DEVELOPMENT OF A PIPELINE MONITORING SYSTEM FOR DETECTION, LOCALISATION AND CHARACTERISATION OF DAMAGE EVENTS IN PIPES

BY

ABA, Emmanuel Ngbede
PhD/SEET/2015/803

DEPARTMENT OF MECHANICAL ENGINEERING
FEDERAL UNIVERSITY OF TECHNOLOGY
MINNA

JUNE, 2021

DEVELOPMENT OF A PIPELINE MONITORING SYSTEM FOR DETECTION, LOCALISATION AND CHARACTERISATION OF DAMAGE EVENTS IN PIPES

BY

ABA, Emmanuel Aba
PhD/SEET/2015/803

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF DOCTOR OF PHILOSOPHY (PhD) IN MECHANICAL ENGINEERING (DESIGN AND SOLID MECHANICS OPTION)

JUNE, 2021

DECLARATION

I hereby declare that this thesis titled: **"Development of a Pipeline Monitoring System for Detection, Localisation and Characterisation of Damage Events in Pipes"** is a collection of my original research work and it has not been presented for any other qualification anywhere. Information from other sources (published or unpublished) has been duly acknowledged.

ABA, Emmanuel Ngbede
PhD/SEET/2015/803
FEDERAL UNIVERSITY OF TECHNOLOGY
MINNA, NIGERIA

_____
SIGNATURE/DATE

CERTIFICATION

The thesis titled: **"Development of a Pipeline Monitoring System for Detection, Localisation and Characterisation of Damage Events in Pipes"** by: ABA Emmanuel Ngbede (PhD/SEET/2015/803) meets the regulations governing the award of the degree of Doctor of Philosophy (PhD) of Federal University of Technology, Minna and it is approved for its contribution to scientific knowledge and literary presentation.

ENGR. PROF. O. A. OLUGBOJI
MAJOR SUPERVISOR _____
Signature & Date

ENGR. PROF. A. NASIR
CO-SUPERVISOR _____
Signature & Date

ENGR. PROF. M. A. OLUTOYE
CO-SUPERVISOR _____
Signature & Date

ENGR. DR. S. A. LAWAL
HEAD OF DEPARTMENT _____
Signature & Date

ENGR. PROF. Z. D. OSUNDE
DEAN, SIPET _____
Signature & Date

ENGR. PROF. O. K. ABUBAKRE
DEAN, POSTGRADUATE SCHOOL _____
Signature & Date

DEDICATION

This research work is dedicated to Almighty God who has been with me all through the course of this PhD programme and whose favour and strength has helped me to carry out this study successfully, and to my mother, late Mrs. Lucy Aba whose desire was to see me reach the peak of academic excellence.

The research work is also dedicated to my late sister, Ms. Irene Aba.

ACKNOWLEDGEMENTS

I would first like to thank my major supervisor, Prof. O. A. Olugboji. I appreciate all his efforts and dedication to support and guide me every step of the way. His genuine belief in my ability to carry out this research work gave me the confidence, courage, determination and stamina to propel me through this grueling process. It has been a wonderful honor, great privilege and humbling experience to be his student. Words cannot express how grateful I am for all you do for me.

I would like to thank my co-supervisor Prof. A. Nasir for being a member of my supervisory committee. His endearing and very generous support throughout my time at FUT Minna has been an immense light unto my feet. His very insightful advice has not only guided me through my research, but it has also been a source of help and encouragement in my personal life. From my very first encounter with him during this doctoral program, his wisdom and experiences have helped me grow into a better scholar, teacher and father.

I am also thankful to Prof. M. Olutoye being a member of my supervisory committee. His fatherly love, critical eyes, constructive feedback, support, encouragement, and time throughout my program experience helped me to continually pursue a doctoral degree. His support also helped me see that being a good scholar while being a good father is possible; that my family is not completely separate from my research. His flexibility and patience with me inspired me greatly.

I deeply appreciate the time and dedication of Dr. O. Adedipe who volunteered to participate in my study. Your insightful corrections helped to make my work better.

I am also grateful to the Department of Mechanical Engineering, School of Infrastructure and Process Engineering Technology, Federal University of Technology, Minna, for

giving me the opportunity to embark on this research. I appreciate the Head of Department, Mechanical Engineering Department, Federal University of Technology Minna, Dr. S. A. Lawal. My thanks also goes to all the staff of Mechanical Engineering department who have made themselves available when I needed their moral and technical assistance despite their tight schedules.

I am very thankful to the Executive Vice Chairman of NASENI, Prof. M. M. S. Haruna for the opportunity afforded to me to pursue this program, and for the study leave granted to me.

To all my friends who have supported me throughout this process, I am thankful for your countless words of advice and praises, and for your financial support too. I specially appreciate Alexander Ameh, Hilary Daudu, Samuel Agbatar, Best Agbi, Nwakamma Ninduwezour-Ehiobu, Chile Ushahemba, Ijeoma Nnoromele, Abayomi Okesola, Adesoji Oyedeji, Joseph Olowo, Kareem Aduagba, Mercy Madaki, Peter Ngbe, JohnPaul Bassey, Philip Ianna, Bundepuun Anande, Kizito Ameh, Martins Aiyetan. I specially want to appreciate my senior colleagues Dr. O. Olasupo, Dr. A. Agava and Engr. Emmanuel Ajani for all your help and guidance. I love you all.

I am especially grateful for the love, patience, understanding, and support of my wonderful wife, Justina. In every facet of my life, she has been the foundation I have needed to stand on. Her remarkable and unconditional love propelled me daily to pursue my doctoral degree. She has been my constant support through moments of tiredness, sadness, joy and stress. Furthermore, she has been a loving and attentive mother to our kids, Eldad and Ene while daddy was away at school every other week, studying in the evenings, and writing late into the nights for the last five years. Our family is really

blessed to have you in our lives. This could not have been possible without you by my side. I love you.

I would also like to express my deepest gratitude to my family. I thank my dad, Barrister I. C. Aba for encouraging me to embark on this program and for the constant show of love, concern and prayers. Thank you to my loving sister Joy for all your love and care, and Ene for your unquantifiable love, care and financial support. Thank you my loving brothers Andrew and John. You guys have been a pillar of support to me. Thank you my loving brothers Oche and Dr. Benedict and to my loving little sister Agnes. I thank my in-laws, especially my mother-in-law Mrs. Loveline Aturuocha and my dear sister in-law Chichi for their constant show of love, prayers and well-wishes. Everyone's words and actions of encouragement kept me going semester after semester. You also made this all possible. I love you all. Your shared valuable narratives not only enriched my study, but also enriched my life. Lastly, my heart and spirit are eternally grateful to God, from whom my source of life and strength come from. His unconditional favor, mercy, and grace have placed every aforementioned individual in my life to help me become the person I am today. He is my rock and refuge. He heard every prayer throughout this testing time, and He granted me the strength and courage to continue every single day.

ABSTRACT

Mathematical techniques for location of a damage event on a pipe was developed and tested using an experimental test rig. A pulse propagation velocity of 355 m/s was calculated from obtained data when static air was used as the transport fluid and 1538 m/s when flowing water was used as the transport fluid. The reason for the discrepancies between these values and the sound velocity values in air and water was investigated. A 21.2^{o}C and 15.3^{o}C temperature rise above ambient in the pressure pulses were obtained when static air and flowing water used were recorded. A difference of 20mm only was observed between the actual and computed event location when static air was used as the transport medium. When flowing water was used as the transport fluid, a difference of 23 mm was observed. Algorithms for the characterisation of damages in pipes were also developed. These were simulated with the results showing a good agreement between the shapes and magnitudes of the measured original and reconstructed pulses. The simulation was verified with experiments on the test rig. The results showed an underestimation in the magnitudes of the reconstructed pulses in the range of 40 – 45 %. This problem was solved by using a factor *K* obtained by dividing the maximum amplitude value of the original pressure pulse by that of the reconstructed pulse. A *K* value of 1.9 was calculated for the particular experimental data set used. Reconstruction of the measured original pulse at a damage location was achieved from combining the measured pulses from two other close locations using the developed Fourier transform based model. A wireless communication device was developed for transmission and processing of measured pressure pulses wirelessly to an analytics platform (ThingSpeak) for real time monitoring. Fifteen experiments were conducted on the experimental test rig using this device at pressure readings of 0.8 bar and 1 bar respectively in the pulse generator. The amplitude of the pulse at sensor 2 denoting the event location using 0.8 bar was 901 mm while the amplitude of the other four pulses were 499, 477, 420 and 346 mm respectively. For the pressure rating of 1.0 bar, the pulse at sensor 2 had an amplitude of 908 mm while the other four sensors were 509, 487, 429 and 355 respectively. In both experimental results, the pulse amplitudes ranged in values according to the distances of the five sensors from the event location with the pulses from the sensors closest to the event location having the highest values and those farthest from the event location having the least values. This showed the effectiveness of the wireless communication device.

TABLE OF CONTENTS

Content		Page
Cover Page		i
Title Page		ii
Declaration		iii
Certification		iv
Dedication		v
Acknowledgements		vi
Abstract		ix
Table of Contents		x
List of Tables		xiv
List of Figures		xv
List of Plates		xx
Symbols and Abbreviations		xxi
CHAPTER ONE		
1.0	**INTRODUCTION**	1
1.1	Background to the Study	1
1.2	Statement of Research Problem	3
1.3	Significance of the study	5
1.4	Aim and objectives of the study	7
1.5	Scope of the study	8
CHAPTER TWO		
2.0	**LITERATURE REVIEW**	9

