a small set of known input signals an interpolation algorithm is then used to compute a sensor transfer function using curve fitting techniques (Figure I-11). The inverse transfer function is then calculated so that input signal can be derived from the sensor output. The other possibility is to find a curve fit function for the inverse sensor transfer function straight away. The advantages of curve fitting are: an inverse transfer function is obtained for the complete signal range and memory requirements are low due to a small number of the coefficients. The disadvantages can be that sometimes higher order polynomials are necessary to get the desired accuracy. This involves more complex calculations and may require high accuracy computations (floating point arithmetic) as well.

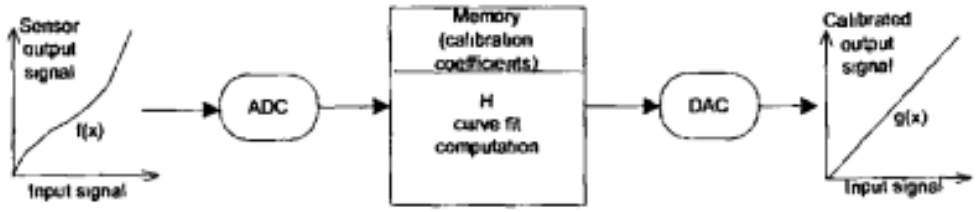

Figure I-11: Inverse transfer function method for linearizing calibration.

Progressive polynomial calibration

Progressive polynomial calibration operates on the principle that each calibration measurement is used directly to calculate one calibration coefficient in the correction function. This correction is then applied to the sensor output. Each step is independent of the previous one. The first measurement is used to correct the offset, the second corrects the gain and all the rest of the measurements are used for nonlinearity correction. The advantages are that it gives good linearisation for a minimum number of points, has low memory requirements due to a small number of the coefficients and uses a repetitive algorithm (step by step calibration).

The disadvantages are that some knowledge is needed when choosing calibration measurements.

I.8. Conclusion

In this chapter, we have explored the critical role of metrology in ensuring the proper functioning of physical processes. Accurate measurements are fundamental to industrial activities, enabling rapid advancements and playing a pivotal role in fields such as chemical processes. In automatic control, precise measurements directly influence the quality of regulation and the reliability of controlling physical quantities.

Sensors are key components in measurement systems. Non-linear sensors, such as thermistors, air flow sensors, gas composition sensors, and ionic sensors, present additional complexities. Their sensitivity to the measured quantity varies, making it difficult to achieve accurate measurements without proper linearization. Linearization is crucial for extending the measurement range and improving the precision of these sensors.

We discussed classical methods of sensor linearization, including both analog and digital approaches. Analog methods involve using non-linear circuits with components like diodes and transistors to approximate the inverse function of the sensor's characteristic curve. These are suitable for low-cost and high-speed applications but are limited by their sensitivity to

noise. Digital methods, on the other hand, utilize look-up tables and interpolation techniques, offering more precise linearization by accounting for disturbances.

We presented various linearizing calibration methods, emphasizing the importance of minimizing calibration costs while maintaining accuracy. The methods include look-up tables, piecewise linear interpolation, polynomial interpolation, and error minimization techniques. Each method has its advantages and is chosen based on the specific requirements of the sensor application.

In conclusion, the linearization of sensor characteristics is essential for enhancing the accuracy and reliability of measurements in various industrial and scientific applications. By understanding and addressing the non-linearity of sensors, we can ensure better control and monitoring of physical processes, ultimately leading to improved performance and efficiency.

Chapter.II. Functional Link Artificial Neural Networks

II.1. Introduction

Artificial Neural Networks (ANNs) have revolutionized the field of machine learning (ML) by mimicking the information processing capabilities of the human brain. In the context of this study, we are using neural networks as a fundamental tool for the intelligent calibration of level sensors. This motivated us to include an introduction to neural networks in this chapter to get better insight about them.

At the heart of ANNs lies the fundamental unit, the neuron, which performs a simple nonlinear function of its input variables. When neurons are interconnected in specific ways, they form neural networks capable of solving complex problems that individual neurons cannot. These networks, known as ANNs, are classified into two main types based on their structure: feedforward networks, which are static, and recurrent networks, which are dynamic. This chapter explores the architecture, functioning, and learning processes of neural networks, highlighting their applications and advantages. We also illustrate a typical example showing how neural networks could contribute to the crucial problem of compensating the nonlinearity of a linear variable differential transformer (LVDT).

II.2. Artificial Neural Networks

A basic neuron has limited applications; indeed, a neuron performs a simple nonlinear, parameterized function of its input variables. The interest in neurons lies in the property that results from their association in a structure, through a certain logic of interconnection. This structure is called a neural network, abbreviated as ANN (Artificial Neural Network). The collective behavior thus obtained allows the realization of higher-order functions compared to the elementary function performed by a single neuron.

In such a network, a neuron's inputs are either the overall network inputs or the outputs of other neurons. The values of the network weights are generally determined through a process called learning.[21]

Depending on the chosen logic of interconnection, neural networks are distinguished into two major families: feedforward networks (static) and recurrent networks (dynamic). Figures II-1 and II-2 illustrate the block diagrams of these two types of networks, respectively.

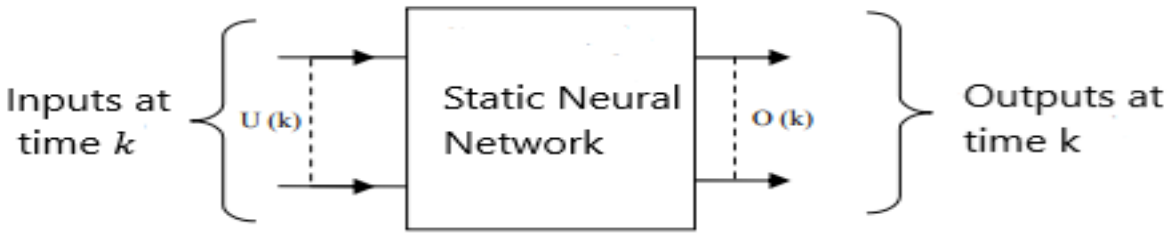

Figure II-1: Diagram of a Feedforward Neural Network (Static)

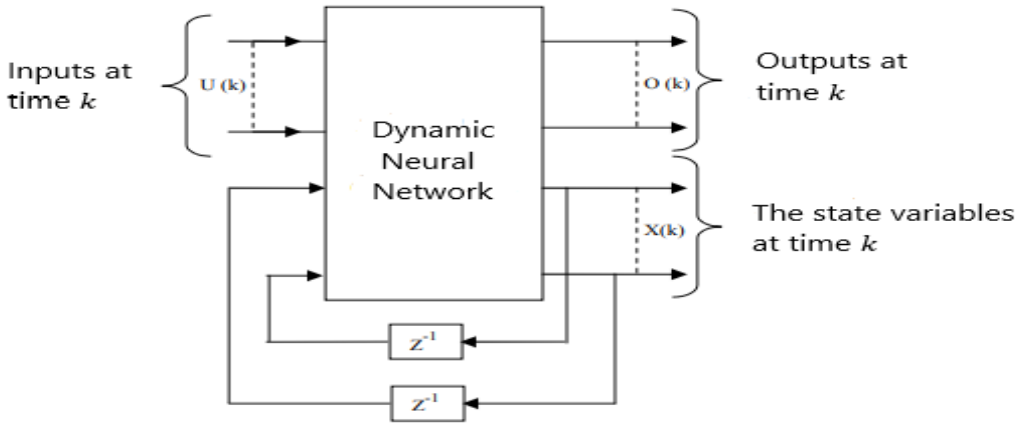

Figure II-2 : Diagram of a Recurrent Neural Network (Dynamic)

II.2.1. Recurrent Neural Networks (RNNs)

The most general architecture for a neural network is the recurrent neural network (RNN), where the connection graph is cyclic. When navigating through the network following the direction of connections, it's possible to find at least one path that returns to its starting point. Therefore, the output of a neuron in the network can depend on its own output from a previous time step, explicitly considering the concept of time [21]. Each connection in a recurrent neuron network is associated with a delay, which is a multiple integer of the chosen time unit (see Figure II-3).

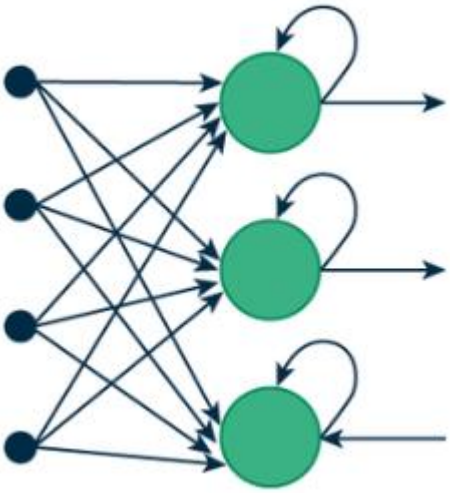

Figure II-3: Recurrent Neural Network

II.2.2. Feedforward Neural Networks

A feedforward neural network performs one or more algebraic functions of its inputs by composing the functions performed by each of its neurons. This network is graphically represented by a set of interconnected neurons (see Figure II-4). In such a network, the flow of information moves from inputs to outputs without "backtracking"; if you navigate through the network starting from any neuron and follow the connections, you cannot return to the starting neuron [21].

The neurons responsible for performing the final function composition are known as output neurons, whereas those involved in intermediate calculations are referred to as hidden neurons.

Another approach to organizing neurons is to arrange them into layers, ensuring that there are no connections between neurons within the same layer or between neurons in non-consecutive layers. These networks, known as multilayer perceptrons (MLPs), are illustrated in Figure II-4. The example MLP in the figure consists of three input neurons, one hidden layer with four neurons, and one output neuron.

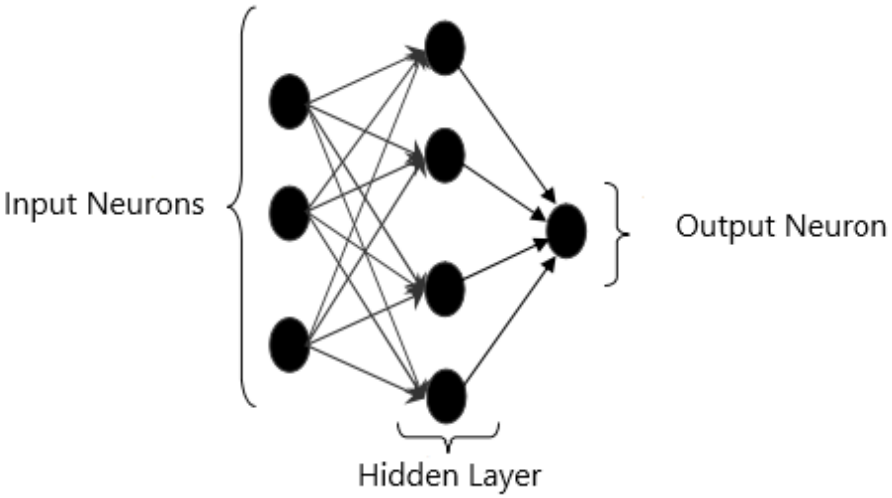

Figure II-4:Illustrative Example of a Multilayer Perceptron (MLP)

II.3. Neural Network Learning

The primary objective of developing neural network models is to achieve learning, which is accomplished by adjusting the network's connection weights using specific algorithms to find their optimal values. This process aims to fine-tune the network's performance to best address the target problem using provided training examples. These examples must be sufficiently representative, covering the network's desired operating domain as comprehensively as possible. A training dataset for a network consists of N examples, each comprising an input vector and a corresponding desired output vector. Depending on the learning rule applied, learning can be classified into two main types: unsupervised and supervised learning.

II.3.1. Unsupervised Learning

In unsupervised learning, the network aims to detect common patterns among the presented examples by adjusting the weights to produce similar outputs for inputs with similar characteristics.

Unsupervised learning is well-suited for modeling complex data such as images, sounds, etc., typically symbolic data [22], where the behavior of systems to be modeled by neural networks is governed by less precise rules.

II.3.2. Supervised Learning

As previously mentioned, a feedforward neural network performs an algebraic function mapping from its inputs to its outputs. Therefore, such a network can be tasked with performing a nonlinear algebraic function. In supervised learning, we provide the network with input-output pairs and adjust the weights based on the error between the desired output and the actual output (see Figure II-5).

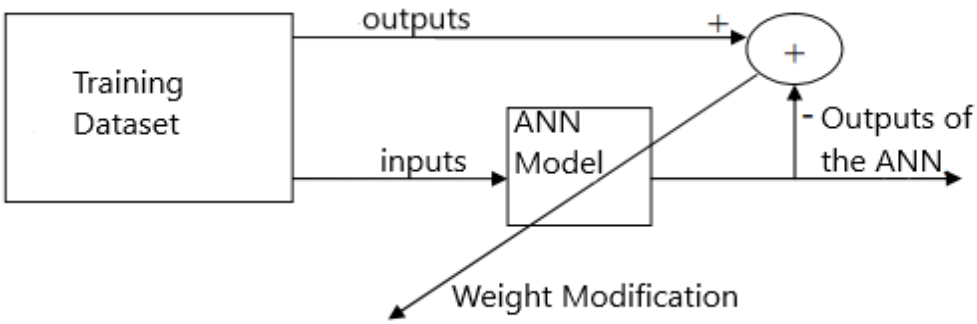

Figure II-5:Supervised Learning of an Artificial Neural Network (ANN) [22]

In this context, we can divide the algebraic function performed by this network into two parts:

- Known Analytical Function: This is a function that we understand analytically, and the network's task is to approximate this function using its learned parameters (weights and biases).
- Unknown Analytical Function with Finite Data: This refers to a function that is not analytically known, but we have a finite set of values for this function. These values may result from measurements taken in a physical, chemical, or other processes. In this case, the network performs static modeling or regression tasks [22].

II.4. Functional Link Artificial Neural Network

Functional Link Artificial Neural Networks (FLANN) represent a powerful extension of traditional single-layer neural networks, enhancing their capability to model complex relationships within data [23]. Unlike conventional neural networks that rely on multiple hidden layers to capture non-linear patterns, FLANN employs a unique approach involving the functional expansion of input data [24].

In a FLANN architecture, the original input features are transformed using a set of basis functions, such as trigonometric functions (e.g., sine and cosine), polynomial functions, or other non-linear transformations [25]. This expansion projects the input data into a higher-dimensional space, enabling the single-layer network to effectively approximate non-linear relationships [26]. The transformed features are then processed by the network, which comprises a fixed number of weight parameters optimized during training [27].

The training process of a FLANN typically employs gradient descent or similar optimization algorithms to iteratively adjust the weight parameters. This method aims to minimize the error between the predicted outputs and the actual target values, ensuring that the network provides the best possible approximation of the underlying function [28]. Due to its architectural simplicity and computational efficiency, FLANN is well-suited for a variety of applications, including function approximation, classification, and signal processing [29].