2.1 Pipeline Monitoring and Security 9

2.1.1 Oil Spill Prevention in Nigeria 10

2.2 Methods of Pipeline Monitoring 12

2.2.1 Time Delay between Pulse Arrivals at Sensors (Cross Correlation Method) 12

2.2.2 Acoustic Monitoring 13

2.2.3 Wavelet Based Analysis 13

2.2.4 Inverse Least Square Methods 14

2.2.5 Inverse Methods 15

2.3 Damage Detection and Location Techniques 24

2.3.1 Transient Modeling 24

2.3.2 Acoustic Methods 25

2.3.3 Transient Analysis 26

2.4 Signal Processing 27

2.4.1 Signal Processing Methods 28

2.5 Sampling Signals 38

2.6 Signal Reconstruction 39

2.7 Digital Signal Processing for Characterisation of Damages in Pipes 40

2.8 Petroleum pipeline monitoring using Internet of Things (IoT) 44

2.9 Summary of Literature Review and Research Gaps 46

CHAPTER THREE

3.0 MATERIALS AND METHODS 52

3.1 Materials 52

3.1.1 Wireless Communication Device 54

3.1.1.1 The Choice of Micro Controller 55

3.1.1.2 Wi-Fi Module 56

3.2 Methods 57

3.2.1 Damage Location Based on Pressure Pulse Analysis 57

3.2.2 Propagation of Damage-Induced Pressure Pulses in anAir-Filled Pipe 58

3.3 Experimental Validation 60

3.4. Numerical Simulation and Modeling of Pressure Pulse Propagation for Characterisation 62

3.4.1. Mathematical Model 62

3.4.1.1 Mathematical Model for Sampling 63

3.4.1.2 Mathematical Model for Signal Reconstruction 64

3.5 Simulation of Mathematical Model 66

3.6 Experimentation for Validation of Simulation 66

3.6.1 Experimentation to Mimic Damage Event Caused by Explosion 67

3.6.2 Experimentation to Mimic Damage Event Caused by Drilling 67

3.6.3 Experimentation to Mimic a Potential Damage Event Caused by Vehicular Movement 68

3.6.4 Experimentation to Mimic a Potential Damage Event Caused Corrosion 68

3.7 ThingSpeak IoT Analytics Platform 70

3.7.1 Optimisation of Detection, Location and Characterisation System 72

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION 73

4.1 Simulation of Derived Mathematical Models 73

4.2 Experimental Results 85

4.3 Velocity of Pressure Pulse Propagation in Static Air 95

4.4 Velocity of Pressure Pulse Propagation in Flowing Water 97

4.5 Event Location in Static Air 98

4.6 Event Location in Flowing Water 100

4.7 Characterisation of Damage 101

4.7.1 Event Reconstruction 101

4.7.2 Damage Pattern Recognition 112

4.7.2.1 Damage from Explosion 112

4.7.2.2 Damage from Drilling 115

4.7.2.3 Damage from Vehicular Motion 119

4.7.3.4 Damage from Corrosion 123

4.8 Wireless Processing and Transmission of Data 126

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS 137

5.1 Conclusion 137

5.2 Recommendations 138

5.3 Contributions to Knowledge 138

REFERENCES 140

APPENDICES 146

LIST OF TABLES

Table		Page
3.1	Major Equipment	52

LIST OF FIGURES

Figure		Page
1.1	Proposed device operation of pipeline monitoring system	3
2.1	Computer-aided detection and classification based on pressure pulse signal analysis	28
2.2	Signal Fourier Transformation	30
2.3	Signal Wavelet Transformation	31
2.4	Digital low-pass filter	33
2.5	Schematic of the functions of low-pass, high-pass, band-pass and band-stop filters	35
2.6	An ideal sampling and reconstruction process	39
2.7	Wave Spectra of speech signal from a human in a neutral, angry, bored and happy state respectively	41
2.8	Wave spectra of bio-electric ECG signals showing a normal heart rhythm and that of a post myocardial infarction	42
2.9	Wave spectra of bio-electric ECG signals showing a Myoc`ardial infarction	42
3.1	Circuit diagram of connection between Arduino and Wi-Fi module	55
3.2	Schematic representation of sensors on a pipeline	58
3.3	Schematic of experimental flow loop	61
3.4	Schematic of experimental test rig for corrosion	69
3.5	IoT system adopted for pipeline monitoring	71
4.1	Original simulated pulse for data set 1	73
4.2	Sampled simulated pulse for data set 2	74
4.3	Reconstructed and original simulated pulse of data set 1	75

4.4	Original simulated pulse of data set 2	76
4.5	Sampled simulated pulse of data set 2	77
4.6	Reconstructed and original simulated pulse of data set 2	78
4.7	Original simulated pulse of data set 3	79
4.8	Sampled simulated pulse of data set 3	80
4.9	Reconstructed and original simulated pulse of data set	81
4.10	Measured pressure pulses from experimental rig sensors at pressure of 1 bar with air as transport fluid	86
4.11	Measured pressure pulses from experimental rig sensors at pressure of 0.8 bar with air as transport fluid	87
4.12	Measured pressure pulses from experimental rig sensors at pressure of 0.6 bar with air as transport fluid	88
4.13	Measured pressure pulses from experimental rig sensors at pressure of 1 bar with water as transport fluid	91
4.14	Measured pressure pulses from experimental rig sensors at pressure of 0.8 bar with water as transport fluid	92
4.15	Measured pressure pulses from experimental rig sensors at pressure of 0.6 bar with water as transport fluid	93
4.16	MATLAB® presentation of computed event location estimates using an air-filled pipe	99
4.17	MATLAB® presentation of computed event location estimates using a water-filled pipe	101
4.18	Measured pressure pulse at all six sensors	102
4.19	MATLAB® plot of pressure pulse at sensor 1	103
4.20	MATLAB® plot of pressure pulse at sensor 2	104
4.21	MATLAB® plot of pressure pulse at sensor 3	104
4.22	MATLAB® plot of pressure pulse at sensor 4	105
4.23	MATLAB® plot of pressure pulse at sensor 5	105

4.24	Sampling of original pressure pulse s_2 with MATLAB®	106
4.25	Reconstruction of original pressure pulse s_2 from samples of s_2	107
4.26	Sampling of original pressure pulse s_1 with MATLAB®	108
4.27	Sampling of original pressure pulse s_3 with MATLAB®	109
4.28	Reconstruction of original pressure pulse s_2 from pressure pulses s_1 and s_3	110
4.29	Reconstruction of original pressure pulse s_2 from pressure pulses s_1 and s_3 using reconstruction factor	111
4.30	Typical pressure pulse measured at experimental rig to mimic damage due to explosion (0.8 bar pressure reading at single sensor)	113
4.31	Typical pressure pulse measured at experimental rig to mimic damage due to explosion (1 bar pressure reading at single sensor)	114
4.32	Typical pressure pulse measured at experimental rig to mimic damage due to drilling with air as transport fluid (1000 Watts drill input power at single sensor)	115
4.33	Typical pressure pulse measured at experimental rig to mimic damage due to drilling with air as transport fluid (750 Watts drill input power at single sensor)	116
4.34	Typical pressure pulse measured at experimental rig to mimic damage due to drilling with water as transport fluid (1000 Watts drill input power at single sensor)	117
4.35	Typical pressure pulse measured at experimental rig to mimic damage due to drilling with water as transport fluid (750 Watts drill input power at single sensor)	118
4.36	Typical pressure pulse measured at experimental rig to mimic damage due to vehicular motion with air as the transport fluid (measurement at sensor 2)	119
4.37	Typical pressure pulse measured at experimental rig to mimic damage due to vehicular motion with air as the transport fluid (measurement at sensor 3)	120

4.38	Typical pressure pulse measured at experimental rig to mimic damage due to vehicular motion with water as the transport fluid (measurement at sensor 2)	121
4.39	Typical pressure pulse measured at experimental rig to mimic damage due to vehicular motion with water as the transport fluid (measurement at sensor 3)	122
4.40	Typical pressure pulse measured at experimental rig to mimic damage due to acid (25 ml of acid)	124
4.41	Typical pressure pulse measured at experimental rig to mimic damage due to acid (10 ml of acid)	125
4.42	MATLAB® representation of measured pressure pulse at sensor 1 using the wireless communication device for a 0.8 bar pressure reading	127
4.43	MATLAB® representation of measured pressure pulse at sensor 2 using the wireless communication device for a 0.8 bar pressure reading	128
4.44	MATLAB® representation of measured pressure pulse at sensor 3 using the wireless communication device for a 0.8 bar pressure reading	128
4.45	MATLAB® representation of measured pressure pulse at sensor 4 using the wireless communication device for a 0.8 bar pressure reading	129
4.46	MATLAB® representation of measured pressure pulse at sensor 5 using the wireless communication device for a 0.8 bar pressure reading	129
4.47	MATLAB® representation of measured pressure pulse at sensor 1 using the wireless communication device for a 1 bar pressure reading	130
4.48	MATLAB® representation of measured pressure pulse at sensor 2 using the wireless communication device for a 1 bar pressure reading	130
4.49	MATLAB® representation of measured pressure pulse at sensor 3 using the wireless communication device for a 1 bar pressure reading	131
4.50	MATLAB® representation of measured pressure pulse at sensor 4 using the wireless communication device for a 1 bar pressure reading	131

4.51 MATLAB® representation of measured pressure pulse at sensor 5 using the wireless communication device for a 1 bar pressure reading 132

4.52 ThingSpeak analytics platform page showing measured pressure pulses from all sensors for a 1 bar pressure reading 134

LIST OF PLATES

Plate		Page
I	Hwayeh CH34og +MEGA 328P Arduino	56
II	ESP01 ESP8266 Wi-Fi module	57
III	Experimental test rig1 showing various components of the rig with air as the transport fluid	82
IV	Experimental test rig 2 with wireless communication device and air as the transport fluid	83
V	Experimental test rig 3 with water as the transport fluid	84
VI	ThingSpeak analytics platform page showing measured pressure pulses from sensor channels 1 and 2	135
VII	ThingSpeak analytics platform page showing measured pressure pulses from sensor channels 3, 4, and 5	136

SYMBOLS AND ABBREVIATIONS

SYMBOLS	DESCRIPTION
Alphabetic	**Description**
a	Scale dilation parameter of wavelet transform
b	Translation (shifting) parameter of the wavelet transform
c	Velocity of wave (pulse)
dt	Small increment
E	Modulus of elasticity of the pipe material
E_v	Bulk modulus of elasticity of medium
f_i	Input vector
f_s	Sampling frequency
f (t)	Time domain function, a real valued function
$\dot{f}(\omega)$	Spectrum of f (t), a complex wavelet function
F_i	Imaginary component of spectrum
h (t)	Ideal reconstruction low-pass filter
j	$\sqrt{-1}$
n	Discrete integer variable
pi	Discrete impulse response function
P	Static Pressure
Pa	Absolute pressure [value is 10^5]
Pi	Discrete power spectrum

Po	Reference pressure
si	Discrete real or complex signal
Si	Discrete component of si
t	Time
t_1,t_2,t_3,t_4	respective time of events
T	Temperature
T_s	Sampling period
W_f (b, a)	Continuous wavelet transformation of function f (t)
W_t	Wall thickness of pipe
x_1,x_2,x_3,x_4	Respective sensor distances
x_o (t)	original signal
x_r (t)	reconstructed signal
X_{ts} (n)	sampled signal or time-discrete sequence

Greek

∞	Infinity
α	Chirping parameter
δ	Identity matrix or Kronecker Delta function
φ	Instantaneous frequency
ρ	Density
ρ_0	Reference density [998]
ω	Digital angular frequency

$\dot{\omega}$	Real angular frequency

Operators

$\otimes$	Convolution operation
$\odot$	Convolution operation

Abbreviations

CFD	Computational fluid dynamics
CSD	Cross-spectral density
CWT	Continuous wavelet transform
DAC	Digital-to-analog converter
DC	Direct current
DFT	Discrete Fourier transform
DSP	Digital signal processing
ECG	Electrocardiography
EEG	Electroencephalography
FFT	Fast Fourier transformation
FIR	Finite impulse response
GPIO	General Purpose Input/Output
GSM	Global system for mobile technology
HHT	Hilbert Huang transformation
HSA	Health Safety and Environment
IACC	Impressed alternating cycle current

ID	Internal diameter
IFFT	Inverse Fast Fourier transform
IoT	Internet of Things
MATLAB®	Technical, high-level computing software by MathWorks Inc.
ML	Maximum likelihood
MRI	Magnetic Resonance Imaging
NI	National Instruments
NNPC	Nigerian National Petroleum Corporation
PCB	Printed Circuit Board
OD	Outer Diameter
PSD	Power Spectral Densities
PVC	Polyvinyl Chloride
RAPID	Real-time Active Pipeline Integrity Detection
RF	Radio Frequency
SMS	Short Message Service
STFT	Short-time Fourier transformation
SWLD	Smart water leakage detection
USB	Universal serial bus

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

Large pipelines carry vast volumes of crude oil, natural gas, and petroleum products around the world. These pipelines play an important role in modern societies, as they provide essential fuels for critical functions such as transportation, power generation and heating. A ruptured pipeline has the potential to cause significant environmental harm due to the toxic properties of the goods being transported through these pipelines. Ruptured pipelines also cause huge economic as well as humanitarian losses. The risk associated with pipeline in terms of safety of people, damage to the environment and loss of income has been a major concern to pipeline integrity managers. The effects of oil pipeline vandalism in terms of the climate, in relation to Nigerian security, and economic, environmental, and humanitarian losses and consequences have been clearly demonstrated (Onuoha, 2009). Pipeline failures are caused by structural issues 40% of the time, operator error 6% of the time, other factors 25% of the time, outside force harm 27% of the time, and control issues 2% of the time (Agbaeze, 2000). According to the Nigerian Gas and Oil Pipeline Standards (GOST), a pipeline's normal lifespan is 33 years, but research has revealed that 42% of pipeline failures are mechanically induced, 18% are caused by corrosion, 24% are caused by third-party intervention, 10% are caused by operational error, and 6% are caused by human error (NNPC, 2007).

The monitoring of pipelines via sensors has garnered considerable awareness. A good example is seen in the use of acoustic sensors for monitoring and inspection of natural gas

pipelines (Park *et al.*, 2007, Agrawal, 2008). This study intends to use sensors for detecting, locating and characterising a variety of damage events in pipelines using pressure pulses, Wi-Fi for information transmission, and signal processing techniques based on the principle of vibration in pipes.