The advantages of FLANN include its reduced complexity compared to deep neural networks, faster training times, and the ability to effectively handle non-linear data relationships [30]. These attributes make FLANN a valuable tool for tasks where capturing intricate patterns without extensive computational resources is crucial [31].

II.4.1. Implementation of the FLANN neural network

The FLANN consists of N basis functions $\{\emptyset_1, \emptyset_2, \dots \emptyset_N\} \in B_N$ with the following input-output relationship for the jth output:

$$\hat{y} = \rho(S_j); S_j = \sum w_{ji} \emptyset_i(X) \quad \text{(II.1)}$$

Where X ∈ A ⊂ R^n, i.e., $X=[x_1 x_2 \dots x_n]^T$ is the input pattern vector, and $\hat{y} \in R^m$,i.e., $\hat{y}=[\hat{y}_1 \hat{y}_2 \dots \hat{y}_m]^T$ is the output vector. The weight vector associated with the jth output of the FLANN is $w_j=[w_{j1} w_{j2} \dots w_{jN}]^T$. The nonlinear function (.)=tanh(.).

Considering the m-dimensional output vector, equation (1) can be written as:

$$\hat{y} = W\Phi \quad \text{(II.2)}$$

where W is an $m \times N$ weight matrix of the FLANN given by :

$W=[w_1 w_2 \dots w_m]^T$, $\emptyset = [\emptyset_1(X) \emptyset_2(X) \dots \emptyset_N(X)]^T$ is the basis function vector, and $S=[S_1 S_2 \dots S_N]^T$, is a matrix of linear outputs of the FLANN. The m-dimensional output vector $\hat{y}$ may be given by:

$$\hat{y} = \rho(S) = f_w(X) \tag{II.3}$$

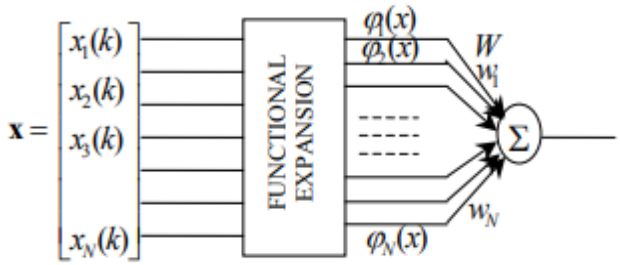

Figure II-6:The structure of a FLANN

Using the backpropagation (BP) algorithm for a single layer, the update rule for all the weights of the FLANN is given by:

$$w(k+1) = w(k) + \mu\delta(k)\emptyset(X_k) \tag{II.4}$$

where $w(k)$=$[w_1(k)w_2(k)\ldots w_m(k)]^T$ is the $m\times N$ dimensional weight matrix of the FLANN at the kth time instant, $\delta(k)$=$[\delta_1(k)\delta_2(k)\ldots\delta_m(k)]^T$, and $S_j(\mathrm{k}) = (1 - \hat{y}_j(\mathrm{k})^2)e_j(\mathrm{k})$.

Suitable orthogonal polynomials for functional expansion, such as Legendre, Chebyshev, and trigonometric polynomials, are commonly used. In this chapter, we have used trigonometric functional expansion. The trigonometric polynomial basis functions given by $\{1,\cos(\pi u),\sin(\pi u),\cos(2\pi u),\sin(2\pi u),\ldots,\cos(N\pi u),\sin(N\pi u)\}$ provide a compact representation of the function in the mean square sense.

II.4.2. FLANN types

Functional Link Artificial Neural Network (FLANN) is a type of higher Order Artificial Neural Networks that uses higher combination of its inputs [32].

It has been successfully used in many applications [33]. Here, we mentioned three types of functional expansions i.e., trigonometric expansion, Chebyshev expansion and Legendre expansion have been used to mitigate the nonlinear effects and their respected structures are shown in Figures II-7, II.8 and II.9, respectively.

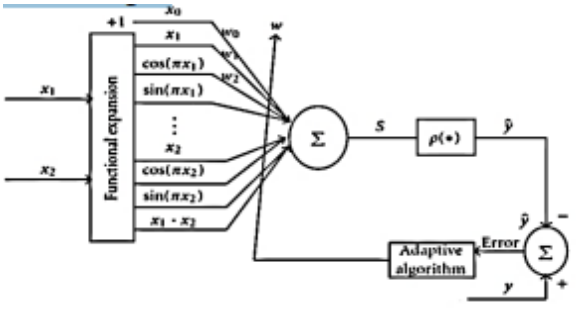

Figure II-7: Trigonometric expansion

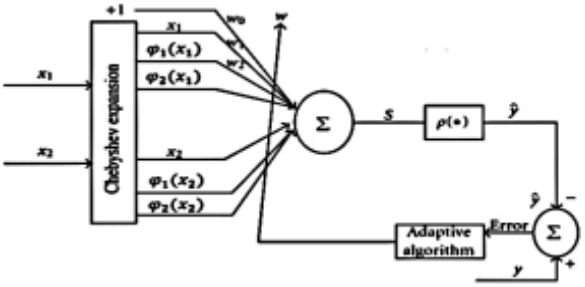

Figure II-8:Chebyshev expansion

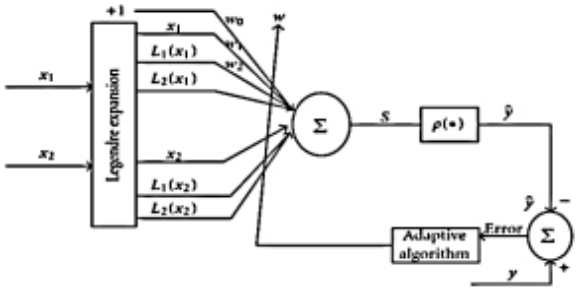

Figure II-9:Legendre Structure

Trigonometric expansion

Let us consider a two-dimensional input pattern $X=[x_1 x_2]^T$. This input pattern has been enhanced by functional expansion using Trigonometric functions as shown in equation (II.5).

$[x_1\ ,cos\ (\pi x_1),\ sin(\pi x_1),\ cos(2\pi x_1),\ sin\ (2\pi x_1),\ x_2,cos(\pi x_2),\ \sin(\ \pi x_2)\ ,\ cos\ (2\pi x_2),\ sin(2\pi x_2)\ ,\ x_1 x_2]^T$ (II.5)

Chebyshev expansion

The input can be expanded using Chebyshev polynomial as shown in equation (II.6).

$$\varphi=[\varphi_1\ (x_i\ (k)),\ \varphi_2(x_i\ (k)),\ldots\ldots\ \varphi_p\ (x_i\ (k))] \quad \text{(II.6)}$$

Legendre expansion

Legendre functional link artificial Neural Network (Le-FLANN) has a wonderful striking feature of faster training rate as compared to T-FLANN and Ch-FLANN. The Legendre expansion polynomials are represented by L_n, where n is the order of polynomial. The 0th and 1st order Legendre polynomials are given by $L_0(x)=1$ and $L_0(x)=x$. Legendre network higher order polynomials are formed by following mathematical equation (II.7).

$$\mathrm{L}_{n+1}(x)=\frac{1}{n+1}\left[(2n+1)\mathrm{L}_n(x)-n\mathrm{L}_{n-1}(x)\ \right] \quad \text{(II.7)}$$

For a two-dimensional input pattern $X = [x_1 x_2]$. This input pattern has been enhanced by Legendre functional expansion using the equation (II.8).

$$X^e = [1, L_1(x_1), L_2(x_1), L_3(x_1), L_1(x_2), L_2(x_2), L_3(x_2)] \quad \text{(II.8)}$$

For Legendre neural network, training procedure of network is same as that in FLANN.

II.5. Some Training algorithms

II.5.1. Gradient Descent Algorithm

Gradient Descent Algorithm (GDA) iteratively calculates the next point using gradient at the current position, scales it (by a learning rate) and subtracts obtained value from the current position (makes a step). It subtracts the value because we want to minimize the function (to maximize it would be adding). This process can be written as the equation (II.9):

$$p_{n+1} = p_n - \eta \nabla f(p_n) \quad \text{(II.9)}$$

There's an important parameter **η** which scales the gradient and thus controls the step size. In machine learning, it is called **learning rate** and have a strong influence on performance.

- The smaller learning rate the longer GDA converges, or may reach maximum iteration before reaching the optimum point
- If learning rate is too big the algorithm may not converge to the optimal point (jump around) or even to diverge completely.

In summary, Gradient Descent method's steps are:

1. Choose a starting point (initialization)
2. Calculate gradient at this point
3. Make a scaled step in the opposite direction to the gradient (objective: minimize)
4. Repeat steps 2 and 3 until one of the criteria is met:

- maximum number of iterations reached
- step size is smaller than the tolerance (due to scaling or a small gradient).[34]

II.5.2. Artificial bee colony (ABC) algorithm

The Artificial Bee Colony (ABC) algorithm mimics the foraging behavior of honey bees and consists of three types of bees: employed bees, onlookers, and scouts. Employed bees search for food sources based on their memory of previously visited sites, onlookers choose food sources based on the information shared by employed bees, and scouts randomly search for new food sources. The key components of the ABC algorithm include the initialization stage, where food sources (potential solutions) are randomly selected, and the repeated cycles of search, where employed bees visit known food sources and share their nectar information,

onlooker bees select food sources based on this shared information and search nearby, and scouts explore for new food sources when existing ones are exhausted.

The algorithm follows these steps. First, the population of solutions (food sources) is initialized. Then, in each cycle, employed bees search and share nectar information, onlooker bees select food sources based on this shared information and search nearby, and scouts randomly search for new food sources. Employed and onlooker bees modify solutions based on a probability related to nectar amount (fitness value), and a greedy selection mechanism ensures that bees retain the better solution between the current and new one. Scouts replace abandoned food sources with new randomly generated ones. The ABC algorithm uses three control parameters: the number of food sources (solutions), the limit for abandoning food sources, and the maximum cycle number.

The ABC algorithm efficiently balances exploration and exploitation. Exploitation is carried out by employed and onlooker bees focusing on improving known solutions, while exploration is managed by scouts searching for new solutions to maintain diversity. This balance leads to the rapid discovery and utilization of optimal solutions, analogous to how a bee colony optimally finds and exploits food sources.[35]

II.6. FLANN-based LVDT non-linearity compensation

In this section, we provide an illustrative example of application of the FLANN to the problem of non-linearity compensation of linear variable differential transformer (LVDT). The empirical data collection from an LVDT is detailed in [36]. The input-output characteristics have been analyzed and discussed, highlighting the issues that need to be addressed when modeling the non-linearity compensator unit for this sensor. We evaluated the performance of the LVDT set to the highest range. Experimental data is presented in Table II-1, revealing the presence of nonlinearity. We calculate the percentage of nonlinearity using this equation (II.10):

$$Percentage\ of\ Nonlinearity = \frac{Max\ deviation\ from\ the\ idealized\ straight\ line}{\text{Full scale deviation}} \times 100 \qquad \text{(II.10)}$$

Core Displacement (in mm)	Signal Conditioned Output Voltage (in Volts)
-30	-5.185
-25	-5.017
-20	-4.717
-15	-4.039
-10	-2.896
-5	-1.494
0	0.001

5	1.462
10	1.810
15	3.962
20	4.799
25	5.225

Table II-2:Experimental observations of LVDT

In this example, we present a simple design of an inverse model to compensate for the non-linearity exhibited by the LVDT. The basic functionality of an inverse model is to generate a mirror replica of the system under consideration. By cascading the non-linear LVDT sensor with the adaptive inverse model, we achieve overall linearity. The proposed model for this experimental setup is shown in Figure II-10.

We achieve the functionality of the inverse model through the realization of a Functional Link ANN (FLANN), which dynamically learns the system and generates its inverse characteristics.

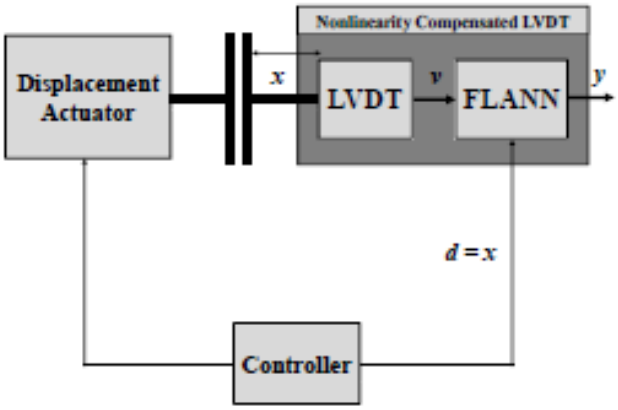

Figure II-10:Scheme of a non-linearity compensator of LVDT

To validate our model, we conducted predictions using different expansion number, P=10 and P=50, for the points cited in Table 1 and for other points that we did not explicitly include. These predictions help illustrate the performance and accuracy of the model under varying conditions. The results demonstrate how well the FLANN compensates for non-linearity in the LVDT sensor across different scenarios, providing a comprehensive evaluation of its effectiveness.

The predictions for the training points yielded the results cited in Table II-2.

Expansion number	Percentage of nonlinearity (%)

P=10	2.163
P=50	0.060

Table II-3: Prediction of training points

Figure II-(11-12) represent the results of compensation for the training points at *P*=10 and *P*=50, respectively. These figures visually depict the effectiveness of the FLANN model in mitigating non-linearity, highlighting the precision of the predictions. By comparing the predicted outputs to the actual measurements, we can clearly see the improvements in linearity. The model's adaptability and accuracy in compensating for non-linearities across various conditions underscore its robustness and reliability for practical applications.

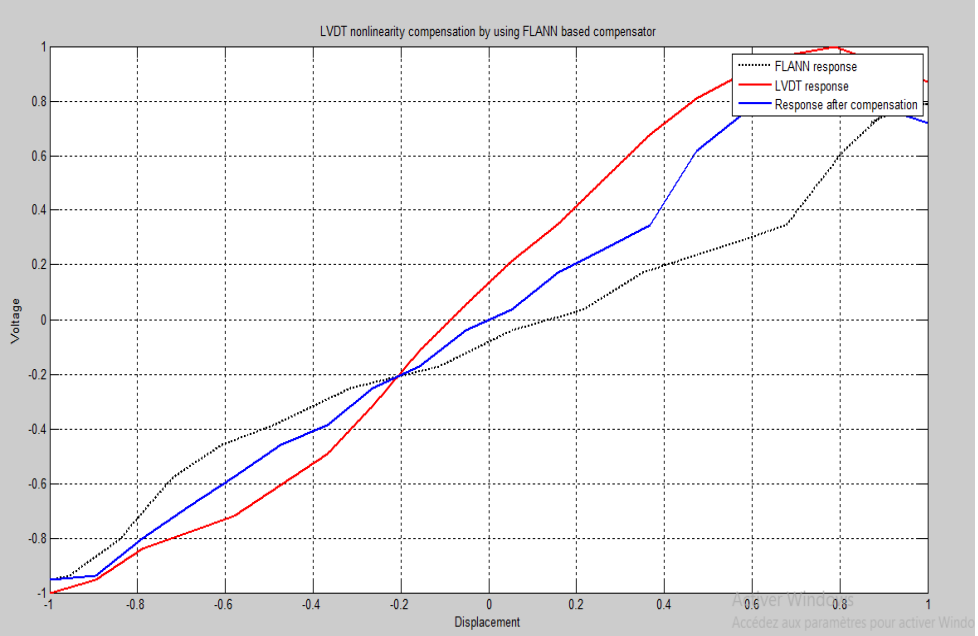

Figure II-11:Prediction for training points when P=10.