In general, pipeline defects can occur in the manufacture, construction, and operation processes. In Nigeria, pipeline damage are prominently as a result of unknown causes, third party activity, mechanical failure, corrosion and operational error (NNPC, 2007). Continuous monitoring or periodic assessments of the transmission pipelines' integrity are needed to ensure the pipelines' continued safe operation. The ultimate goal of pipeline monitoring and inspection is to locate defective positions and obtain precise measurements and assessments of the defects so that human operators can take effective action to avoid further damage. The aim of this research is to create a real-time pipeline monitoring and inspection system that will provide monitoring that is remote and continuous, and uses wire/wireless sensors to provide early detection, and warning of defects like leaks.

There are several activities that must be completed by a pipeline control and inspection system ranging from detection of leaks to inspecting pipes for structural flaws. In order to achieve this, several types of sensors and actuators are used (Agbakwuru, 2011). Fiber-optic sensors, acoustic sensors, and magnetostrictive sensors are only a few of the types of sensors that have been researched and tested for pipeline inspections. Each sensor type has its own set of characteristics and operating conditions. In this study, monitoring of pipelines using wire/wireless sensors are surveyed and a pipeline monitoring and inspection platform based on an active sensor network described. Also, the fundamental components of the proposed

sensor networks, as well as signal processing techniques for detecting, localising, and characterising bursts, leaks, and other anomalies in a pipeline system, will be addressed. A system for transmission and processing of the obtained signal data using a Wi-Fi module is also discussed.

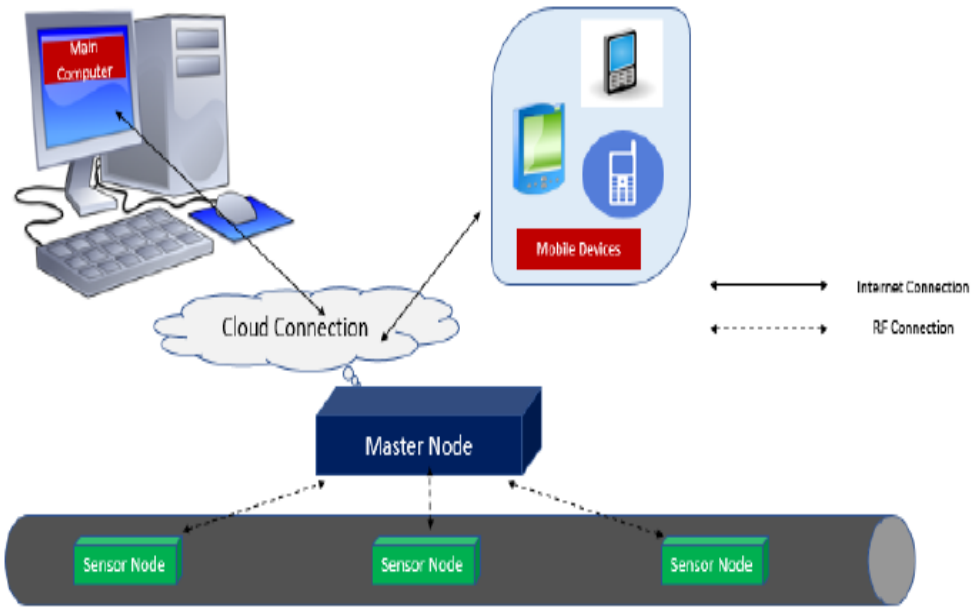

Figure 1.1: Proposed device operation of pipeline monitoring system (Ninduwezuor-Ehiobu, 2017)

In Figure 1.1, the Sensor node would sense pipeline intrinsic property such as pressure through the pressure pulses or waves, and transmit the wave to a master-node which distributes the information to a readable device in real-time.

1.2 Statement of Research Problem

The monetary and social expenses related with oil pipeline damage are quickly ascending to unsuitably significant levels. These expenses can be named follows:

- Petroleum product loss. Petroleum products loss until the damaged pipeline section is isolated after the burst has occurred can be considerably large. The financial implication of these petroleum product loss is always very high with negative consequences on the nations' economy (Agbakwuru, 2011).
- Punctuated flow. The damage event creates certain changes to the flow of the fluid. Losses through the burst can reduce the pressure in the pipeline to an unacceptable level. The flow may be hindered at various points along the pipe. If there should arise an occurrence of pipeline damage, an extensive number of buyers can be left without petroleum products (fuel, diesel, crude oil, gas) (Ninduwezuor-Ehiobu, 2017).
- Damage to infrastructure. Pipeline blasts can be amazingly hurtful for the encompassing structures. Enormous blasts can cause flooding of encompassing roads, buildings and water bodies leading to the death of aquatic life and health hazards in humans.
- Expenditure on repairs. Based upon the size and time-taken of the burst, fix expenses can turn out to be on the high side. The rebuilding of the affected environment and payment for lost property are the expenses that must be added on top of the sum spent for fixing the damaged pipeline.
- Environmental effects. Natural gas explosions can result in massive fires and massive amounts of destruction in the nearby towns. And, because methane is another greenhouse gas that contributes to climate change, exploding methane gas pipelines can do just as much physical and environmental damage. Crude oil spills can impact human health and the environment, resulting in fish and wildlife injuries or deaths, as well as poisoning of drinking water supplies.

Outcomes of pipeline damages recorded above are costly and hurtful. Accordingly, these outcomes are exceptionally unwanted. The general harm brought about by a pipeline damage relies upon the time between the real event and its seclusion. To limit the expenses related with pipeline damage, a procedure for speedy burst discovery, location and characterisation is essential.

1.3 Significance of Study

I. Prompt Action.

One major cause of persistent oil spillage across the globe is the inability of existing pipe leakage detecting devices to detect pipeline leakage in real time. In Nigeria for example pipeline leak prevention is done by onsite monitoring by individuals through the use of helicopters or speedboats; which is ineffective because in 2014 alone Shell Petroleum Development Company and ENI recorded 204 and 349 spills respectively (Amnesty International, 2015). This further justifies our mission to “live in the future (zero pipeline leaks across the globe) and build something interesting (leak detector with multiple functions) to tackle present day problem (pipeline leak)”.

II. Preservation of life, health and agriculture.

Over the years, vandals have regularly damaged the facilities of Nigeria's oil pipelines. These pipeline damages have great adverse effects on human, animal and plant life also. As these petroleum products flow from damaged pipes due to the late detection of area of damage, and poorly equipped response team; it leads to land and water pollution. Residents of such areas where the damage occurred loose access to

clean water for drinking, cooking, and other domestic activities. This in the long run leads to health hazards in the occupants of such areas, and sometimes leads to death. Aquatic life is also not spared as the waters are no longer conducive for life. Farming is also greatly impaired. Plants also die as they are choked by the pollution caused by these petroleum products.

III. Reduction in the financial strain on the Nigerian economy by loss of product.

Also water supply and sewer pipe works, electrical and telephone cable works, and other public works projects are often executed around pipelines. These third-party works by contractors has severally caused damage to petroleum pipelines. The financial strain on the Nigerian economy and the environmental impact of petroleum pipeline damage cannot be over emphasised (Shell in Nigeria, 2019).

IV. Efficient way of safeguarding pipelines.

There is need for development of a system that would monitor these pipelines, detect, locate damage along the pipelines, and determine the actual nature of damages on these pipelines-whether natural in nature (erosion, corrosion, ageing, welding defects, pressure surge problems) or mechanical harms from third parties (vandalisation, heavy duty equipment). This will definitely reduce loss of product and environmental pollution as the response team will be well equipped with the right tools and measures to tackle the damage even before arrival at the event location. This research work will also provide an efficient means of safeguarding these pipelines. Successful completion of the research will go a long way in increasing the country's revenue from petroleum products.

V. Available Market

According to epoxy pipeline engineers and other market reports, the pipeline leak detection industry is expected to grow to about $1.8 billion dollars in five years (Epoxy Oilserv, 2017), thus we venturing into this sector now means that we shall benefit from this exponential growth in the coming years. Also as a justification, there is a large transverse of pipeline network across Nigeria. The pipe length in km used in transport of oil and gas across Nigeria stands at about 12,714 km of pipe length (124 km condensate; 4045 km gas; 164 km light petroleum gas; 4441 km oil; 3940 km refined products) that require real-time monitoring of pipe integrity (CIA World Factbook, 2019). This also goes to show that there is available market waiting to harness this technology.

1.4 Aim and Objectives of the Study

The aim of this research is to develop a petroleum pipeline monitoring system for detection, localisation and characterisation of damage events in pipes. This aim can be achieved through the following objectives:

i. To develop mathematical techniques for locating, detecting and characterising damage events in pipes.
ii. To simulate the developed model using MATLAB® computer software
iii. To validate the simulation by experiments using a simulated pipeline with air and water as the transport fluid
iv. To optimise the designed location, detection, and characterisation system.

1.5 Scope of the Study

The scope of this research work was limited to the development of a system to locate, detect and characterise damage events in pipes. The approach of the technique for damage location and detection had to do with capturing of pressure pules through the length of the pipe combined with the review of these pressure pulses in the time domain. This work was carried out for both air and water filled pipes. The principle of digital signal processing was employed for the characterisation of damages in pipes through signal reconstruction process. The Fourier Transform was used to compute the spectrogram of the signals arising from various impulsive events by applying it to developed mathematical models. The Sinc Filter with the Fast Fourier Transform (FFT) was adopted for the filtering of the signal. The developed mathematical model was simulated using the MATLAB® computer software. This simulation was verified by developing experimental test rigs consisting of air and water filled PVC (Polyvinyl Chloride) for the flow of pressure pulses; capturing and recording of pulse data by an instrumentation system, a pulse generator, a Wi-Fi communication device for capturing, processing, recording and transmission of signal data to ThingSpeak, an Internet of Things (IoT) analytics platform service that allows you to aggregate, visualise, and analyse live data streams in the cloud from any location in the world.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Pipeline Monitoring and Security

Several types of technologies have been applied to pipeline networks to avoid spills and characterise damage/defects on pipes in real-time. Pipeline leakage has been a problem in existence for over a century now, from water pipeline leakage that leads to flooding, and crude-oil/natural gas pipeline leakage that leads to environmental degradation (Oil-Spillage) (Ninduwezuor-Ehiobu, 2017). Over the years, different inventions and innovations have been put in place to prevent and contain leakage on pipeline networks, but in the past few years, pipeline leakage in Nigeria is still present. In the Niger Delta, as a result of continued oil theft and acts of criminality, pipeline security and monitoring remains a high priority. The main sources of pollution in the Niger Delta currently are third-party interference and illegal refining. The Shell Petroleum Development Company of Nigeria Limited in 2018 alone lost more than 100 kilograms worth of spills and about 11,000 barrels of oil per day average theft from its operated Joint Venture (SPDC JV) pipelines. About 90% of these spills were caused by third party interference. These sabotage-related spills increased from 62 in 2017 to 111 in 2018 (Shell in Nigeria, 2018). One reason for this increase is due to the attractive price of crude oil products as they are seen as an avenue to wealth hence the increase in illegal refining.These statistics show that continuous air and ground surveillance as well as action by the governments security apparatus to prevent oil theft and third-party interference remains a necessary and important activity.

Crude oil theft, illegal activities and attacks continue to affect oil facilities being operated by both indigenous and international oil and gas companies in Nigeria leading to a very volatile security situation. This volatile security situation has led to several disruptions to production of petroleum products in the country and several incidents of environmental pollution. The federal government's revenue which is predominantly from sale of crude oil has also been affected. Industries, businesses and the public/private sector services have been hit also due to shortage of gas supply for production of electricity.

Operational conditions of the pipelines also have led to oil spills in Nigeria. Apart from spills cause by criminals, they were operational spills totalling 15; which were more than 100 kilograms in volume from facilities of Shell Companies in Nigeria alone. This was more than the 10 spills recorded by Shell companies in Nigeria in 2017. A total volume of 0.1 thousand tonnes of oil was spilled from operational activities in 2017 while in 2018, the total volume of oil spilled from operational faults was approximately 0.4 thousand tonnes (Shell in Nigeria, 2018).

2.1.1 Oil Spill Prevention in Nigeria

The Federal Government of Nigeria as well as foreign oil companies operating in Nigeria have in the past carried out several clean-ups and remediation of areas with oil spillage, but in areas where there are challenges in accessing the incident sites to carry out investigation of cause of spills and stoppage of leaks, the spills are often made worse.

Currently in Nigeria, several methods are being adopted the federal government, and foreign as well as indigenous oil and gas companies to prevent oil spillage. The Shell Petroleum

Development Company Joint Venture (SPDV JV) appraises the condition of pipelines regularly, maintains and replaces key sections of pipelines and flow lines. The company in 2018 installed 70 km of flow lines bringing its total distance of replaced pipelines and flow lines to about 1,300 km within a seven years period. The company has been able to achieve this through its pipeline and flow lines integrity management system that regularly manages pipeline integrity, inputs barriers at required places, and to prevent failures makes a recommendation as to where and when pipeline sections should be replaced. An enhancement was made by the company in 2018 to its integrity management system to manage integrity threats emanating from regular vandalisation/sabotage of pipelines.

Surveillance operations have also been carried out by the security forces of the Nigerian government to minimise and prevent oil spills caused by theft and sabotage of its facilities across Nigeria. The Shell Petroleum Development Company (SPDC) continues to do the same. The SPDC JV's operational areas, including its pipeline network in 2018 were safeguarded by SPDC's on-ground surveillance activities. This was to prevent occasions of third-party interference and to make sure that spills were detected and quickly responded to as best possible. In addition to this, the SPDC continues to carry out daily over-flights of the areas with pipeline network with the aim of identifying new spill activities or incidents. The SPDC also operates with specialised helicopters equipped with state-of-the-art high definition cameras which to a high extent, improves the surveillance of SPDC JV's assets. Anti-theft protection devices on very important structures, such as manifolds and wellheads have also been implemented by the SPDC.

Presently in Nigeria, when a leak on a pipeline is identified, production is stopped, and a visit is made to the site of the spill to determine the cause of the damage to the pipeline. An SPDC reports states that when a leak is identified on one of its pipeline, in line with the Nigerian government's regulation, a Joint Investigation Visit (JIV) team pays a visit to the site of the oil spill to determine the volume of oil spilled and the cause of the spill. The investigation team normally comprises persons from SPDC, federal government, state governments, regulators, security agencies and communities; often times local NGOs are invited as observers (Shell in Nigeria, 2018).

2.2 Methods of Pipeline Monitoring

2.2.1 Time Delay between Pulse Arrivals at Sensors (Cross Correlation Method)

In locating the position of a damage event on a pipeline, the use of sensors along the pipeline on opposite sides of the event is required. This method determines this from the arrival times of pressure pulses and the positions of sensors on the pipe. A major advantage of this method is that the pressure pulse time arrivals at sensors does not need to be measured to obtain the time delay as this is given directly. This method also stands out due to the fact that a good estimate of the time delay in pressure pulse arrival times using cross correlation is possible even when noise is added to these pressure pulses. This is due to the fact that the noise obtainable within these pressure pulses is averaged using the cross correlation method. This method when compared to the peak detection and threshold crossing methods gives a poorer fit. This is a disadvantage of using this method (Olugboji *et al.,* 2015).

2.2.2 Acoustic Monitoring

Rarefaction waves are associated with leak transients, and when these waves make contact with the pipe, a pressure (acoustic) waves is produced. These acoustic waves produces travel long distances inside the fluid, carrying information about what must have caused the impact on the pipe (Misiunas, 2004). Hydrophone sensors are mounted at distinct distances along pipelines to detect third-party interference (TPI) and leaks using acoustic monitoring technology. The accuracy of a leak position may be improved or boosted using acoustic leak correlators with portable computers or microprocessors. Various unwanted interruption noises, such as those from wind, water, traffic, and air craft are drawbacks of this system. Tiny holes with high pressure create clearer and louder sounds, making them easier to spot than a wide pipe burst with weak noise, and is surrounded by groundwater or escaped water. Also, plastic pipes tend to have very strong acoustic damping which in turn gives a wrong location of the damage.

2.2.3 Wavelet Based Analysis

A transient passing through a pipeline is partially mirrored and partially transmitted as it enters a leak. The leak position can be determined as long as the first reflected wave can be determined from the measurements (Wang *et al.,* 2001). Its benefit is that the method's definition is easy and straightforward to implement, given that the transient's initiating time is less than the reflection time between the leak and the measurement spot. It can also be used in real-time monitoring of pipeline ruptures. The disadvantage is that transient reflections can be caused by pipeline elements such as modified pipe diameter, elbows, or partially closed valves. In certain cases, depending on the location of the transient measurement, leak induced transient reflection cannot be detected (Wang *et al.,* 2001).

2.2.4 Inverse Least Square Methods

A regression between the measured transient pressure pulses and the modeled transient pressure traces is adopted in this method. The leak position and its size is obtained from minimising the deviation between calculated and measured pressure pulses when a discrete modelling at nodes in the network is carried out. Given enough measurement data, this method can be used for the calibration of any system parameter. Real tests have shown that inaccurate boundary conditions in a pipeline network and inaccurate transient modelling are challenges experienced with using this method.This is a major disadvantage of this method.

2.2.5 Inverse Methods

An Inverse problem is one where there is a relationship between the model parameters and the data (Olugboji and Yisa, 2012). This relationship is referred to as the model and the model usually takes the form of one or more formulas that the model parameters and data are anticipated to follow. Inverse method preserves monotonicity and correlation which helps in variance reduction methods, generating truncated distributions and order statistics.

The leak detection and location techniques used in transient analysis make use of different signal processing techniques for the computing and analysis of the transient waves or pressure pulses generated when a damage event occurs on a pipe.

Several research activities relating to damage detection, monitoring, location and sizing in gas and petroleum pipelines have been carried out in the past. Vibration-based approaches have been developed in the past and have been proven to be relatively successful in detecting damage in pipes (Razi *et al.*, 2013). For a long period of development, vibration-based and guided waves-based damage detection technologies have produced a good number of

research results. Some of the noteworthy research findings are reviewed here in order to improve this present work on the development of a petroleum pipeline monitoring system for detection, location and characterisation of damages.

Junxiao *et al.*, (2017) used the negative pressure wave (NPW), a stress wave caused by a pipeline leakage that spreads from the point of leakage to both ends of the pipeline, and the hoop strain variation along the pipe wall to detect damage in a gas pipeline. The study measured strain variation allowed accurate (within 2% error) and repeatable location (within 4% variance) of five manually controlled leakage points. Experimental results were then used to verify the effectiveness and the location accuracy for leakage in a 55 meter long model pipeline.

Guofeng *et al.*, (2017) used a stress wave propagation method with piezo-ceramic transducers to create a wavelet packet-based damage index matrix to classify crack damage in pipeline structures. On the specimen, four cracks were artificially cut, each with six damage cases corresponding to different crack depths. This aided in the simulation of cracks in various locations with varying degrees of impact. One piezo-ceramic transducer acted as the actuator for producing a pressure pulse that propagated through the pipeline specimen, while other piezo-ceramic transducers detected pulse responses in each crack event as they were deployed as sensors. In the study, based on the proposed damage index matrix, experimental results showed that the proposed approach can evaluate the crack severity and estimate the crack position in the pipeline structure. The proposed method's sensitivity reduced as the distance between the crack and the attached piezoceramic transducers grew.

Ravi *et al.*, (2016) monitored pipelines using vibro-acoustic sensing- a method of detecting leaks and foreign particles in pipelines for fluid transportation. The sensing method used in the work was based on pipe shell vibrations produced by interaction with pipe or flow, and the remote identifications of fluid transients. The system performance was a function of the thermodynamics properties of fluids. An analysis of pressure transient propagation in gas filled pipelines was carried out in the work. The study was able to detect pressure noise level produced by third party interference and leak events in the 50 to 300 Hz bandwidth at a distance of 10 km.

Golmohamadi (2015), used wavelet transform for processing signals to recognise damage and leak location in a hardware-based technique which used ultrasonic wave emission. In the study, leak site was identified using the wave emission speed and flight duration of backscattered signals when waves encountered leakage and were reflected.

Changhang *et al.,* (2017) used a low-power piezo-ceramic transducer as the actuator of vibrothermography and explored its ability to detect multiple surface cracks in a metal part. The Fourier Transform signal processing technique was employed in the work, and the obtained results showed that using the proposed low-power vibrothermography, all cracks can be identified quickly and easily.

Enrique *et al.;* (2016) developed a new and integrated damage detection procedure by combining directed waves and electro-mechanical impedance techniques based on smart sensing. They explored this combination of techniques and proposed an Electro-Mechanical Power Dissipation (EMPD) based indicator for damages. The proposed technique's

applicability was assessed in a number of experiments using structures that were either real-scale or lab-scale. In the study, the applicability of the proposed technique was tested through different experimental tests, with both lab-scale and real-scale structures and it was found that the technique can successfully contribute to positive damage identification even for instances of damage as complicated as those on concrete structures.

Kia *et al.*, (2018) proposed a new approach to damage detection of a concrete column structure subjected to blast loads using embedded piezo-ceramic smart aggregates (SAs). An active-sensing based approach was proposed where pressure pulses could be detected, as well as generated using the SAs that were embedded, and act as sensors and actuators.

Namuq (2013), designed and tested a laboratory experiment to determine the data transmission mechanism in boreholes using mud pulse telemetry. A flow loop, four pressure transducers at various points along the loop, a mud sire, a centrifugal pump, and a data collection system were all part of the test facility. It also included an actuator system that could simulate the noise patterns generated by common duplex or triplex mud pumps. To simulate dynamic pressure pulse propagation behaviour in the fluid within the flow loop, a theoretical model was developed using commercial Computational Fluid Dynamics (CFD) code. The theoretical approach was checked and calibrated using laboratory data that simulated a variety of process for data transmission in boreholes. The study was able to achieve a reasonably good agreement between expected and measured strain. In the laboratory, a novel method (continuous wavelet transformation) was used for detecting continuous pressure pulses obtained in a noisy environment, and it was applied to various simulated drilling activity conditions for data transmission in boreholes. In comparison to the

traditional approach, the results showed that the continuous wavelet transformation can be used to clearly distinguish and better detect continuous pressure pulse intervals, frequencies, and discontinuity positions in the time domain. Namuq (2013) focused on drilling as the source of pressure pulses, this study looked into the causes of various pressure pulse generating events on a pipeline in order to describe these damage-causing events. As a result, the Fourier transform was chosen because it is essential for calculating Cross-Correlation Functions (CCFs) between asymmetrical images in order to assist in averaging and comparing them.

Olugboji (2011), developed a model for determining the point of a damage event along a pipe. The cross correlation method was adopted for measuring the arrival times of pulses at any two sensors, and the principle of delay in arrival times between pressure pulses was the basis of the developed model. The obtained results in this work suggested that the developed model could be adopted in the monitoring of actual pipelines. The work was basically studied using pipes with static air. A major advantage of this method is that the pressure pulse time arrivals at sensors does not need to be measured to obtain the time delay as this is given directly using the cross correlation component of the MATLAB® software. This method also stands out due to the fact that a good estimate of the time delay in pressure pulse arrival times using cross correlation is possible even when noise is added to these pressure pulses. This is due to the fact that the noise obtainable within these pressure pulses is averaged using the cross correlation method. This method when compared to the peak detection and threshold crossing methods gives a poorer fit, with each application distortion form and size determining the error size. This is considered to be a disadvantage of adopting this method (Olugboji *et al.*, 2015).

According to Nandi and Wong (2014), vibration signals can be used to diagnose the health of rotating devices. The work gives an overview of how recent developments in pattern analysis techniques, together with the introduction of miniature vibration sensors and high-speed data acquisition technologies, have created a unique opportunity to develop and apply in-situ, beneficial, and non-intrusive condition monitoring and quality assessment methods for a wide range of rotating machineries. The study also included a study of traditional methods for vibration signal processing in the time and frequency domains. Following that, a series of recent computational intelligence-based approaches in this problem domain was presented, along with case studies that used single and multi-dimensional signals. The datasets they used in their case studies came from a number of real-world issues.