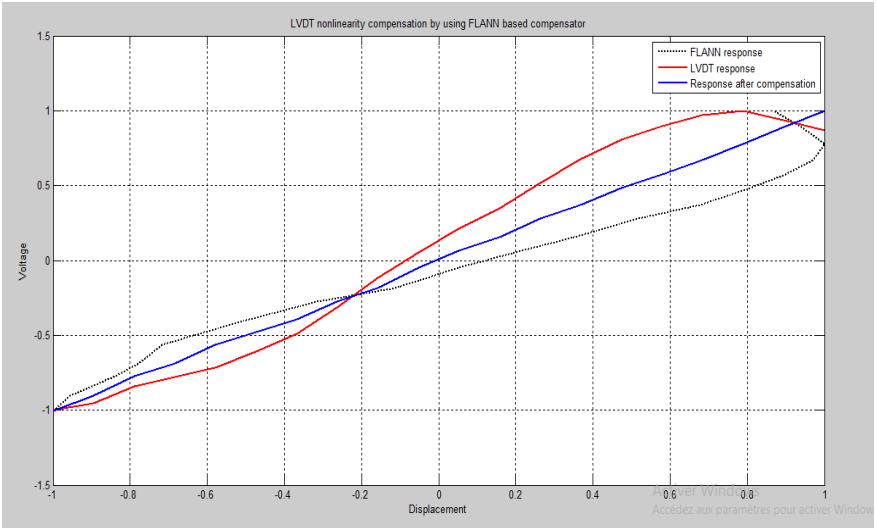

Figure II-12:Prediction for training points when P=50.

However, when we made predictions for other points that were not part of the initial training dataset, the results varied significantly, as shown in Table II-3. This discrepancy indicates that while the FLANN model is highly effective within the range of trained points, its performance can differ when applied to untrained data points. This highlights the importance of extensive training across a wide range of values to ensure consistent performance and accuracy in practical applications.

Expansion number	Percentage of nonlinearity (%)
P=10	6.61
P=50	104.915

Table II-4: Prediction of testing points.

Figure II-(13-14) represent the results for the testing points (unseen data).

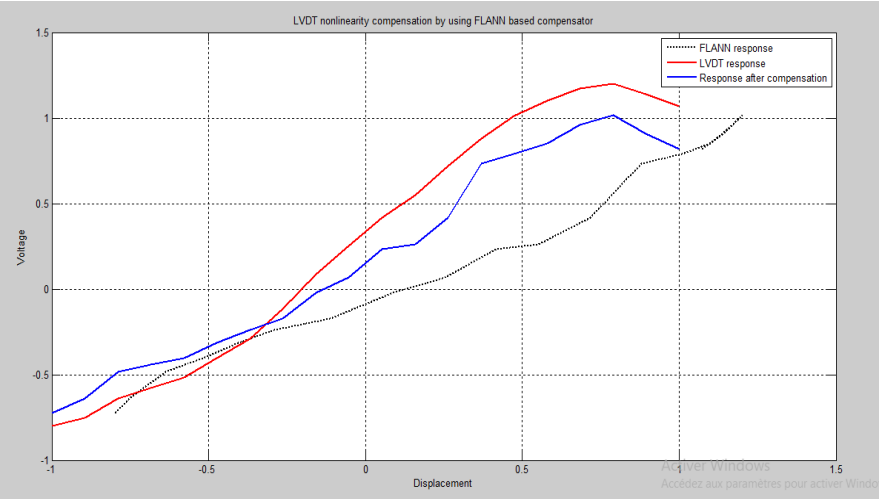

Figure II-13:Prediction for unseen points when P=10.

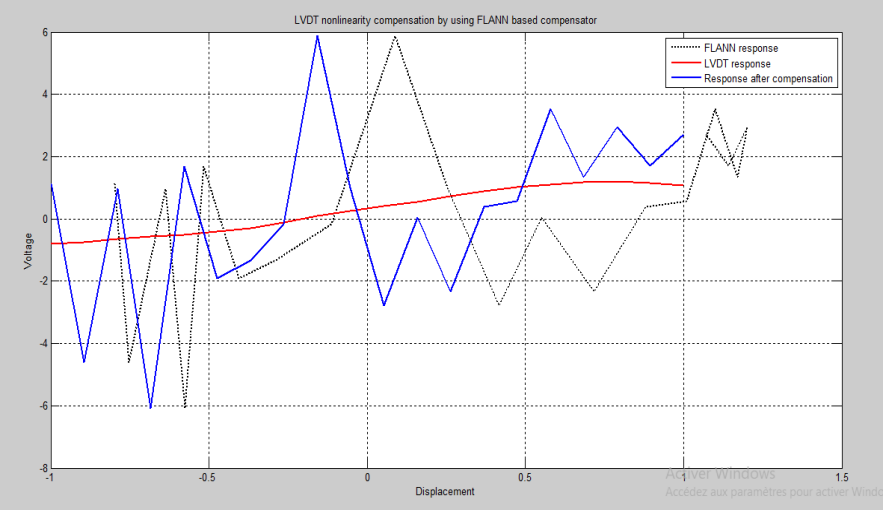

Figure II-14:Prediction for unseen points when P=50.

II.6.1. Interpretation of Results

In this section, we interpret the results obtained from our experiments with the FLANN model. While our FLANN model demonstrated significant improvements in compensating for non-linearity in the LVDT sensor for the trained points, it exhibited a problem of generalization when predicting unseen points.

Another crucial observation is the impact of increasing the parameter P on the model's performance. For the trained points, increasing the number of P results in nearly perfect predictions, as evidenced by the increasingly accurate results shown in Figure II-(11-12). This indicates that the FLANN model can effectively learn and predict within the confines of its training data when given more parameters.

Conversely, for untrained points, increasing the number of P results in worse predictions, as shown in Figure II-(13-14). This phenomenon suggests that the model, when provided with more parameters, tends to overfit to the training data, thus losing its ability to generalize to new unseen data. This limitation is a critical consideration for the practical deployment of the model, necessitating further research and enhancement to improve its generalization capabilities. By addressing this issue, we can develop a more robust and versatile model that performs reliably across a broader spectrum of conditions.

In summary, while increasing P improves performance for trained points, it highlights the model's generalization problem for untrained points. This balance between fitting the training data and generalizing to new data is crucial for the effective application of the FLANN model in real-world scenarios.

II.7. Conclusion

In conclusion, Artificial Neural Networks (ANNs) represent a significant advancement in computational technology, capable of addressing a wide array of complex tasks through their unique architectures and learning algorithms.

Additionally, specialized networks like Functional Link Artificial Neural Networks (FLANN) enhance the ability to model complex relationships with reduced computational demands. The effectiveness of ANNs is largely attributed to their learning processes.

The results of application of the FLANN to the LVDT problem discussed in this chapter underscore the robust performance of FLANN in non-linearities compensation, while also pointing to areas where further research and development could enhance their generalization capabilities and overall effectiveness.

Chapter.III. FLANN-based level sensors calibration and implementation

III.1. Introduction

This chapter explores the application of Functional Link Artificial Neural Networks (FLANN)for the intelligent calibration of level sensors, both with and without the integration of Piecewise Linear (PWL) interpolation. FLANN offers promising capabilities in handling sensor non-linearities, while PWL interpolation further enhances its adaptability. Through a comparative analysis of calibration results and model validation, this chapter aims to elucidate the efficacy of FLANN with and without PWL interpolation in achieving precise and reliable sensor calibration.

III.2. Experimental setup presentation

An experimental setup used for controlling a 3-tank system has been shown in Figure III-1, typically found in process control and systems engineering applications. This setup includes several key components and their respective functions. The PC (Personal Computer) features a data screen that displays real-time data from the system, such as tank levels and control signals. An A/D-D/A converter converts analog signals from the plant's sensors into digital signals for processing and vice versa, enabling communication between the PC and the plant. Although not shown in detail, a plotter is indicated for creating physical plots of the data for analysis.

The actuator section consists of a servo amplifier, a disturbance module, and a signal matching unit. The servo amplifier receives control signals from the PC and amplifies them to drive the pumps. The disturbance module introduces controlled disturbances into the system to test its response and robustness. The signal matching unit ensures that the control signals are compatible with the input requirements of the plant's components, such as sensors and pumps.

The plant section includes two pumps (Q1 and Q2) that control the flow of liquid into the 3-tank system, with adjustable flow rates based on control signals. The 3-tank system is composed of three interconnected tanks where the levels of each tank (h1, h2, h3) are monitored and controlled. Sensors measure the liquid levels in each tank and send this information back to the PC for monitoring and control purposes.

The signal flow within this setup operates as follows:

PC to Actuator: The PC processes data and generates control signals, which are sent to the servo amplifier via the A/D-D/A converter. The servo amplifier adjusts the signal strength and sends it to the pumps in the plant.

Actuator to Plant: The output from the servo amplifier controls the pumps (Q1 and Q2), which adjust the flow rates into the 3-tank system. The disturbance module can inject disturbances into the system to simulate real-world variability and test the system's response.

Plant to Actuator and PC: Sensors in the 3-tank system measure the levels (h1, h2, h3) and send this data back to the signal matching unit. The signal matching unit ensures the sensor signals are correctly formatted and sends them to both the servo amplifier for real-time adjustments and the PC for monitoring and further control adjustments.

Data Monitoring and Feedback: The PC continuously receives sensor data, displays it on the data screen, and makes necessary adjustments to the control signals based on the feedback. This closed-loop system ensures that the liquid levels in the tanks are maintained as desired, despite any disturbances introduced.

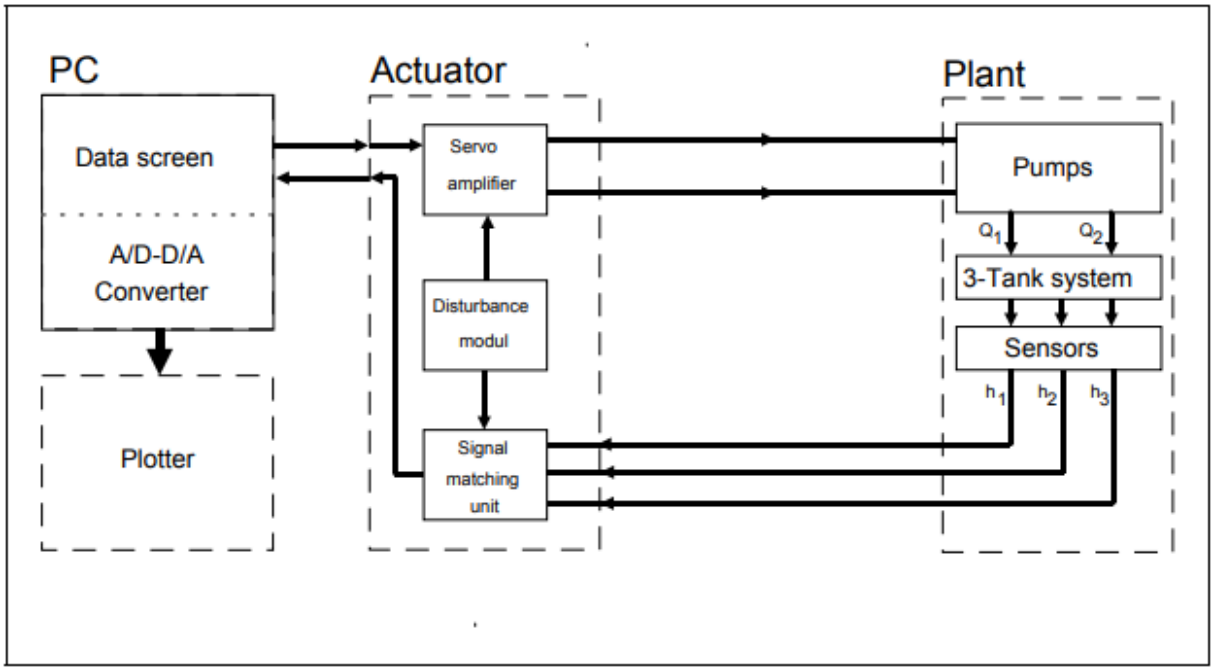

Figure III-1:Closed-loop control System for a 3-tank process with disturbance Injection

III.2.1. Three-Tank-System

The System "Three-Tank-System"(Figure III- 2) consits of three plexiglas cylinders with the cross section. These are connected serially with each other by cylindrical pipes with a cross section. The outflowing liquid (usually distilled water) is collected in a reservoir, supplies the pumps 1 and 2. Here the circle is closed. [38]

Figure III-2:Three-Tank-System

III.2.1.1. Level sensor overview

Necessary level measurements are carried out by piezo-resistive difference pressure sensors VEGABAR 14 which is a process pressure sensor designed for P measurement of gas, steam and liquid products. The CERTEC cell with its robust ceramic membrane constitutes the measuring element. The process pressure leads to a variation of capacity in the cell via the ceramic membrane. This signal is converted into an electrical signal.

The mesured pressure can be used to determine the liquid level in a tank. The pressure at the bottom of a liquid filled container is directly related to the height of the liquid. The sensor measures this hydrostatic head pressure and gives the resulting liquid level. To get an accurate reading, the measurement device needs to be located at the lowest point you want to measure; typically mounted or laying on the bottom of the container. When measuring liquid level, specific gravity must be taken into account. [39]

Consider the following equation: $\boldsymbol{H = \frac{P}{SG}}$ where :

H : Height of the liquid being measured.

P : Hydrostatic head pressure at the bottom of the tank.

SG : Media's specific gravity.

III.2.1.2. Simulink calibration platform

This Simulink diagram shown in Figure III-3 represents an experimental setup for controlling a system, likely involving multiple tanks and sensors, interfaced with an Arduino. The model includes several components with specific functions. The Arduino Analog Read blocks are used three times to read analog signals from sensors connected to the Arduino.

These signals are then combined with step input signals using Product blocks, creating modified signals for testing purposes.