Hale and Olugboji, (2011) studied damage detection in gas pipelines by remote impact measurements. The work went a step further to show that using the Digital Signal Processing (DSP) technique of digital filtering and deconvolution, a fair reconstruction of the source of pressure pulse propagating through a gas pipe is possible. The work also stressed that a better reconstruction requires the use of inverse methods that take account of the non-linearity of the system.

Agbakwuru (2011), looked at how pipeline systems for notifying of encroachment, and leak prevention have advanced in the petroleum sector. The magnitude of oil spillage in the Niger-Delta area as a result of intended/unintended damages was highlighted, and a potential control method was proposed. The best solution for avoiding emissions due to pipeline failure, according to the work, is that of ensuring hydrocarbon exit from pipelines do not

occur. The study considered several methods, and it was suggested that monitoring using acoustic methods could be useful in detecting anomalies that are sound related. The output of an acoustic transmission on steel pipelines submerged in water was compared to that of a similar study on plastic water pipelines. It was discovered that pipelines immersed in water had low attenuation as compared to pipes buried in dirt.

Rashid *et al.,* (2014) proposed a transient pressure wave-based technique coupled with wavelet analysis to achieve reliable detection and localisation of abrupt bursts and leakages using Transform Analysis. A technique that uses the information carried in the transient pressure signal was presented. An algorithm was also proposed which is distributed in nature, and run on low power sensor nodes. The algorithm was deployed in-field on a custom pipeline test bed and performance results were documented for various testing scenarios. However, only pipeline burst detection was achieved with this proposed method but not leakage localisation, size estimation, and distance calculation. The proposed method was also best suited for high noise generating long pipeline networks.

Chung *et al.*, (2015) developed a built-in non-destructive inspection method called Real-time Active Pipeline Integrity Detection (RAPID) for detecting, locating, and monitoring the progression of pipeline damages. The system was based on an acoustic ultrasonic detection technology used in the Aerospace industry for detecting and monitoring damage to aircraft structures. RAPID made use of a sensor system which comprised of a group of piezoelectric transducers and receivers that establish an acoustic ultrasonic meshed network on the monitored pipeline structure. One of the system design requirements was that the sensor system needs to function continuously in the operational environment with peak temperature

up to 300 ºC for 15 years or more. The sensor network needed to be coated and robust in the harsh service environment. The meshed sensor network was designed to provide a detection sensitivity of 95 percent Probability of Detection. Standard installation procedures and tools were developed and the implementation was verified through field-testing. The RAPID prototype development was finished, significant progress was made in validating the system. To verify the capabilities of the system, a series of tests were performed by Acellent in partnership with Chevron utilising sections of 8in diameter steel pipes. During the tests a number of different sizes and depths of defects were introduced into the pipeline sections. These tests verified that the RAPID system was effective in detecting the occurrence of corrosion in the pipeline and monitoring its growth over time and the system was presented to both risk management companies and the gas pipeline industry, including Pacific Gas and Electric, Southern California Gas, and Chevron.

Olugboji and Yisa (2012), devised an inversion technique for reconstructing a pulse after it had propagated through a pipe; a dynamic pulse that has become increasingly distorted. The developed method made use of the theory of inverse problems. An inverse problem is one that occurs in many branches of science and mathematics where the values of some model parameter(s) must be obtained from the observed data. Inverse problems are those in which the causes are unclear, but the solutions are known. In a scenario that the problem's outcomes or effects are known but not what caused it. It is essential to have knowledge of a forward model capable of predicting data when the model parameters are already known. Measurements of these effects are taken, and calculations are made to decide what caused them in an inverse problem. This necessitates a definition of the data, which in most inverse problems is just a table of numerical values, for which a vector is a convenient representation.

A propagation model was developed and several tests were carried out to verify the model. From the test results obtained using the developed model for pulse propagation, it was found out that the inverse method of pulse reconstruction gives better results than the deconvolution filter method with a 15 – 20 % difference only between the original and reconstructed pulses was realised when using the inverse methods. Using the deconvolution method, a 40 % difference was obtained. However, it was noted that the inverse method consistently underestimated the pulse magnitude, whereas the deconvolution filter overestimated it, so there could be a case for using both methods in a practical application to obtain the best possible estimate. The work was validated experimentally and the experimental results were consistent with the model results which gave a similar level of underestimation.

Mclintyre (2017), created a technique for obstacle or breach detection and location in a conduit with a fluid medium. The device consisted of a number of individually recognisable Radio Frequency (RF) tags that were to be inserted into the conduit at an upstream stage. According to the work, each of the multitude of uniquely identifiable RF tags was included within a sensor pod with a customised size for effectively determining the size of the breach or obstacle, as well as a sensor for measuring the flow velocity of the fluid medium within the respective sensor pod. A second transceiver was mounted on the outside of the conduit and positioned near a downstream point in the fluid medium's flow direction.

Adnan *et al.,* (2015) explored the fundamentals of wave propagation in a gas pipeline, and some methods for revealing concealed information in a signal using synthetic signal simulation. The study's comparative approach for gas leak detection was the acoustic method. According to the research, wave propagation in the pipeline system can cause friction and

provide useful information for detecting leaks. In the work, the best combination of acoustic signal as wave propagation and Hilbert Huang Transforms (HHT) was proposed as the most effective method for detecting gas pipeline leaks.

Haseloh and LaFleur (2013), devised a pipeline leak detection system that included a pipeline, pressure sensors placed at regular intervals along the pipeline to track pressure inside the pipeline, and a monitoring station that received data from the pressure sensors. In the absence of a leak, the monitoring station developed a standard pressure profile of relative pressures collected by the pressure sensors along the pipeline, and issues an alert if one of the pressure sensors provides data that is uncharacteristic of the normal pressure profile, suggesting a leak near that particular sensor.

Jihoon *et al.*, (2017) suggested a new leak detection and position approach based on vibration sensors and generalised cross-correlation techniques. Estimation errors of power spectral densities (PSDs) and cross-spectral densities (CSDs) were considered in this analysis. An updated maximum-likelihood (ML) pre-filter with a regularisation factor was used in the proposed process. In this study, they found the optimal regularisation factor that minimises the theoretical variance in functional water pipe channels, as well as a theoretical variance of the time difference estimation error by summation in the discrete-frequency domain was established. Numerical simulations using a water pipe channel model were used to equate the proposed approach to other traditional correlation-based techniques, and field measurements revealed that the proposed updated ML pre-filter outperformed conventional pre-filters for generalised cross-correlation. When different types of pipes are linked, the study also presented a method to measure the leak position using the time difference estimate.

Motaz and Yousef-Awwad (2017) investigated the use of wireless sensor networks to detect leaks in underground water pipes in order to solve the issue of water dispersion in water delivery networks. Prevention of leakage, and detection in water delivery networks were found to be critical for efficient natural resources usage in the study. To solve this problem and make the leakage detection process easier, the researchers created a wireless network framework that uses mobile wireless sensors to break detection, and also for saving electricity, money, and time. The system could also measure the water level in the tank, and switch on the pump when the water level fell below a certain level because it had Smart Water Leakage Detection (SWLD) in the pipelines. The work was primarily divided into two sections: the first stage of their project was an alarm based on Global System for Mobile technology (GSM) to send Short Message Service (SMS) to the owner. Sensors, a GSM board, an Arduino, and relays to power the computer comprised the system's basic components. The control portion of the project included using an Android program to control the pump. The use of their proposed system resulted in increased operational performance, shorter wait times, and lower maintenance costs after detection of a leak.

2.3 Damage Detection and Location Techniques

These techniques are basically the techniques used in oil and gas industries, and those based on transient methods used for detecting existing leaks.

2.3.1 Transient Modeling

To continuously observe irregularities in pipelines, a continuous pressure pulse simulator can be used. A good number of techniques dependent on fluid flow rates and pressure abound. In

the petroleum industry, the majority of them are used. Two methods of transient modelling exist: the dynamic volume balance method and the pressure discrepancy method. An analysis between measured data and simulated data is used by the two methods. A computation of two flow discrepancies at the outlet and inlet is carried out using the dynamic volume balance method. An assumption that the pipeline is intact is made during the calculations, a fault sensitive method. A comparison between the pressure values at a number of points across the system is carried out using the pressure discrepancy method. A 15-30 s sampling interval is used while collecting the flow measurements and real-time pressure at both ends of the pipe. In this method, the flow and the measured pressure are used as the boundary conditions. The measurements made eventually reveals the leak when it occurs. As a result, a divergence occurs between the pressure values and the simulated flow. A response time of less than an hour on average is expected using this method. (Henrie *et al*., 2016).

2.3.2 Acoustic Methods

For detecting and locating leaks, acoustic measurements are a viable option to the model-based methods previously described. A feature of the generated leak is an acoustic signal that propagates along the walls of the pipe or within the flowing fluid. Fixed continuous monitoring systems and the use of geophones manually are part of the complexities associated with using acoustic systems (Rajtar and Muthiah, 1997). Detection and location of abnormalities in water distribution systems, petroleum and chemical systems, and nuclear power plants are areas where acoustic leak detection systems have successfully been used. Not-withstanding, according to Wang et al. (2001), acoustic methods have several drawbacks. The performance of acoustic methods for detection and location of leaks are affected by the following factors:

- The leak characteristics. A signal with low strength is produced by the escaped water surrounding a huge pipe leak
- Traffic, wind, etc. (undesired noise/interference)
- Large number of leaks which lead to wrong leak positions
- Plastic pipes with very strong acoustic damping
- Different conditions of sound propagation across the pipeline

2.3.3 Transient Analysis

Along a pipeline, when a leak occurs suddenly, a decrease in pressure occurs after which a pressure pulse travels downstream or upstream. This pressure pulse when analysed can be used to detect and locate the leak or damage. Extraction of damage information from the measured pressure pulse is the primary objective of all methods for transient leak detection. Pulse generators, solenoid side discharge valves, and other special devices along with pumps, inline valves and other system elements are used to generate transient events or create a damage scenario. This method requires pressure measurements sampled at high frequencies because the speed of the transient wave can be over 1000 m/s. The method that is employed for further analysis determines the generated transients' characteristics and the position measurement choice.

In carrying out leak detection, the simple and most straightforward application of transient analysis is the method of leak reflection (Brunone and Ferrante, 2001). At the leak point along a pipe, partial reflection of the travelling transient wave occurs. Employing calculations, the location of the leak can be found as long as upon measurement of the pressure pulse, the reflected wave is identified. The time domain reflectometry (TDR) is the

principle upon which this method is based. Regression of measured and modeled transient pressure pulses is used by the inverse method (Olugboji and Yisa, 2012). Modelling of the leak is done at nodes within the network. Leak location and size are produced as a result of minimising the deviation between the calculated and measured pressures. Having adequate measurement data, apart from leak detection, calibration of system parameters can be achieved using inverse methods.

In tracing a leak, the transient trace damping rate is used by the transient damping method (Wang *et al*.; 2002). Leaks and pipe friction are causes of the transient wave decay. Comparison between the simulated results and the measured pressure that contains the leak-induced damping will lead to leak detection. From literature searched in the course of this study, and to the best of the author's knowledge, investigation into how this method can be possibly applied in pipe networks has not been previously carried out. The analysis of the transient response while in the frequency domain is used by the frequency method (Ferrante and Brunone; 2003, Lee *et al*., 2003, Olugboji, 2011). Transformation of time-domain data to that of the frequency domain is carried out by Fourier transform. The leak position can then be gotten by making comparisons between the dominant frequencies of leaking pipelines and no leak pipelines.

2.4 Signal Processing

Signal processing is carried out for various reasons. Some of the objectives of signal processing as illustrated in Figure 2.1 which includes:

- Information gathering for signal analysis: determination of system state
- Detection: detection of abnormality by comparison to reference/normal values

- Monitoring: obtaining continuous or periodic information, identifying relative changes to the state of the system
- Remedy and control: intervention based on measurements
- Evaluation (Abraham, 2017).