The signals are subsequently scaled using Gain blocks (Gain3, Gain4, and Gain5) and shifted using Add blocks, which sum up the scaled signals with constant values (provided by Constant, Constant1, and Constant2 blocks). To ensure clean signals for display and further processing, the model employs Filter blocks (Filter, Filter1, and Filter2). The filtered signals are then displayed in real-time using Display blocks (Display6, Display7, and Display8), and also sent to the MATLAB workspace for further analysis and logging through the To Workspace blocks (To Workspace, To Workspace1, and To Workspace2).

Control signals are generated from the processed signals and adjusted using Slider Gain blocks, which allow manual tuning through a slider interface. The initial gain values for these control signals are set by Constant3 and Constant4 blocks. The adjusted control signals are then sent back to the Arduino via Arduino Analog Write blocks to drive actuators, such as pumps.

Additionally, the model includes a Clock block that generates a time signal for synchronization purposes, which is displayed in real-time using Display5. The Arduino IO Setup block configures the Arduino input/output settings, and the Real-Time Pacer block ensures the Simulink model runs in real-time, aligning the actual time progression with the simulation time.

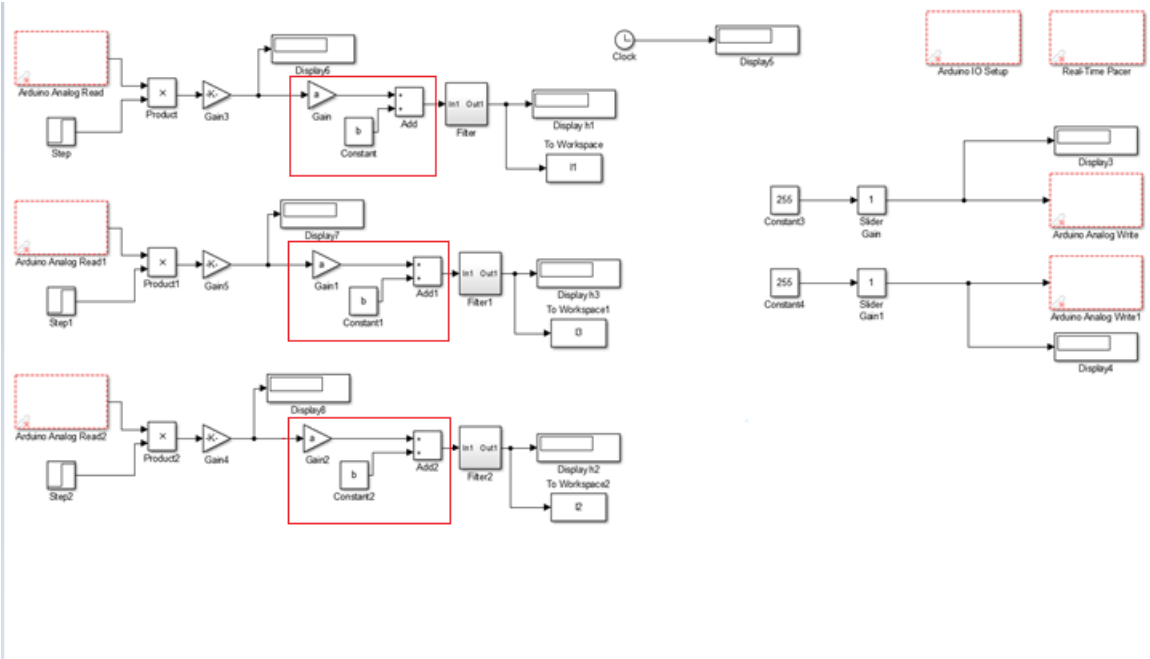

Figure III-3:Real-Time Control System for a Three-Tank Setup Using Arduino and Simulink

The blocks inside the red square represent classic calibration used to calibrate the sensor signals. The method used in these blocks is simple linear regression, which aims to find the best-fitting straight line (called the regression line) through the data points by minimizing the sum of the squares of the vertical distances of the points from the line. This line can then be used to predict the tank levels based on new values of the sensor output signals, following the equation: $output = a \times input + b$

This method has several limitations, particularly when the sensor characteristics exhibit significant nonlinearities. In such cases, using simple linear regression can lead to substantial errors. To address this, nonlinear compensation methods are required to more accurately model and correct for these nonlinearities in the sensor output.

We improve this model by using intelligent calibration based on the FLANN (Functional Link Artificial Neural Network) model, which provides effective compensation for nonlinearities.

III.2.2. Data collection

We gradually filled the three tanks one at a time, recording the corresponding sensor output voltage value at each level point, (15mm to 585mm with a step = 10mm).

We obtained an input-output dataset containing 58 elements. The data obtained by conducting experiments on the three level sensors are given in Tables III-1, III-2 and III-3.

Sensor 1 input level (mm)	Sensor 1 output voltage (V)
15	0.960
25	1.006
35	1.045
45	1.089
55	1.120
65	1.172
75	1.235
85	1.270
95	1.320
105	1.345
115	1.400
125	1.470
135	1.480
145	1.528
155	1.570
165	1.600
175	1.650
185	1.714
195	1.748

205	1.807
215	1.840
225	1.885
235	1.924
245	1.970
255	2.017
265	2.086
275	2.100
285	2.173
295	2.188
305	2.230
315	2.271
325	2.329
335	2.368
345	2.407
355	2.440
365	2.500
375	2.549
385	2.588
395	2.670
405	2.666
415	2.730
425	2.770
435	2.800
445	2.850
455	2.886
465	2.954
475	3.000
485	3.050
495	3.096
505	3.105

515	3.150
525	3.188
535	3.270
545	3.286
555	3.315
565	3.345
575	3.418
585	3.477

Table III-1: Input-output dataset of tank 1

Sensor 2 input level (mm)	Sensor 2 output voltage (V)
15	0.950
25	1.025
35	1.084
45	1.120
55	1.187
65	1.216
75	1.284
85	1.304
95	1.305
105	1.350
115	1.380
125	1.514
135	1.520
145	1.590
155	1.680
165	1.650
175	1.690
185	1.740
195	1.900
205	1.820
215	1.940

225	1.890
235	1.970
245	1.971
255	2.021
265	2.060
275	2.119
285	2.212
295	2.180
305	2.275
315	2.324
325	2.350
335	2.368
345	2.420
355	2.466
365	2.510
375	2.490
385	2.500
395	2.550
405	2.600
415	2.640
425	2.670
435	2.720
445	2.750
455	2.800
465	2.830
475	2.880
485	2.935
495	2.970
505	3.040
515	3.070
525	3.105

535	3.145
545	3.200
555	3.230
565	3.293
575	3.345
585	3.374

Table III-2: Input-output dataset of tank 2

Sensor 3 input level (mm)	Sensor 3 output voltage (V)
15	1.000
25	0.990
35	1.020
45	1.060
55	1.120
65	1.160
75	1.190
85	1.230
95	1.280
105	1.330
115	1.370
125	1.410
135	1.460
145	1.500
155	1.550
165	1.600
175	1.650
185	1.700
195	1.750
205	1.800
215	1.850
225	1.890

235	1.920
245	1.950
255	2.000
265	2.050
275	2.100
285	2.140
295	2.120
305	2.150
315	2.200
325	2.230
335	2.260
345	2.310
355	2.350
365	2.410
375	2.440
385	2.490
395	2.540
405	2.590
415	2.630
425	2.680
435	2.720
445	2.750
455	2.800
465	2.860
475	2.900
485	2.940
495	3.000
505	3.040
515	3.080
525	3.120
535	3.150

545	3.200
555	3.240
565	3.290
575	3.311
585	3.380

Table III-3: Input-output dataset of tank 3

We split the data into training and testing sets, allocating 75% for training and 25% for testing. It is imperative not to use any elements from the test set during training, as this will result in a completely inaccurate neural model. The test set is reserved solely for the final measurement of the model's performance. In other words, it is used to verify if the neural network performs well on examples it has not learned, the "test set." However, there is always a risk of overfitting when using this technique. Overfitting occurs when the network learns too many parameters to model a function that is not very complex. This will lead to an overfitting problem, which manifests as an increase in error on the validation set [37].

III.2.3. Intelligent calibration system design

The use of neural networks as calibration model, to correct the response of sensors, has grown considerably in recent years. Indeed, ANNs have the advantage of a great adaptation to the various problems caused by the non-ideality of sensors [39].

Calibration using ANN-based adaptive inverse model (Intelligent calibration) is basically finding the “best” approximation inverse function $x = f^{-1}(v)$ and the overall process could then be called the linearising calibration process. The inverse transfer function is then calculated so that input signal can be derived from the sensor output.

For the development of the INV-ANN model, we used a simple FLANN-based inverse model that involves quite less computational complexity and is quite suitable for real-time implementation. The learning scheme of the intelligent calibration is illustrated in Figure III-4.

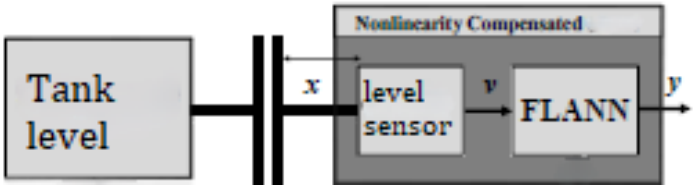

Figure III-4:The learning scheme of the intelligent calibration system

III.2.3.1. FLANN model

In this work, we have used trigonometric expansion because it provides better nonlinearity compensation compared to other types of expansion. The FLANN is subjected to trigonometric expansion to achieve nonlinearly mapped values. These are then multiplied with a set of weights and then added to produce the inverse model output. It is compared with the desired signal (real level values) to produce an error signal. With this error signal, the weight vector of the FLANN model is updated. The training process is repeated until the MSE is minimized to a certain level, from where it cannot further be decreased. [40] The structure of the FLANN is shown in Figure III-5.

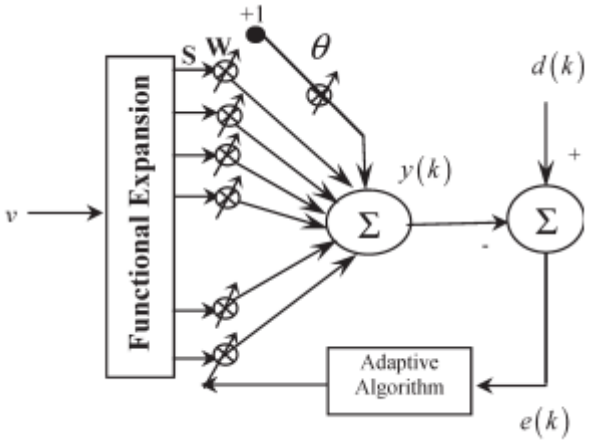

Figure III-5:The learning scheme of the intelligent calibration system

Let V be the input vector of N elements. Let the net configuration have one output. Each element undergoes trigonometric nonlinear expansion to form P elements such that the resultant matrix has the dimension of N × P.

Since the nth input element is v_n, $1 \leq n \leq$ N, the functional expansion is carried out as :

$$s_i = \begin{cases} v_n, & i = 1 \\ \sin(l\pi v_n), & i > 1, i \text{ is even} \\ cos(l\pi v_n) & i > 1, i \text{ is odd} \end{cases} \quad \text{(III.1)}$$

where l = 1, 2,...,P/2, and i = 1, 2,...,P + 1.

In the proposed expansion, the total number of expansion P is always an even number.

In matrix notation, the expanded elements of the input vector **V** is denoted by **S** of size N × (P + 1). An extra unity value (for bias) is padded with the **S** matrix, and hence, the dimension of S matrix becomes N × Q, where Q = (P + 2).

Let the weight vector is represented as **W** having Q elements. The output, y is given as :

$$y = \sum_{i=1}^{Q} w_i s_i \tag{III.2}$$

In matrix notation, the output can be

$$\mathbf{Y} = \mathbf{S} \cdot \boldsymbol{W^T} \tag{III.3}$$

At the k*th* iteration, the error signal e(k) can be computed as :

$$e(k) = d(k) - y(k) \tag{III.4}$$

where d(k) is the desired signal.

Let ξ(k) denote the cost function at iteration k and be given by

$$\Xi(\mathrm{k}) = \frac{1}{2}\sum_{j\in P} e_j^2\,(k) \tag{III.5}$$

- **Gradient-based training of the model**

The gradient descent (GD) is the simplest training algorithm. The weight vector can be updated by the GD algorithm as :

$$w(k+1) = w(k) - \frac{\mu}{2}\widehat{\nabla}(k) \tag{III.6}$$

where $\widehat{\nabla}(k)$ is an instantaneous estimate of the gradient of ξ with respect to the weight vector w(k).

Now :

$$\begin{aligned}\widehat{\nabla}(k) &= \frac{\partial \xi}{\partial \mathrm{w}} = -2\mathrm{e(k)}\,\frac{\partial \mathrm{y(k)}}{\partial \mathrm{w}} \\ &= -2\mathrm{e(k)}\,\frac{\partial\,[\mathrm{w(k)s(k)}]}{\partial \mathrm{w}} \\ &= 2\mathrm{e(k)s(k)}.\end{aligned} \tag{III.7}$$

By substituting the values of $\widehat{\nabla}(k)$ in (5), we get :

$$\mathrm{w(k+1) = w(k) + \mu e(k)s(k)} \tag{III.8}$$

where μ denotes the step size ($0 \le \mu \le 1$), which controls the convergence speed of the GD algorithm.

III.3. Simulation results and discussion

III.3.1. Gradient-based training of the model

The simulation study of Gradient-based training of the FLANN model for each sensor (level sensor 1, level sensor 2 and level sensor 3) has been undertaken using the experimental datasets shown in the previous sections (table III-1, table III-2 and table III-3) for both training and testing data.

The simulation is carried out in the MATLAB 8. 1 environment.

The observed simulation results are shown in various figures listed in Table III-4 Various FLANN models of Table 4 are simulated, and the responses of the models and the overall responses are obtained through simulation. These results, as indicated in Table 4, are shown in Figure III- (6–11). The percentage of nonlinearity defined as :

$$Percentage\ of\ Nonlinearity = \frac{Max\ deviation\ from\ the\ idealized\ straight\ line}{\text{Full scale deviation}} \times 100$$

Device	Model used	Data set	Specifications	Results in figures	Percentage of nonlinearity
Level sensor 1	FLANN-GD	Training data	P = 120	Fig III-6	0.002
Level sensor 2			P =120	Fig III-7	0.968
Level sensor 3			P =120	Fig III-8	0.168
Level sensor 1		Testing data	P = 120	Fig III-9	127.918
Level sensor 2			P =120	Fig III-10	211.075
Level sensor 3			P =120	Fig III-11	106.424

Table III-4 :Simulation results of level sensors calibration

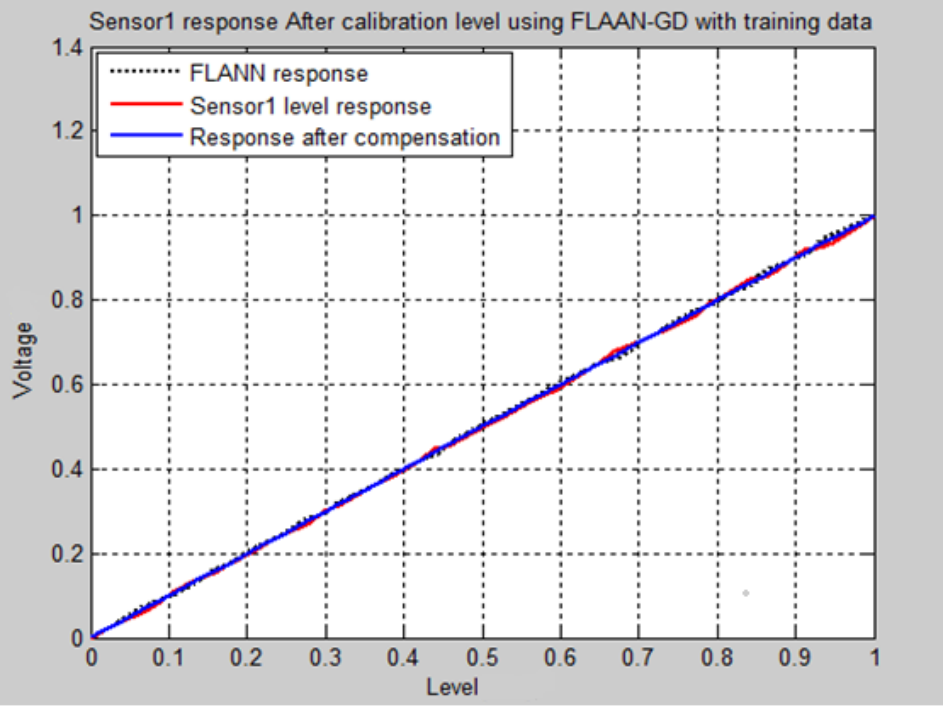

Figure III-6:Sensor 1 response after level calibration using FLANN-GD with training data

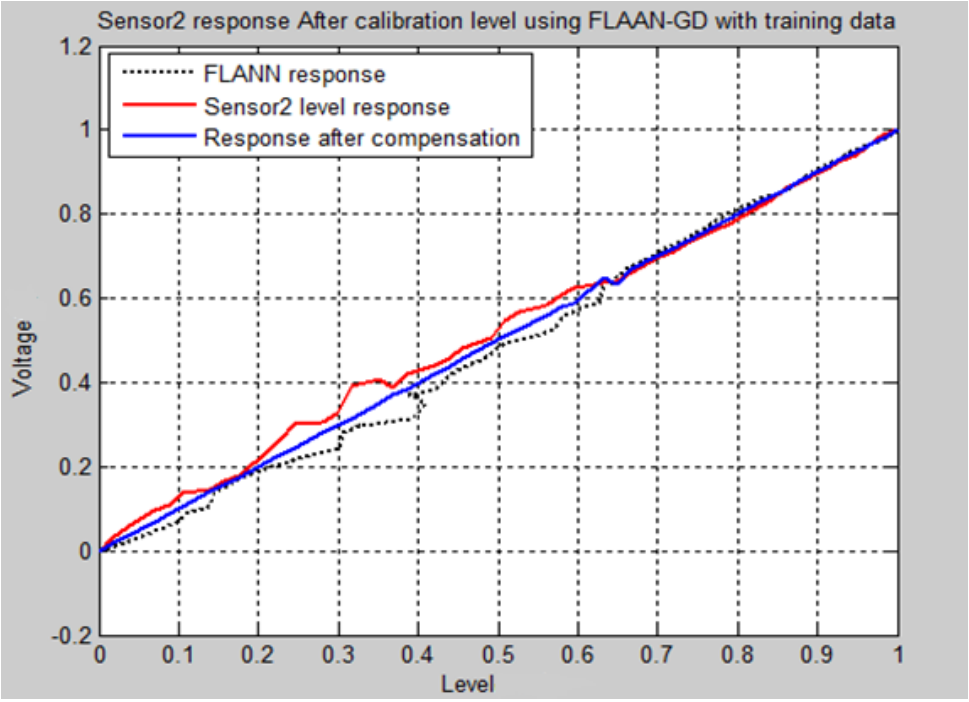

Figure III-7:Sensor 2 response after level calibration using FLANN-GD with training data

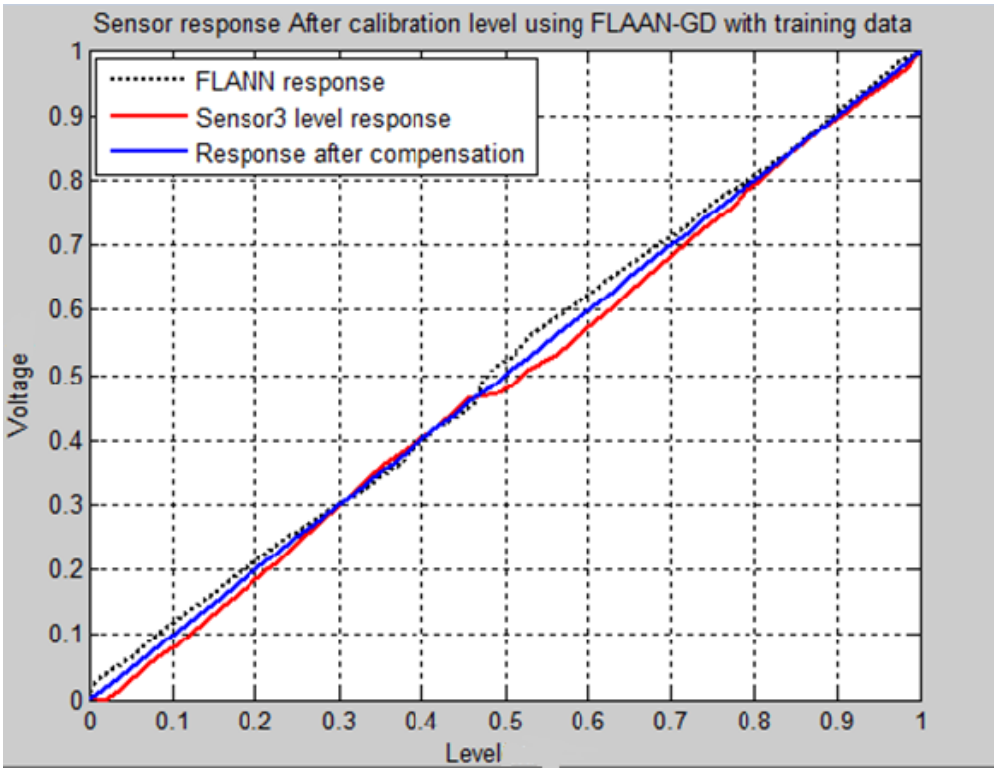

Figure III-8:Sensor 3 response after level calibration using FLANN-GD with training data

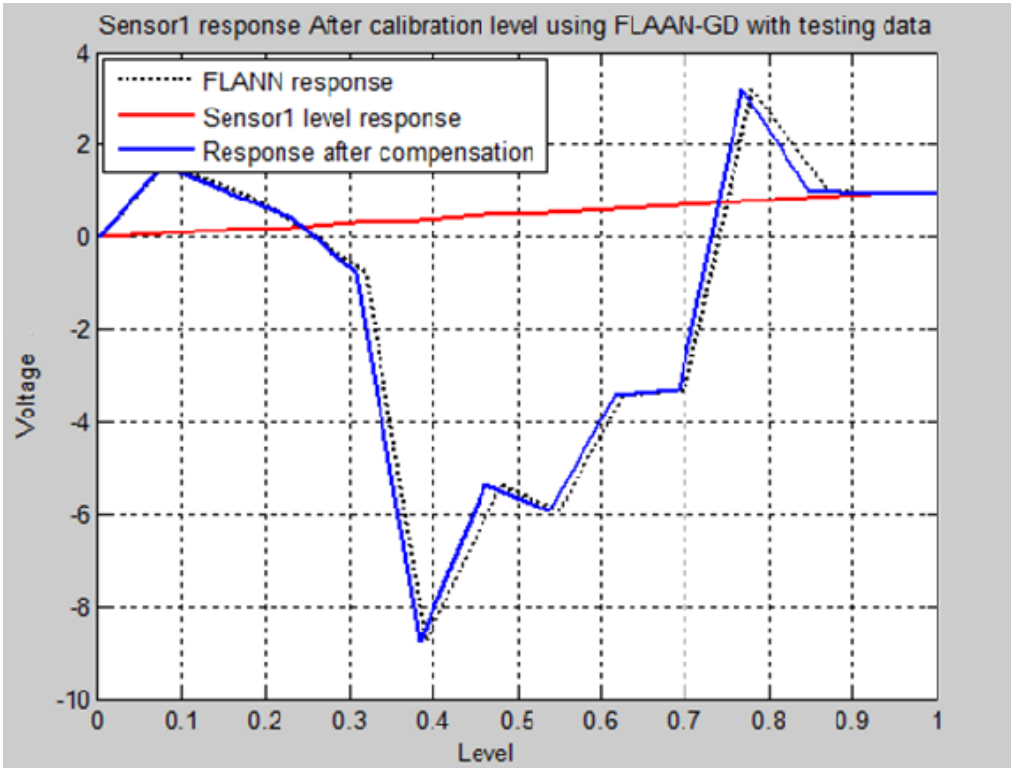

Figure III-9: Sensor 1 response after level calibration using FLANN-GD with testing data

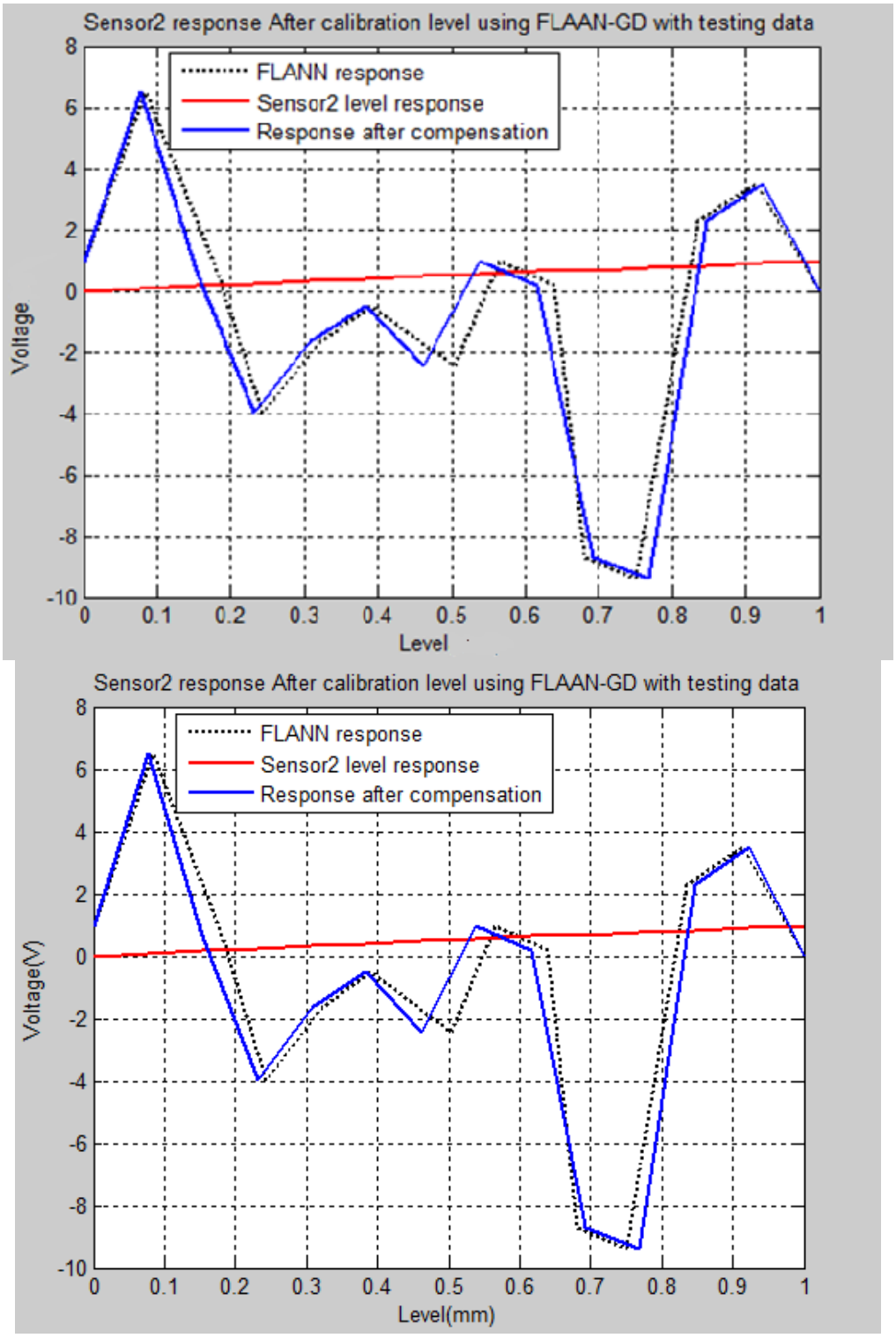

Figure III-10:Sensor 2 response after level calibration using FLANN-GD with testing data

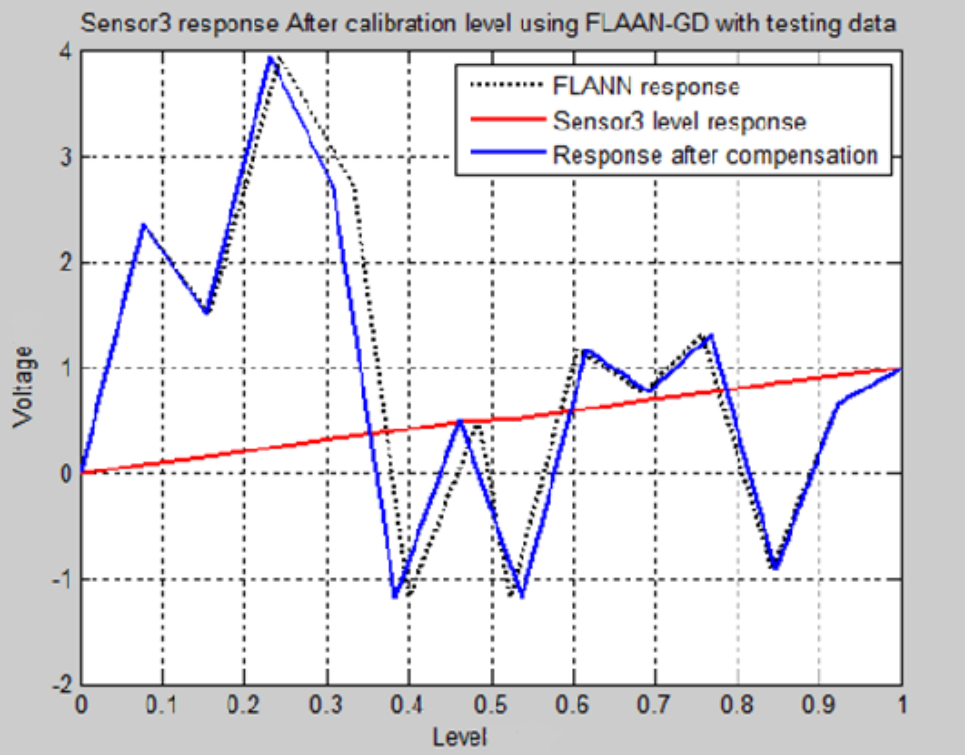

Figure III-11:Sensor 3 response after level calibration using FLANN-GD with testing data

Like we have seen in our results the FLANN model demonstrated improved compensation for non-linearity in the LVDT sensor for trained points but struggled to generalize results for both training and testing data. Increasing model complexity led to overfitting, highlighting the need for further research to enhance its generalization capabilities for practical deployment.