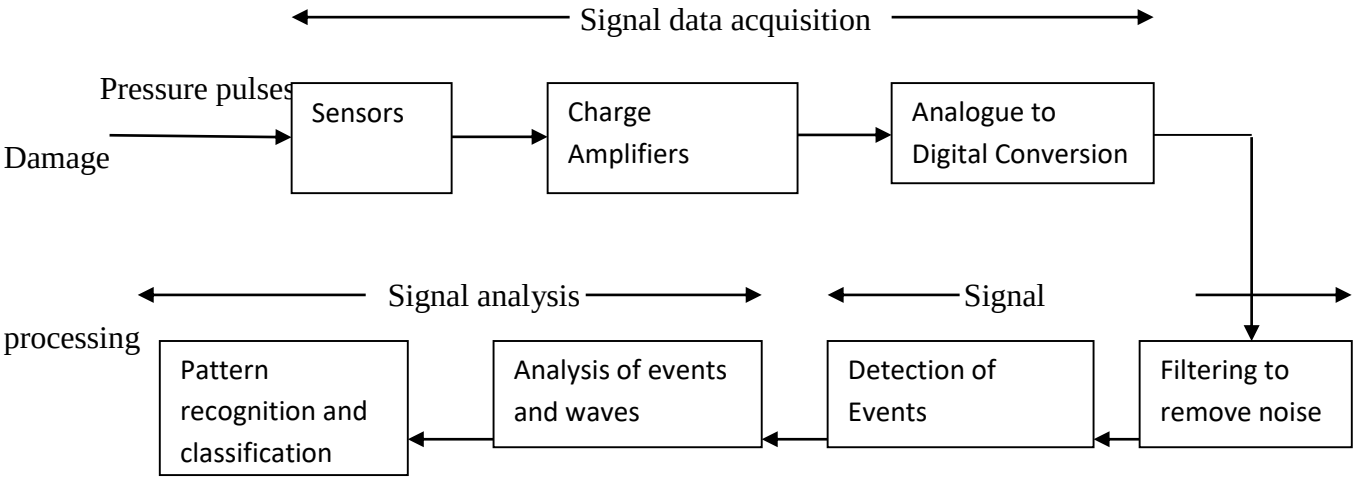

Figure 2.1: Computer-aided detection and classification based on pressure pulse signal analysis (Abraham, 2017).

2.4.1 Signal Processing Methods

For analysing and processing of digital signals in their discrete forms, several methods and integral transforms exist. In order to get useful information form these signals, a lot of the existing methods and transforms are can be adopted. Some of these important methods are identified below:

1. Fourier Transform

This method forms the backbone of the various algorithms and methods used for signal processing. It is very much used in several areas of engineering and science (Blackedge,

2006). Revealing hidden information is the main advantage of signal transformation. According to Ortiz *et al*., (2009) and Soliman *et al*., (2001), the Fourier transform is the most popular signal analysis transformation technique. The following equations represent the mathematical expressions that define the Fourier transform (Goswami and Chan, 1999):

$$\dot{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t}\, dt \qquad 2.1$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \dot{f}(\omega) e^{-j\omega t}\, d\omega \qquad 2.2$$

Where $\dot{f}(\omega)$ = function of frequency/Fourier transform of function $f(t)$

$f(t)$ = function of time

t = all real numbers

ω = angular frequency

j = period function

By disintegrating a signal into sine and cosine waves of varying frequencies, the Fourier transform, as shown in Equation 2.1, transforms the function f (t) from a time domain function to a frequency domain function $\dot{f}(\omega)$. The inverse Fourier transform transforms a function from a frequency domain to a time domain using Equation 2.2. For this research work, the analysis of the signals was done in the time domain. This is shown in Figure 2.2.

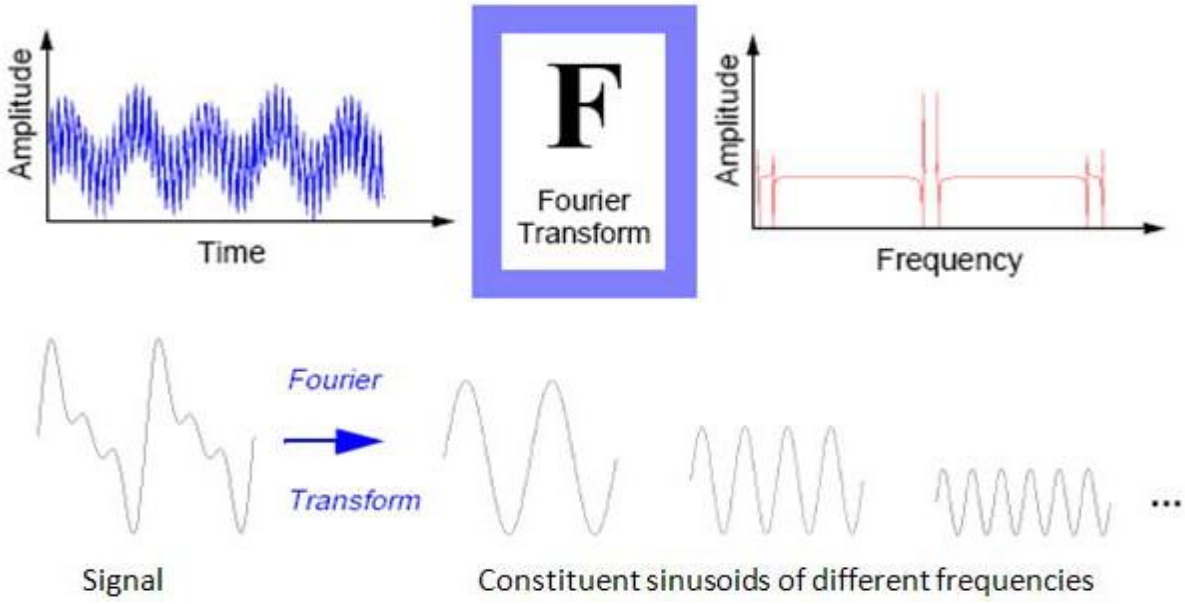

Figure 2.2: Signal Fourier Transformation (Namuq, 2013)

In Figure 2.2, the upper part of the image shows the pressure values which are a function of time being represented as a function of frequency using the Fourier transformation, and it provides information about the frequency content in the signal. In the lower part of the image, Fourier transformation is used to by break up a signal in sine and cosine waves of different frequencies

2. Wavelet Transform

Assuming that the signal being processed is non-stationary, this operation which is correlation-type based, and in terms of the kernel correlation and amplitude, includes a scaling property can be used (Blackedge, 2006). This transformation is a high level method compared to the Fourier transform. It is used in various sciences (Guan *et al*., 2004). The wavelet transformation of a function f (t) with respect to some analysing wavelet ψ is represented by the mathematical expression (Goswami and Chan, 1999):

$$Wf\ (b,a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \dot{\varphi}\ \frac{t-b}{a} dt, \qquad a > 0 \qquad 2.3$$

Where $Wf\ (b,a)$ = continuous wavelet transform

$f\ (t)$ = function of time

t = all real numbers

$\dot{\varphi}$ = impulse response

a, b = wavelet parameters

The wavelet transform distinguishes the frequency components of a signal using a window technique with varying sizes while preserving the signal's time dependence (Soliman *et al*., 2001, Ortiz *et al*., 2009). The wavelet transformation and decomposition of a wavelet signal is shown in Figure 2.3.

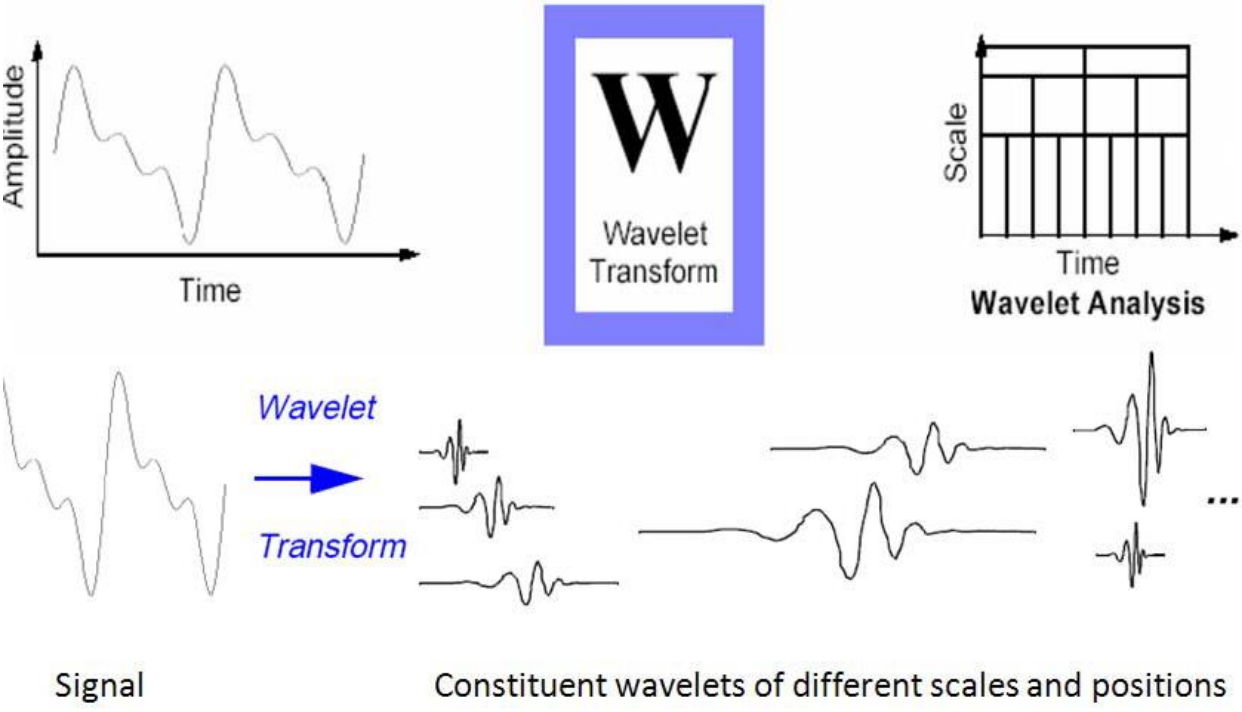

Figure 2.3: Signal Wavelet Transformation (Namuq, 2013)

Figure 2.3 shows the graphical representation of the wavelet coefficients for a range of scales as a function of local domain known as the scalegram. The scalegram is seen in the upper right side of the image. At higher scale lower frequencies will be analysed while at lower scale higher frequencies will be analysed.

3. Digital Filtering

Digital filters can be used to model a collection of data that already exists, but they are typically used to process a series of data samples that have been collected previously and stored, or processed on sampling in real-time. Time domain filtering tends to enhance certain frequencies in the "signal" while attenuating other frequencies in the "noise". In Digital Signal Processing (DSP), digital filters are adopted. These digital filters are actually algorithms, and they are two main classes. These are:

I. Fourier-based filters

II. Convolution-based filters

Non-recursive filters are convolution filters. They are real space filters, which are linear processes that work directly on data. Fourier filters work with data obtained by computing a signal's Discrete Fourier Transform. The Fast Fourier Transform algorithm is used to achieve this (Giurgiutiu, 2014).