III.3.2. FLANN model with piecewise linear interpolation

In this method, the sensor output is measured for a small set of known input signals. These calibration points become anchor points or “break” points on the ideal inverse transfer curve. The inverse transfer curve is completed by joining the “break” points with straight lines. These lines form a piecewise linear interpolation of the ideal inverse sensor transfer curve (Figure III-12)

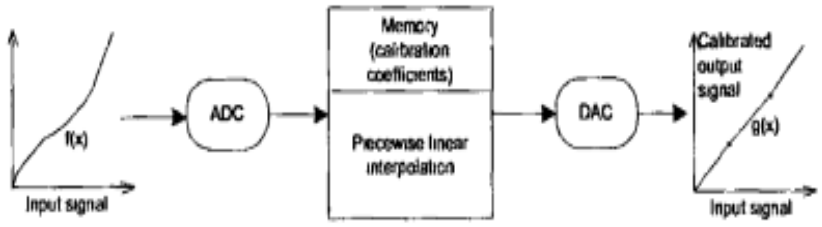

Figure III-12:Piecewise linear (PWL) calibration method

Using the PWL interpolation with the empirical data collected in experiences will increase the size of training dataset. That will help to avoid the problem of bad generalization.

III.3.3. Gradient-based training of the model with PWL interpolation

The simulation results of Gradient-based training of the FLANN model using PWL interpolation are shown in various figures listed in Table III-5.

Device	Model used	Data set	Specifications	Results in figures	Percentage of nonlinearity
Level sensor 1	FLANN-GD With PWL Interpolation	Training data	P = 120	Fig III-13	13.006
Level sensor 2			P = 120	Fig III-14	12.285
Level sensor 3			P = 120	Fig III-15	6.27
Level sensor 1		Testing data	P = 120	Fig III-16	11.292
Level sensor 2			P = 120	Fig III-17	5.128
Level sensor 3			P = 120	Fig III-18	3.732

Table III-5 :Results of Gradient -based training of the FLANN model using PWL interpolation

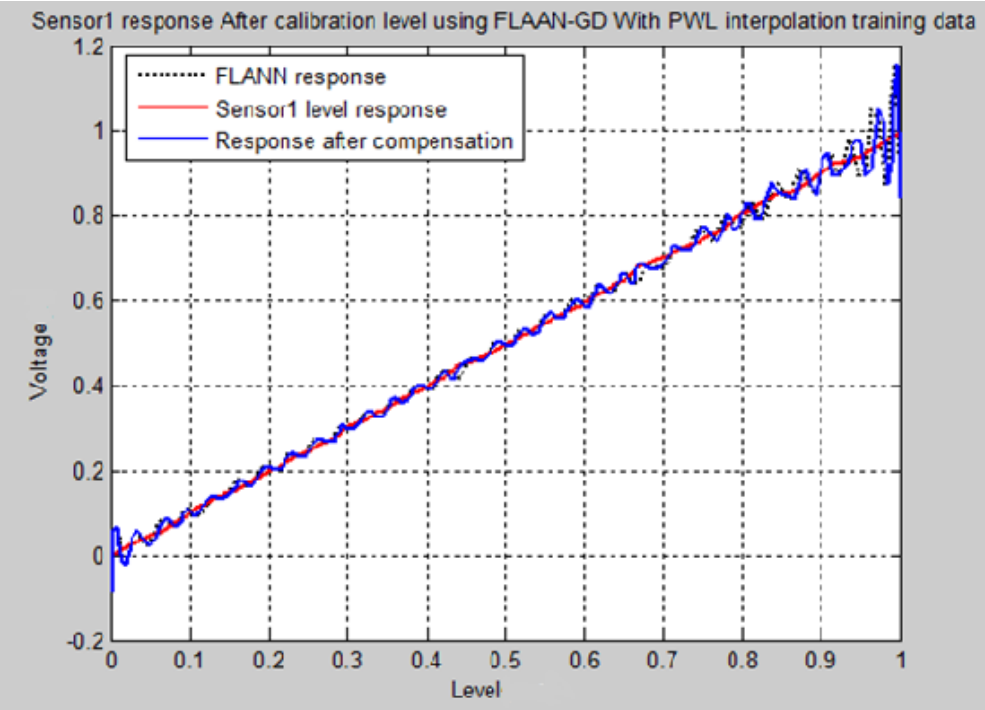

Figure III-13:Sensor 1 response after level calibration using FLANN-GD with PWL interpolation with training data

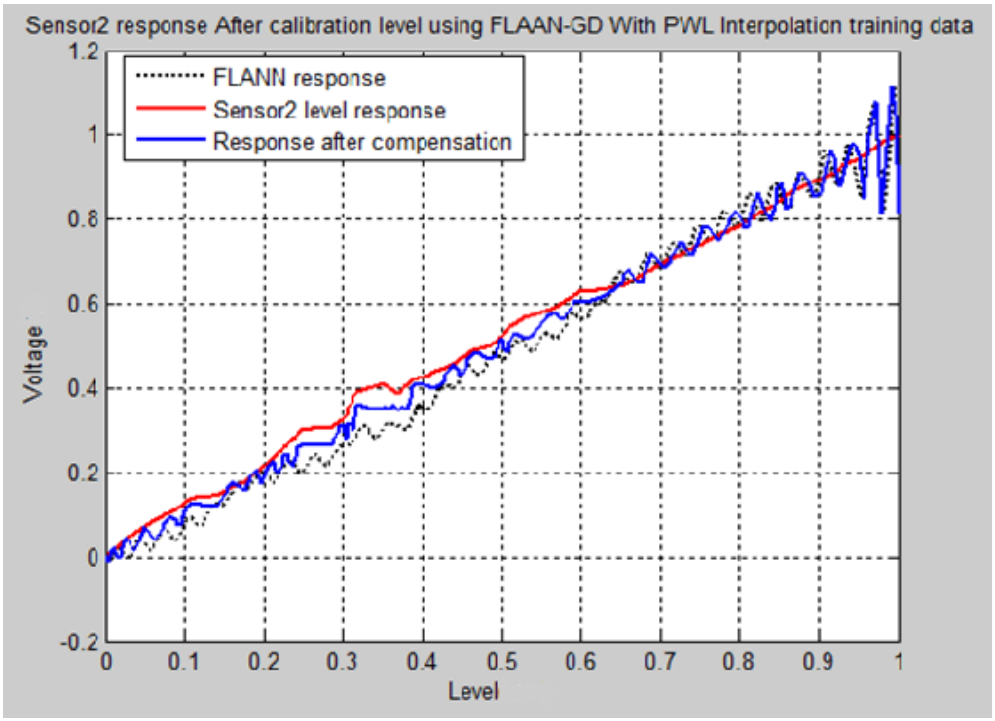

Figure III-14:Sensor 2 response after level calibration using FLANN-GD with PWL interpolation with training data

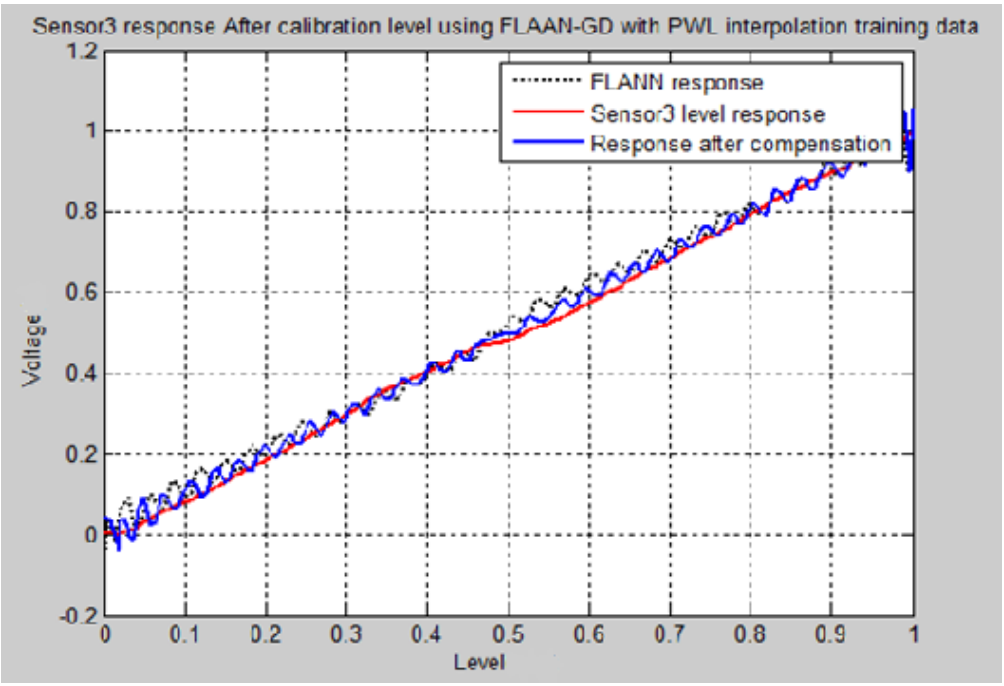

Figure III-15:Sensor 3 response after level calibration using FLANN-GD with PWL interpolation with training data

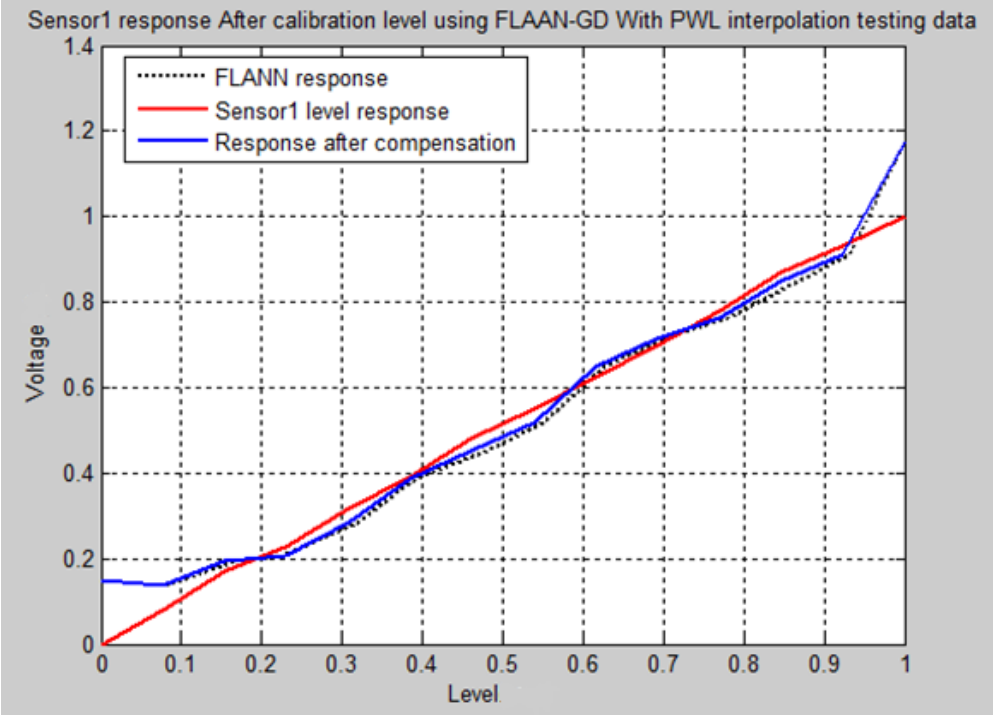

Figure III-16:Sensor 1 response after level calibration using FLANN-GD with PWL interpolation with testing data

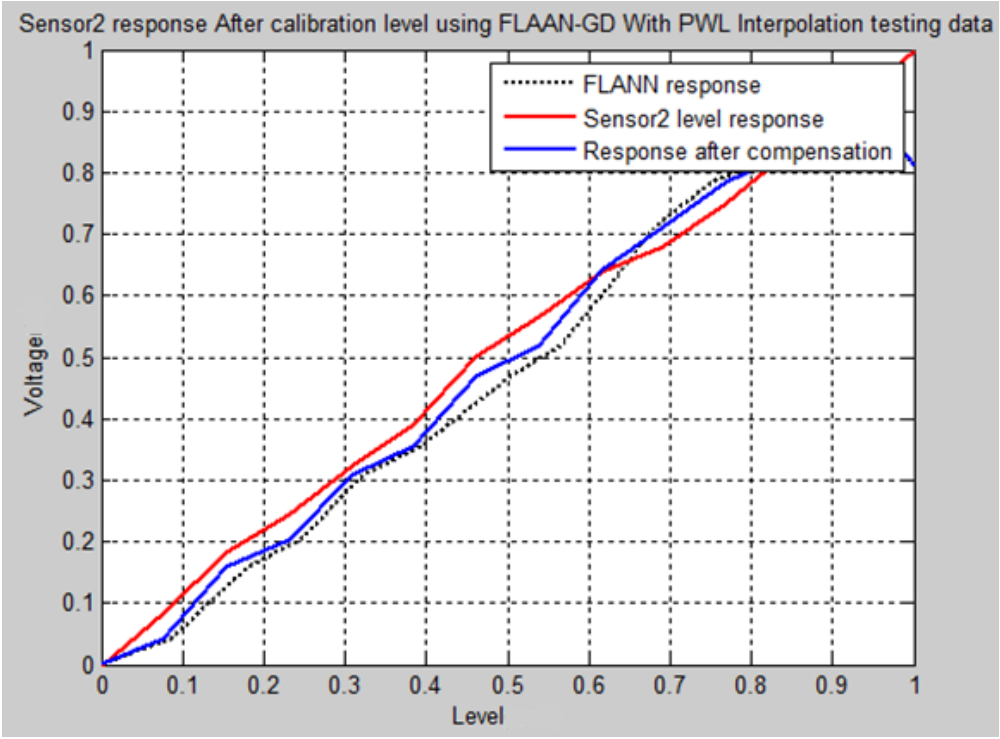

Figure III-17:Sensor 2 response after level calibration using FLANN-GD with PWL interpolation with testing data

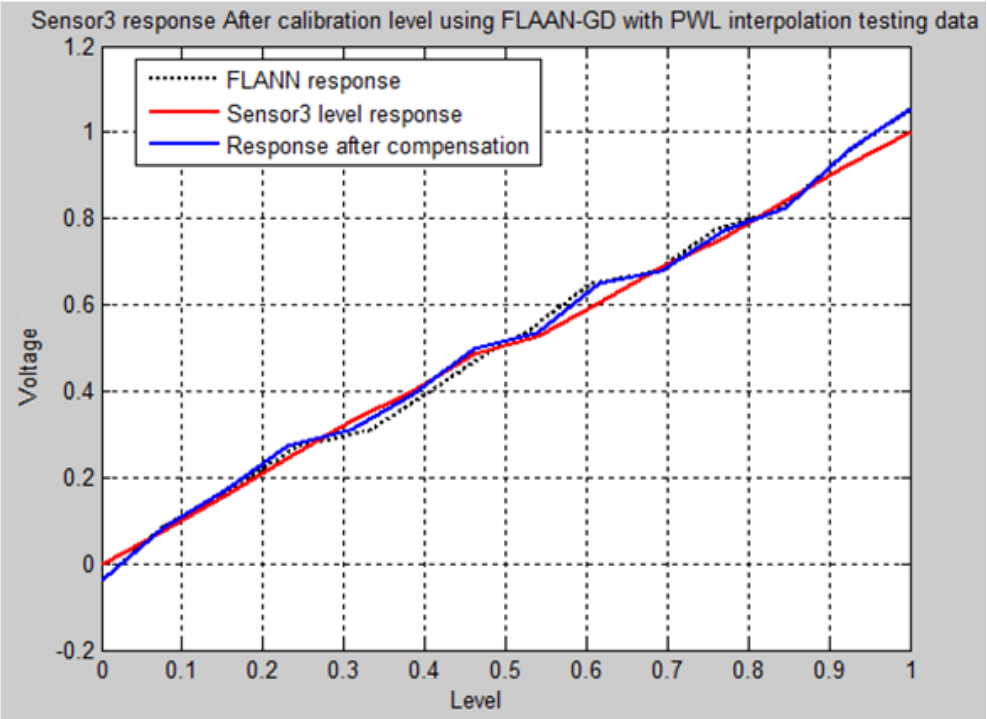

Figure III-18:Sensor 3 response after level calibration using FLANN-GD with PWL interpolation with testing data

The results from Figure III(13-18) showed a marked improvement with the implementation of Piecewise Linear (PWL) calibration. This approach effectively addressed the generalization problem that the FLANN model initially faced, enabling the model to perform reliably across both training and testing data. The enhanced performance indicates that PWL calibration offers a robust solution for improving the generalization capabilities of the FLANN model, ensuring consistent and accurate predictions even with previously unseen data. This development highlights the potential of PWL calibration to enhance the practical applicability of the FLANN model in real-world scenarios.

III.4. Real-time implementation

The implementation of our new model, replacing the classic model (Figure III-19), for intelligent calibration is carried out in two experiences, without and with PWL Interpolation, then according to the results we validated the appropriate model.

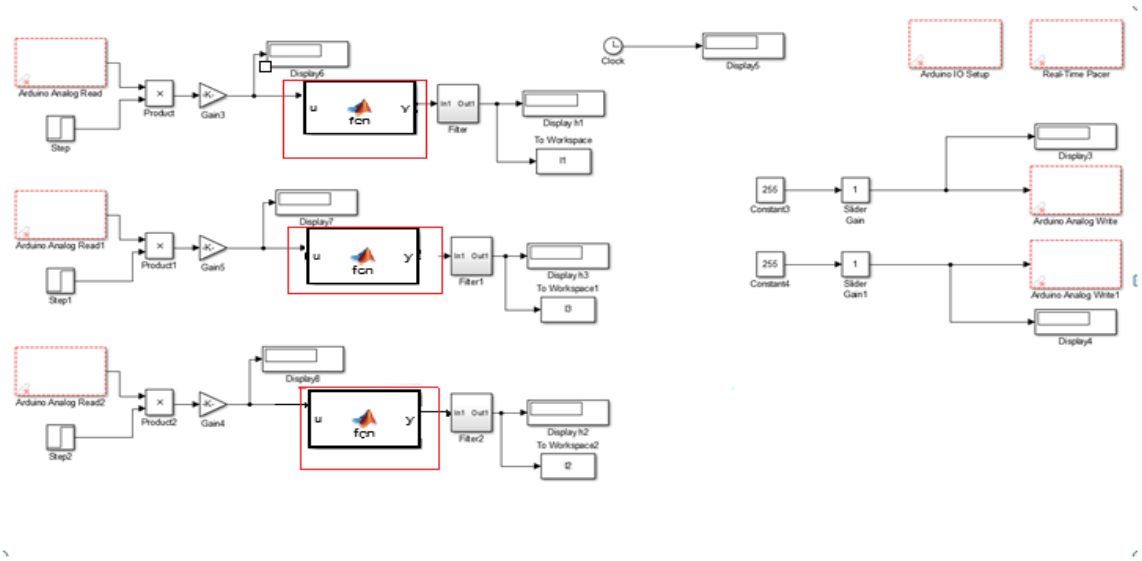

Figure III-19:Real-Time Control System for a three-Tank Setup Using Arduino and Simulink with our new model

III.4.1. FLANN model without piecewise interpolation results

In the first implementation we used the FLANN-GD model without PWL interpolation and compare its response with the perfect response representing the error (mean error) between them. The results are shown in various figures listed in Table III-6.

Device	Model used	Specifications	Results in figures	Mean error (mm)
Level sensor 1	FLANN-GD Without PWL Interpolation	P = 120	Fig III-20	1856.3
Level sensor 2		P = 120	Fig III-21	598.7
Level sensor 3		P = 120	Fig III-22	1260.1

Table III-6: Results using FLANN-GD without PWL interpolation

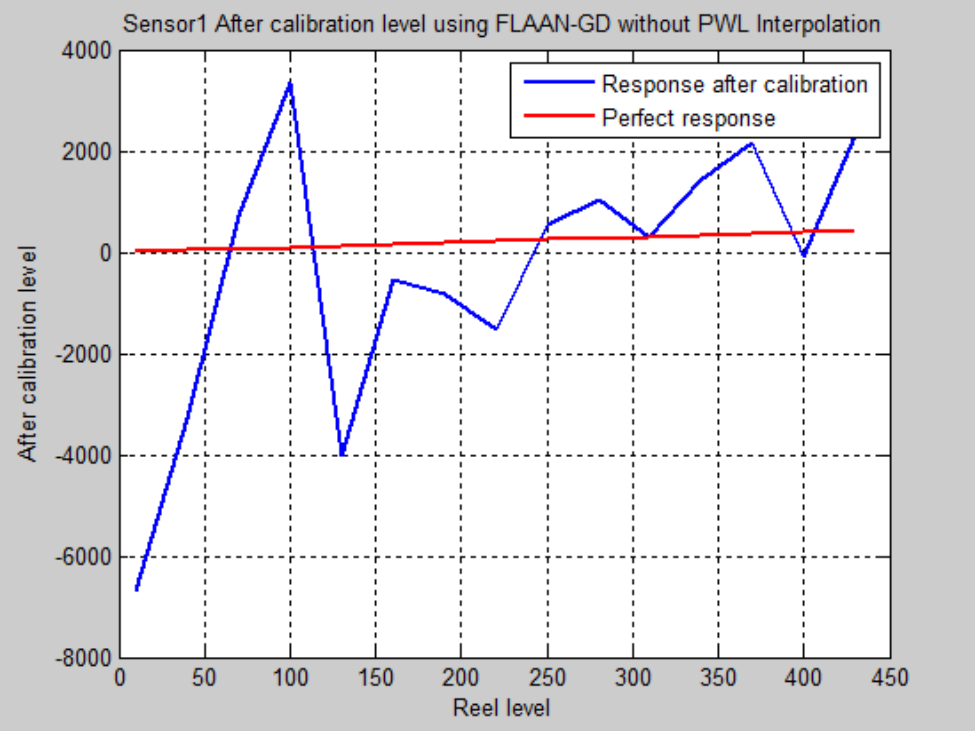

Figure III-20:Sensor1 after level calibration using FLANN-GD without PWL interpolation

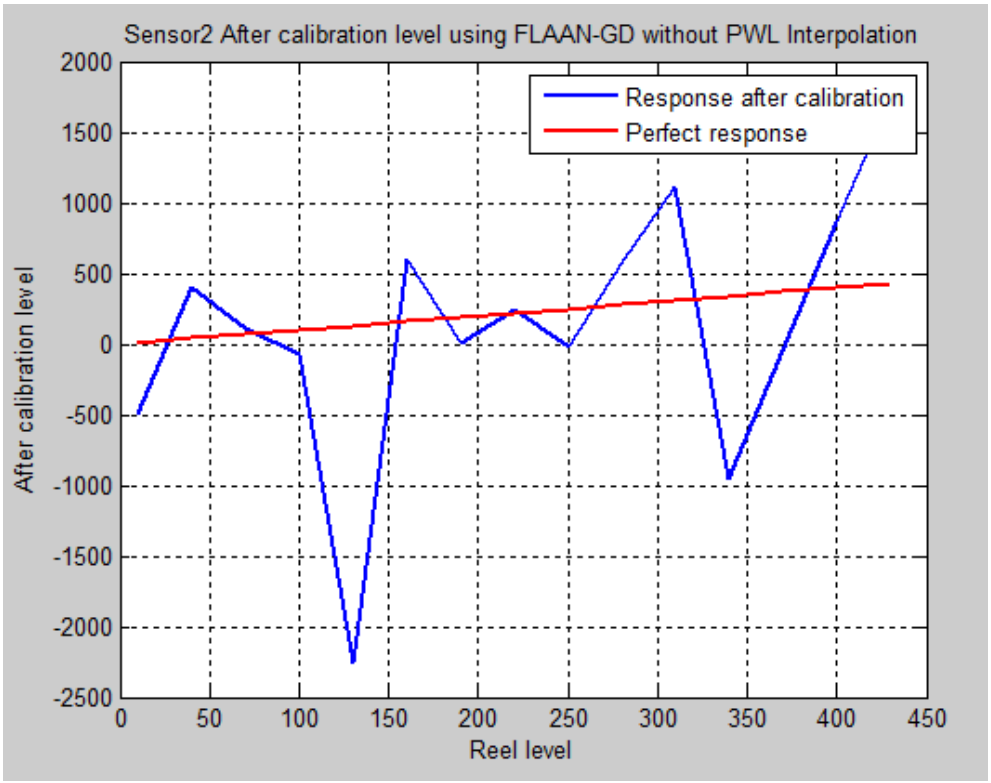

Figure III-21:Sensor 2 after level calibration using FLANN-GD without PWL interpolation

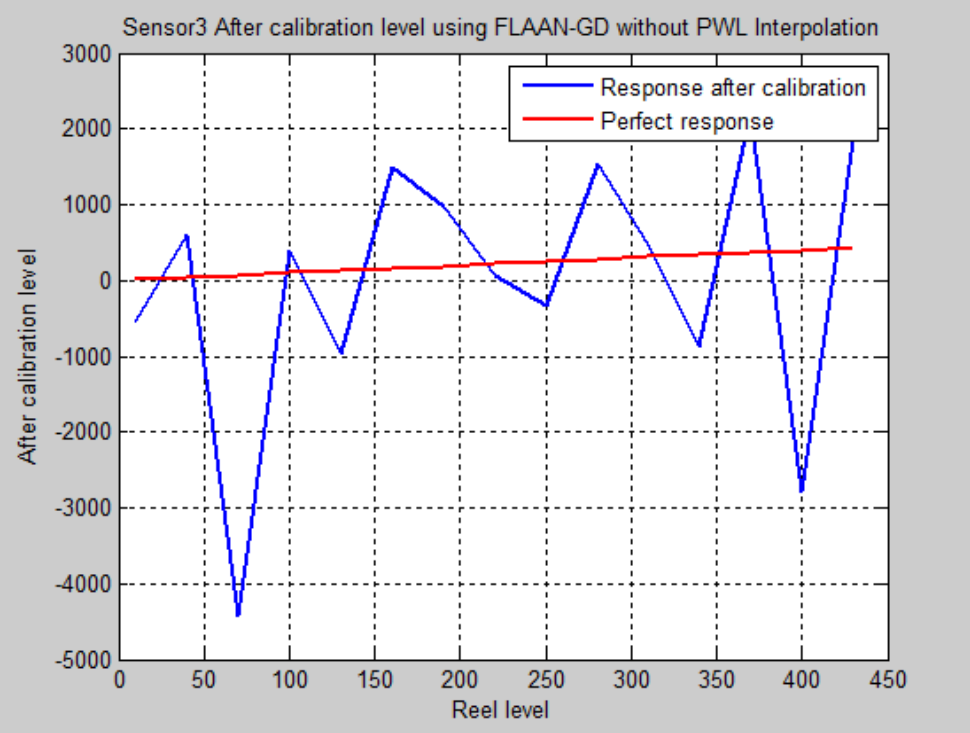

Figure III-22:Sensor 3 after level calibration using FLANN-GD without PWL interpolation

III.4.2. FLANN model with piecewise interpolation results

In the second implementation we used the FLANN-GD model with PWL interpolation and compare its response with the perfect response representing the error (mean error) between them. The results are shown in various figures listed in Table III-7.