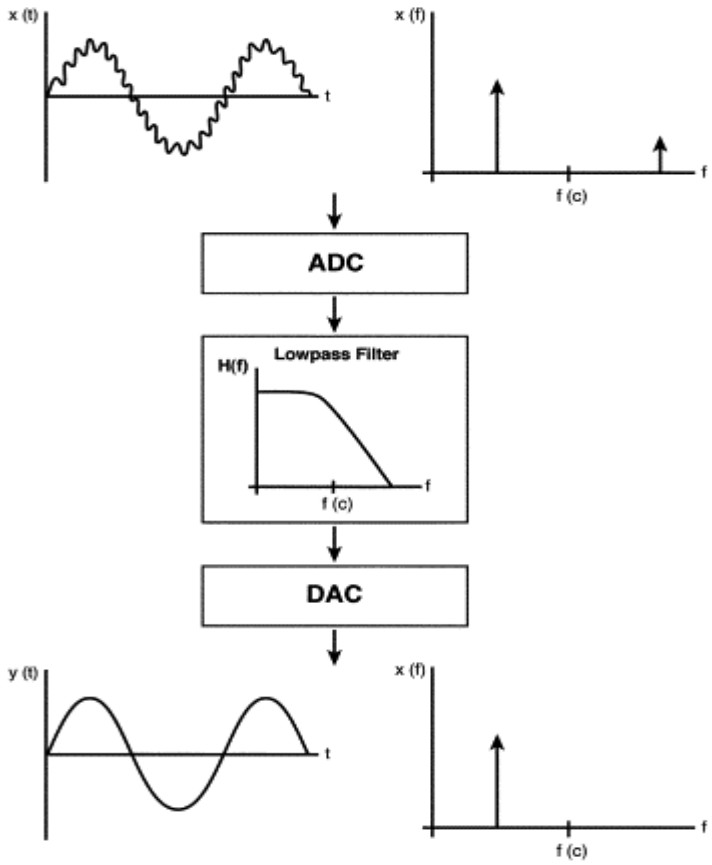

Figure 2.4: Digital low-pass filter (Burgess, 2014)

In figure 2.4, the top and bottom are a digital filter's input and output, respectively. On the left is the conventional time domain tracing, and the frequency domain representation is shown on the right. The input signal, *x*, is a low-frequency sine wave contaminated by a higher frequency sine wave of a lower amplitude. In the frequency domain, these two sine waves are represented by the single bars, each at their respective frequency and with a height proportional to their amplitude. After filtering, the high-frequency sine wave has been attenuated, resulting in an output, *y*, consisting of the pure low-frequency sine wave.

- Real-Space Filters

The principle of moving windows is the basis for these kind of filters. A single value is outputted when an element data sample is processed. This process is then repeated as the next signal element is processed when the window moves to it. The FIR (Finite Impulse Response) filter is an example of a real space filter. This type of filter is non-recursive (Blackledge, 2006):

$$s_i = \sum_j p_i - jf_i \qquad 2.4$$

Where,

f_i = input of the filter

s_i = output of the filter

p_i = filter kernel

- Fourier-Space Filters

These are filters that operate on a signal's DFT. They are operations that are multiplicative in nature. Denoting the DFT's of Si, Pi and Fi with s_i, p_i and f_i, the Fourier space discrete convolution is given by (Blackledge, 2006):

If Si, Pi and Fi are taken to denote the DFT's of s_i, p_i and f_i respectively, then, using the discrete convolution theorem, in Fourier space (Blackledge, 2006)

$s_i = \sum_j p_i - jf_i$ transforms to:

$$S_i = P_i F_i \qquad 2.5$$

A filtering operation is one in which the components of the Fourier experiences changes in its distribution as a result of a process that is multiplicative in nature. In Equation 2.5, Pi may be referred to as a filter and Si can be considered to be a filtered version of Fi. Fourier filters are broadly divided into five filter classes:

i. Low-pass
ii. High-pass
iii. Band-pass
iv. Band-stop
v. Anti-aliasing

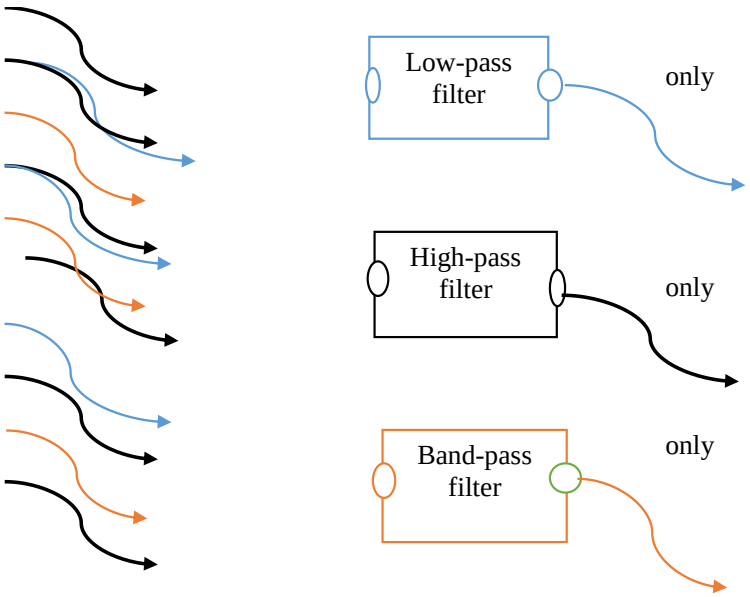

Figure 2.5: Schematic of the functions of low-pass, high-pass, band-pass and band-stop filters

As observed in in Figure 2.5, a low-pass filter within a certain range attenuates or suppresses the high frequency components, while allowing the low frequencies of the same spectrum. A high-pass filter attenuates the low frequency components, while allowing the high frequencies of the same spectrum within a certain range. Only frequencies with a certain band are lowed to pass when using a band-pass filter. Band-pass filters are either high-pass or low-pass filters (Wolf, 2019).

4. Fast Fourier Transform

Desiring less multiplications and additions while computing the Digital Fourier Transform (DFT), the Fast Fourier Transform (FFT) is used. Equation 2.6 and 2.7 gives the DFT of an *N*-point vector in standard form (Blackledge, 2006):

$$\text{Fm} = \sum_n fn\, e^{\frac{(-2\pi inm)}{N}} \qquad 2.6$$

$$\text{Where } \sum_n \; = \sum_{n=0}^{N-1} \qquad 2.7$$

For a DFT with N number of points, the amount of computation needed is given by Equation 2.8 and 2.9:

$$\text{W}_\text{N} = e^{\frac{(-2\pi i)}{N}} \qquad 2.8$$

$$\text{Then } \text{F}_\text{m} = \sum_n W_N^{nm} fn \qquad 2.9$$

Equation 2.10 shows the resulting matrix equation (Blackledge, 2006):

$$\begin{matrix} F_0 \\ F_1 \\ \vdots \\ F_{N-1} \end{matrix} = \begin{matrix} W_N^0 & W_N^{01} & \dots & W_N^{0(N-1)} \\ W_N^{10} & W_N^{11} & \dots & W_N^{1(N-1)} \\ \vdots & \vdots & & \vdots \\ W_N^{(N-1)0} & W_N^{(N-1)1} & \dots & W_N^{(N-1)\,(N-1)} \end{matrix} = \begin{matrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{matrix} \qquad 2.10$$

A multiplication in the form of *N*N* is required for computing the DFT. A Matrix whose coefficients are given by a constant W_N, raised to the power of *nm* is multiplied by an *N*-point vector *fn* to achieve this.

5. Convolution and Correlation

Signal analysis is hugely dependent on the process of convolution. Even though convolution and correlation are have some properties that are significantly different, the two processes are quite similar. Fourier transform is fundamentally associated with convolution and correlation.

- Convolution

Equation 2.11 shows the operation in one dimension that gives the functions *f* and *g* convolution (Blackedge, 2006):

$$\mathrm{f} \otimes \mathrm{g} = \int_{-\infty}^{\infty} f(T) g(T-t)\, dT \qquad 2.11$$

Convolution is commutative, associative and distributive respectively.

- Correlation

Equation 2.12 gives the functions *f* and *g* cross-correlation in a single dimension (Blackedge, 2006):

$$\mathrm{f} \odot \mathrm{g} = \int_{-\infty}^{\infty} f(T)g(T-t)\,dT \qquad 2.12$$

Correlation generally does not commute. Correlation is quite similar to convolution with a very small but significant difference.

2.5 Sampling Signals

For computers to be able to process digital signals, analogue signals are first converted into a number sequence through a process known as digitisation (Blackedge, 2006). The resultant digital form still contains all the information that was obtainable in the original analogue version after the conversion is done. This is only possible if the analogue signal is sampled at the right rate. This is the whole idea behind the sampling theorem. Equation 2.13 defines the sampling theorem which states that if a band-limited continuous function *f (t)* has a complex spectrum $F\ (\omega), /\omega/ \leq \dot{\omega}$, then it is define by values regular interval spacing (Blackedge, 2006):

$$\delta t \leq \frac{\pi}{\varphi} \qquad 2.13$$

The parameter $\dot{\omega}/\pi$ is known as the 'Nyquist frequency'. When sampling an analogue signal the sampling frequency must be greater than twice double of the analogue signal's highest frequency for reconstruction of the original signal from the sampled version to occur. The signal's Nyquist frequency must be at least equal to the sampling rate if loss of information is unwanted when converting an analogue signal to a digital signal. If the above criterion is not adhered to, a distortive effect known as "aliasing" will occur causing the replicated spectrum to overlap.

Satisfying the condition, $\delta t = \frac{\pi}{\varphi}$ when sampling a digital signal gives rise to a Nyquist sampled signal where the Nyquist frequency is given by ϕ̇/π, which is equivalent to two times the value of the signal frequency's bandwidth. This sampling interval must be used to retrieve all of the information present in the original analogue signal from the digital signal while avoiding aliasing.

2.6 Signal Reconstruction

In digital signal processing, the term "reconstruct" has a special meaning. It is related to converting a signal from its discrete form to a continuous form using a Digital-to-Analog Converter (DAC) and an ideal reconstruction low-pass filter (Zisselman *et al.*, 2018).

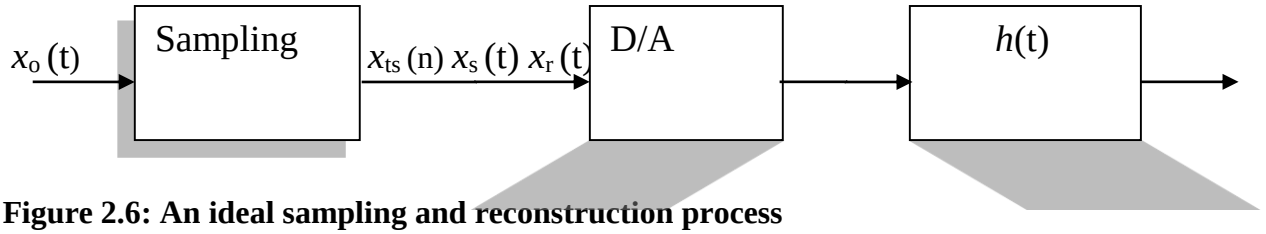

Figure 2.6: An ideal sampling and reconstruction process

x_o (t) = original signal

x_{ts} (n) = sampled signal or time-discrete sequence

h (t) = ideal reconstruction low-pass filter

x_r (t) = reconstructed signal

Figure 2.6 shows an ideal sampling and reconstruction system based on the results of the sampling theorem. This consists of a sampling device which produces a time-discrete sequence x_{ts} (n).*h* (t), an ideal reconstruction low-pass filter and an analog sinc filter, with

$h(t) = \left(\frac{t}{Ts}\right)$. This analog filter cannot be applied directly to the time-discrete signal. The delta function is used to solve this challenge by turning the sequence into an analog signal $x_s(t)$. This is expressed as in Equation 2.14 below:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x_{ts}\ (n)\delta(t - nT) \qquad 2.14$$

According to the sampling theorem, this system will only produce an output or reconstructed signal $x_r(t) = x_o(t)$ when the sampling frequency f_s is at least twice the highest frequency of the original signal $x_o(t)$.

To achieve a practical reconstruction system, only finite length pulses must be inputted into the reconstruction filter. This is achieved by the use of an operation known as the "Hold" operation (Zisselman *et al.,* 2018).

2.7 Digital Signal Processing for Characterisation of Damages in Pipes

Every "real world" signal produced by an event/occurrence/source be it Bio-Electric (EEG, ECG); Electromagnetic (Radio, Radar); Pressure (Speech, Music, Sonar, Drilling, Explosion); Image (Camera, MRI), or others (Seismic) has its own special characteristics. As a result of the special characteristics each of these signals possesses, the appearance of each signal's wave spectra is completely different from the other. In other words, the wave spectra of a signal caused by pressure from an explosion will be completely different from the wave spectra of a radio signal caused by an electromagnetic event. This is shown in the Figures 2.7 and 2.8.