Device	Model used	Specifications	Results in figures	Mean error (mm)
Level sensor 1	FLANN-GD With PWL Interpolation	P = 120	Fig. 23	32.6
Level sensor 2		P = 120	Fig. 24	47.1
Level sensor 3		P = 120	Fig. 25	26.6

Table III-8: Results using FLANN-GD with PWL interpolation

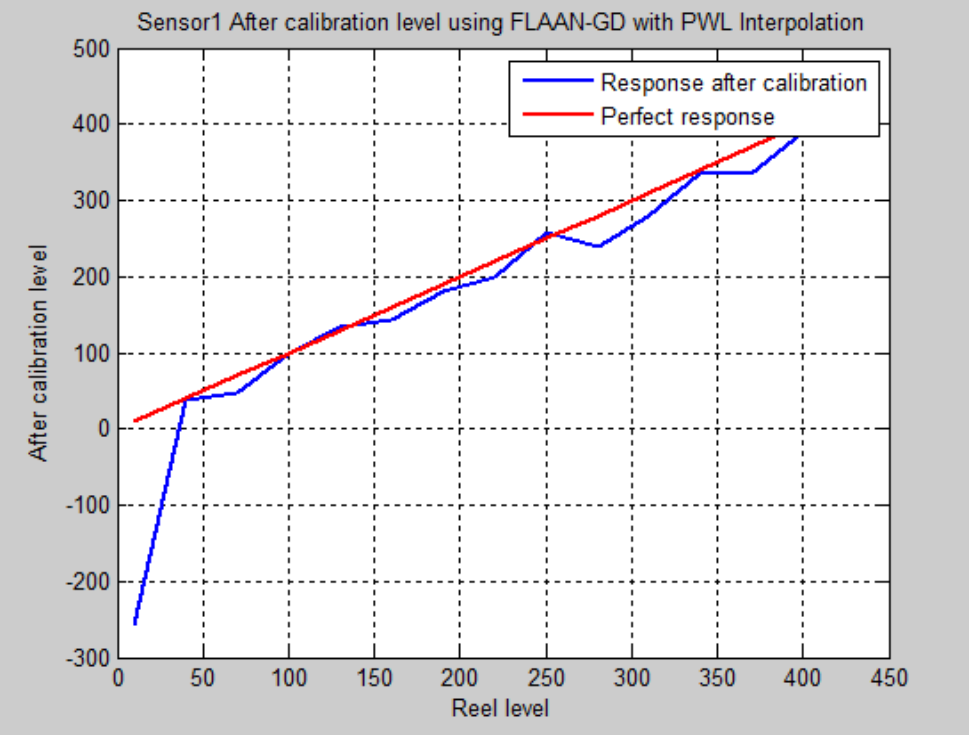

Figure III-23:Sensor 1 after level calibration using FLANN-GD with PWL interpolation

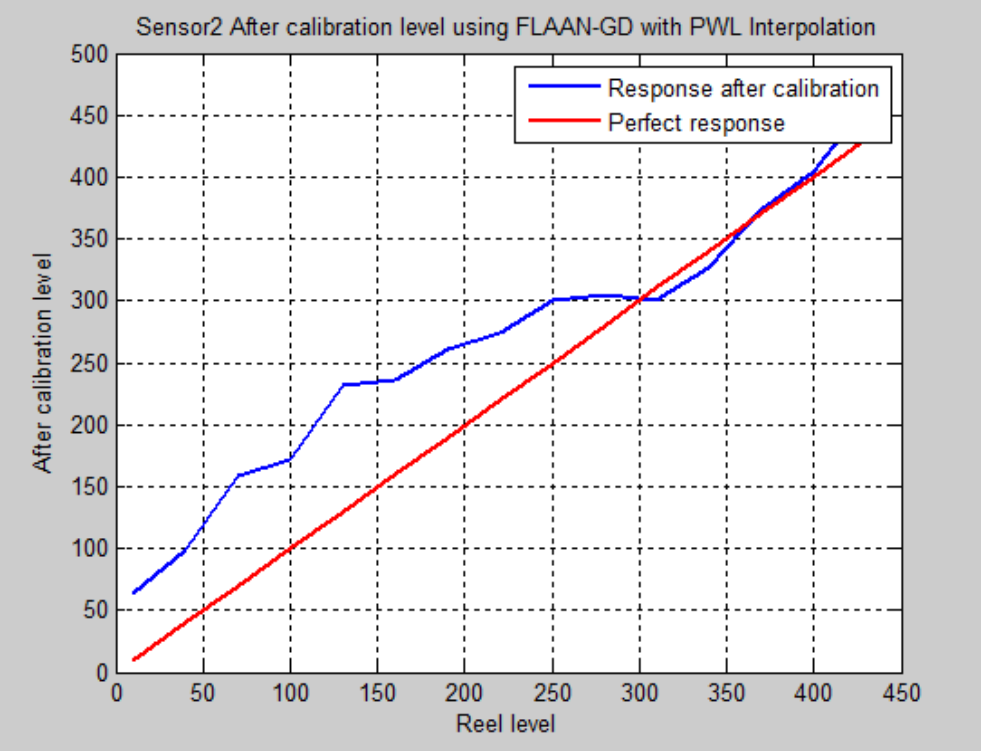

Figure III-24:Sensor 2 after level calibration using FLANN-GD with PWL interpolation

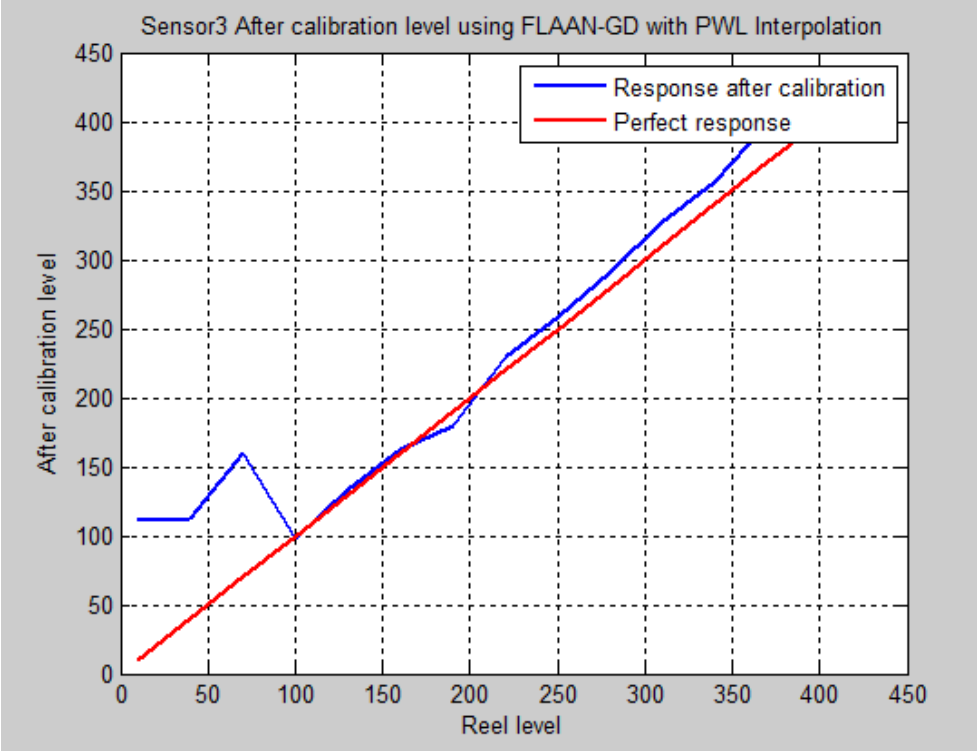

Figure III-25:Sensor 3 after level calibration using FLANN-GD with PWL interpolation

III.4.3. Performance evaluation and discussion

The results of the first and second implementations clearly demonstrate the difference between the FLANN model without PWL interpolation and the one with it. The FLANN model without PWL interpolation produced poor results (high error value) due to its limited generalization capability. On the other hand, the FLANN model with PWL interpolation delivered acceptable results with some errors. It is evident that this model offers better generalization, despite a slight decrease in accuracy and performance.

III.5. Conclusion

In conclusion, this chapter highlights the use of Functional Link Artificial Neural Networks (FLANN) in the intelligent calibration of level sensors, combined by Piecewise Linear (PWL) interpolation. Through a comprehensive analysis of FLANN architecture and learning algorithms, the study reveals significant improvements in handling sensor non-linearities. Case studies and experimental results demonstrate the practical benefits of FLANN with PWL-based calibration systems, offering enhanced accuracy and reliability compared to traditional methods. The successful implementation of these techniques not only optimizes sensor performance but also contributes to advancements in industrial automation and control systems, paving the way for further innovations in neural network-based calibration across various technological domains.

Conclusion

Conclusion

This study has demonstrated the significant improvements that can be achieved in sensor calibration within control systems through the integration of advanced computational models and interpolation techniques. By addressing the limitations of traditional calibration methods, we introduced an innovative approach combining the FLANN (Functional Link Artificial Neural Network) model with Piecewise Linear (PWL) interpolation. Our research showed that while the standard FLANN model offers efficient compensation of non-linearities, its generalization capabilities are limited, resulting in suboptimal performance. The integration of PWL interpolation effectively mitigates these limitations, enhancing the model's ability to approximate complex functions with greater generalization capability. The comparative analysis between FLANN models with and without PWL interpolation highlighted the substantial improvements in both accuracy and generalization. The enhanced model consistently delivered more reliable calibration results, demonstrating its potential for widespread application in modern control systems. In conclusion, the findings of this study underscore the importance of employing advanced algorithms and interpolation methods to achieve precise sensor calibration. This project contributes valuable insights to the field of intelligent systems, offering a robust solution for improving calibration processes. Future research can build on these results by exploring additional optimization techniques and expanding the application scope of the proposed model. The successful implementation of this enhanced calibration model holds promise for various industrial and technological applications, paving the way for more efficient and accurate control systems. As technology continues to evolve, such innovations will be crucial in maintaining the high standards of precision and reliability required in modern engineering practices.

Bibliography

Bibliography

[1] Shapin, S. (1996). The Scientific Revolution. University of Chicago Press.

[2] Bell, S. (2001). *A Beginner's Guide to Uncertainty of Measurement*. National Physical Laboratory.

[3] Thompson, A., & Taylor, B. N. (2008). Guide for the Use of the International System of Units (SI). National Institute of Standards and Technology.

[4] BIPM (2006). The International System of Units (SI). Bureau International des Poids et Mesures.

[5] International Vocabulary of Metrology – Basic and General Concepts and Associated Terms (VIM), JCGM 200:2012.

[6] Taylor, B. N., & Thompson, A. (2008). Guide for the Use of the International System of Units (SI). National Institute of Standards and Technology.

[7] JCGM 200:2012. International Vocabulary of Metrology – Basic and General Concepts and Associated Terms (VIM).

[8] ISO/IEC Guide 98-3:2008. Uncertainty of Measurement – Part 3: Guide to the Expression of Uncertainty in Measurement (GUM:1995).

[9] Bureau International des Poids et Mesures (BIPM). The International System of Units (SI), 8th edition, 2006.

[10] Doebelin, E. O. (2004). Measurement Systems: Application and Design. McGraw-Hill.

[11] Bentley, J. P. (2005). Principles of Measurement Systems. Pearson Education.

[12] Fraden, J. (2010). Handbook of Modern Sensors: Physics, Designs, and Applications. Springer.

[13] Northrop, R. B. (2005). Introduction to Instrumentation and Measurements. CRC Press.

[14] Morris, A. S., & Langari, R. (2012). Measurement and Instrumentation: Theory and Application. Academic Press.

[15] Neubert, H. K. P. (1999). Instrument Transducers: An Introduction to Their Performance and Design. Oxford University Press.

[16] Beckwith, T. G., Marangoni, R. D., & Lienhard, J. H. (2007). Mechanical Measurements. Pearson.

[17] Asch, G. (1982). Les Capteurs en Instrumentation (788 p.). Paris: Dunod.

[18] Attari, M., Boudjema, F. and Heniche, M.M. Linearizing a Thermistor Characteristic in the Range of Zero to 100°C With Two Layers Artificial Neural.

[19] Attari, M. Methods for Linearization of Nonlinear Sensors. Proc. CMMNI-4, Fourth Maghrebin Conference on Numerical Methods of Engineering, Algiers (Algeria), Nov. 1993, Vol. 1, p. 344-350.

[20] Gert van der Horn and Johan L Huijsmg Integrated Smart Sensors Designand Calibration Kluwer Academic Publishers, 1998

[21] G. DREYFUS, M. MARTINEZ, M. SAMUELIDES, M. B. GORDON, F. BADRAN,S. THIRIA, L. HERAULT.Réseaux de neurones méthodologie et application,Ouvrage de l'édition Eyrolles, 2002

[22] Paul BOURRET, James REGGIA, Manuel SAMUELIDES Réseaux neuronaux une approche connexionniste de l'intelligence artificielle,Ouvrage de l'édition TEKNEA, 1991

[23] Dehuri, S., & Cho, S.-B. (2010a). Multi-Objective Genetic Algorithm for Functional Link Artificial Neural Network. Springer.

[24] Dehuri, S., & Cho, S.-B. (2010b). A Comprehensive Survey on Functional Link Neural Networks and an Effective Design for Classification. Journal of Intelligent Systems.

[25] Mishra, D., & Dehuri, S. (2007). Functional Link Artificial Neural Network for Classification Task in Data Mining. Journal of Computer Science.

[26] Pao, Y.-H. (1989). Adaptive Pattern Recognition and Neural Networks. Addison-Wesley.

[27] Pao, Y.-H., & Takefuji, Y. (1992). Functional-Link Net Computing: Theory, System Architecture, and Functionalities. Computer.

[28] Chen, S., Billings, S. A., & Grant, P. M. (1992). Non-linear system identification using neural networks. International Journal of Control.

[29] Zhang, G. P. (2000). Neural Networks for Classification: A Survey. IEEE Transactions on Systems, Man, and Cybernetics.

[30] Haykin, S. (1998). Neural Networks: A Comprehensive Foundation. Prentice Hall.

[31] Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford University Press.

[32]S. K. Nanda and D. P. Tripathy "Application of Functional Link Artificial Neural Network for Prediction of Machinery Noise in Opencast Mines" Advances in Fuzzy Systems, 2011 Article ID 831261, doi:10.1155/2011/831261.

[33] J. C. Patra and R. N. Pal, "A functional link artificial neural network for adaptive channel equalization," Signal Processing, vol. 43, pp. 181- 195, 1995.

[34] Kwiatkowski, Robert (July 2022). Gradient Descent Algorithm — a deep dive

[35] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459-471.

[36] Pao, Y.-H., Park, G. H., & Sobajic, D. J. (1994). Learning and generalization characteristics of the random vector functional-link net. Neurocomputing, 6(2), 163-180.

[37] Paul BOURRET, James REGGIA, Manuel SAMUELIDES. Réseaux neuronaux une approche connexionniste de l9intelligence artificielle, Ouvrage de l9édition TEKNEA, 1991.

[38] Seborg, D. E., Edgar, T. F., & Mellichamp, D. A. (2010). Process Dynamics and Control **(**3rd ed.). John Wiley & Sons.

[39] Informations techniques VEGABAR 14

[40] ARPAIA, P. – DAPONTE, P. – GRIMALDI, D. – MICHAELI, L. ANN-Based Error Reduction for Experimentally Modelled Sensors, IEEE Trans. on Instrumentation and Measurement, vol. 51, no. 1, pp. 23-30, (2002)

[41] Saroj Kumar Mishra, Ganapati Panda and Debi Prasad Das « A Novel Method of Extending the Linearity Range of Linear Variable Differential Transformer Using Artificial Neural Network » , IEEE Transactions, Mai.2010