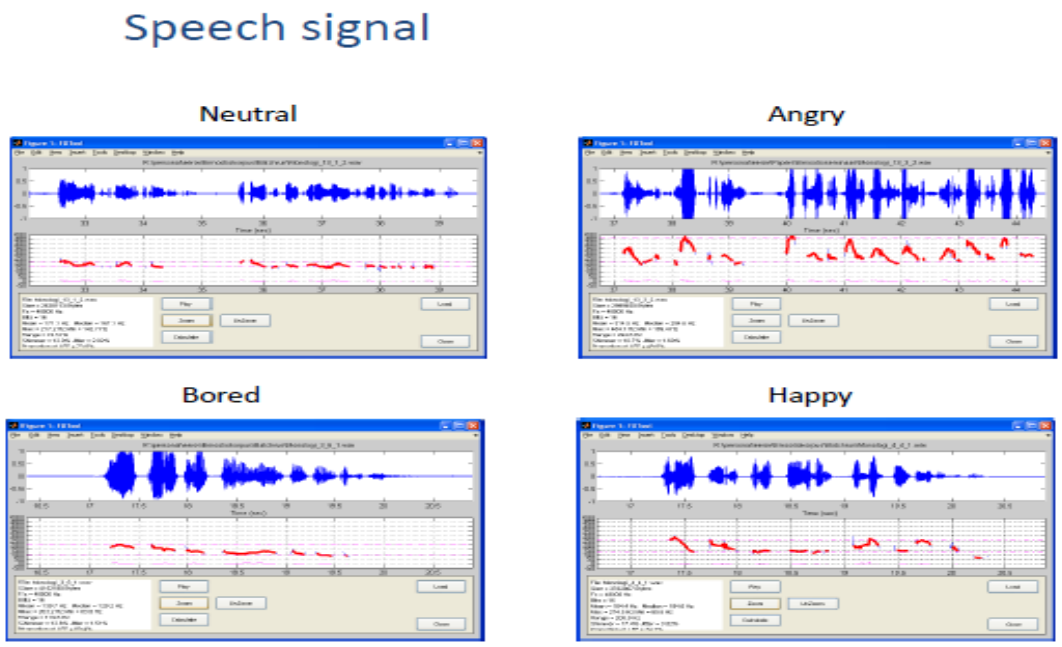

Figure 2.7: Wave Spectra of speech signal from a human in a neutral, angry, bored and happy state respectively (Seppanen, 2016)

It can be seen from Figure 2.5 that the wave spectra of a speech signal from an angry person is completely different from that obtainable when the person is happy or bored.

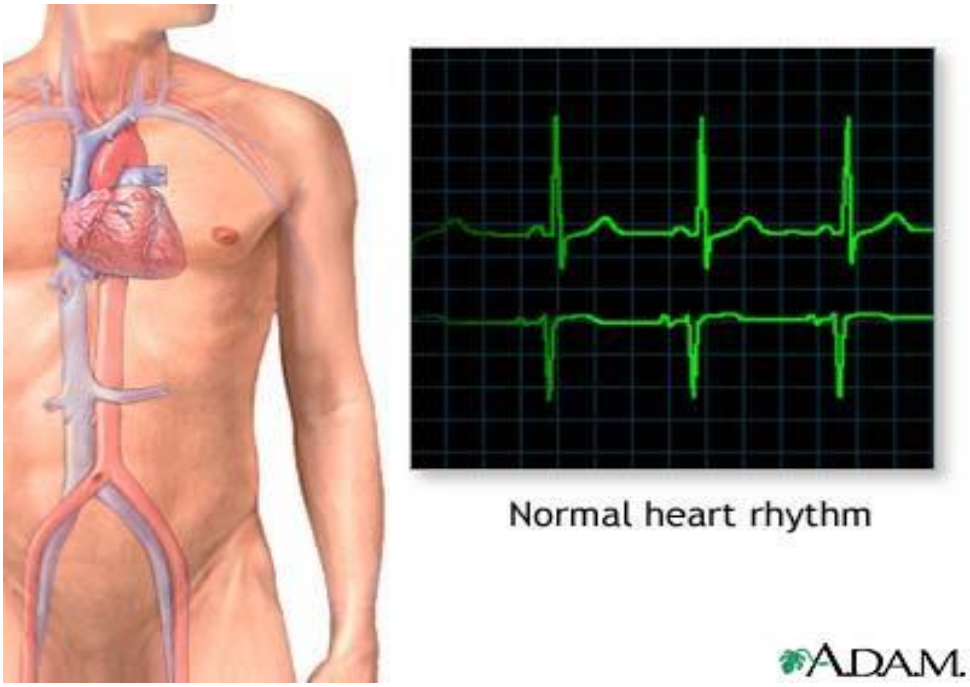

Figure 2.8: Wave spectra of bio-electric ECG signals showing a normal heart rhythm (A.D.A.M., 2017)

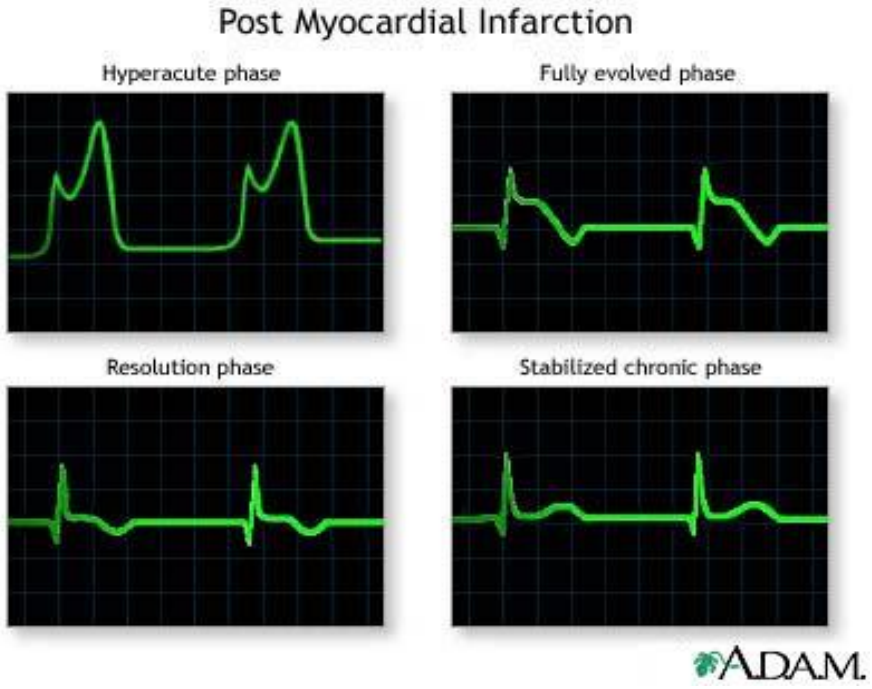

Figure 2.9: Wave spectra of bio-electric ECG signals showing post myocardial infarction (A.D.A.M., 2017)

Also from Figure 2.8, it can be seen that the wave spectra of bio-electric ECG signals of a person with a normal heart rhythm is completely different from that obtainable from a person with experiencing a hyper-acute phase or fully-developed phase of post myocardial infarction as shown in Figure 2.9.

This is the whole idea behind the characterisation of damages in pipes using digital signal processing. Digital signal processing methods would be applied to a pipe that has undergone a damage causing impulsive event to process these signals from their analogue states to their digital states. Filtering of these signals would also be done to remove unwanted frequencies like those caused by noise and then generate the wave spectra of these signals. These DSP methods in order to fully characterise damages in pipes would be able to carry out Pattern Recognition through:

- Abnormal signals (with event introduced) versus the normal signals (with no event)
- Using an average of several outputted normal waveforms as the template
- Detection of new waveforms, segmentation and comparison to the normal template

Pipelines are considered as perhaps the most secure means of shipping oil based goods. They are damaged from time to time as a result of natural events (erosion, earthquakes, etc) or due to third party activities (explosions, drilling activities, vehicular movement, etc). A great challenge that pipeline operators has faced in the past regardless of the fact that the occurrence of damage event had been reported, is the difficulty in pin-pointing what exactly led to the event. Severally, an intrusion on a pipe leads to the flow of pressure pulses in opposite directions through the pipe regardless of the fluid within the pipe. Detection and measurement of these pressure pulses can be carried out at sensors distances away from the

actual location of the event. The pressure pulses released due to the event are very informative as they can be used to determine the cause of the damage.

The information about damage-causing impulsive event contained in these measured pressure pulses can be outputted through proper analysis of the pressure pulses (signals). Therefore, to achieve the characterisation of damage events in pipes, proper signal analysis of measured pressure pulses is carried out.

2.8 Petroleum pipeline monitoring using Internet of Things (IoT)

The oil and gas sector has been slow to embrace IoT technology despite having pipelines and refining facilities, and instrumentation on drilling rigs for decades. Only recently has the extraction industry begun to work with modern IoT (Gold, 2019). This change, is in part due to energy prices that have taken a hit in recent times; and most especially due to the coronavirus pandemic. There is an urgent need for oil-producing nations to safeguard their oil pipeline facilities from oil saboteurs in a bid to save their earnings from oil and gas production. This will in no small measure contribute to the funding of their national budget and boost their foreign exchange earnings (Tomiwa *et al.*, 2020). Oil companies have been working to cut costs; and integration and automation is one of the easiest places to achieve this. An IoT solution that would be able to tie all these different threads of data together has now become a viable option for petroleum companies seeking to minimise human error and obtain real-time insights from the wide range of instrumentation present on the average petroleum pipeline (Gold, 2019).

It was mentioned in (Hill, 2019) that integrating IoT with inputs from experts who can access the live data remotely and provide input via several video streaming channels would allow

pipeline operator access the right information at the right time, with the best analysis which would enable them to move from a reactive to a proactive/predictive operational stand point.

IoT can play several roles in monitoring of pipelines. Operational data from electric submersible pump can be monitored to detect potential failure and automatically stop the pump to prevent damage, and in-turn notify operators to repair or replace the pump based on current machine and maintenance models (Ayn, 2019). IoT can also be used in pipeline optimisation, where it can shut down a valve and send an alert to a mobile device to avoid a major disruption or damage a pipeline (Ayn, 2019).

There are many benefits of using IoT for pipeline control. Without IoT, businesses will have to rely on humans to perform routine checks and maintenance, according to Joshi, (2019). Because of its ability to track pipelines in real-time, the IoT framework assists in the reduction of manual tests. The real-time data can be used to reduce significant risks associated with pipeline leaks and other undesirable circumstances. According to Joshi, (2019), another benefit of using IoT in pipeline control is the efficient management of employees. Employees would only be needed to perform repairs when an issue arises, reducing the need for annual human monitoring and human resources.

To the best of the author's knowledge, not much works exits in the use of IoT for monitoring of petroleum pipelines but related works to the scope of this study exists. Notably, Cheddadi et al. (2020) proposed a low-cost IoT device for collecting and tracking electric and environmental data from a PV solar station in real time. In order to collect, process, store, and analyse data, a low-cost data pipeline for monitoring environmental and electrical parameters in a photovoltaic station was built. The ESP32 DEVKIT V1 was used as the

microcontroller in the proposed monitoring system for collection and processing of incoming data from sensors before transmitting the processed data to the cloud through built-in Wi-Fi.

Also, Ibrahim *et al.,* (2019) proposed an IoT-based greenhouse monitoring and remote control architecture that is applicable to various types of crops. The architecture that was proposed in this work is capable of allowing owner/supervisors to remotely control the green house through the internet and enable the autonomous control of the greenhouse operational conditions. The proposed architecture also allows the owner to monitor and keep record of the progress all through the period of plantation of the crops within the greenhouse. The simulation of the proposed system architecture was carried out using the Riverbed Modeller.

2.9 Summary of Literature Review and Research Gaps

The current state of pipeline monitoring and security in Nigeria has been discussed. It was found that pipeline monitoring in Nigeria and securing of these pipelines is still a challenge to pipeline operators as oil spill due to third party interference and operational faults still occur. Monitoring of pipelines in Nigeria is done through intermittent appraisal of pipelines, use of pipeline integrity management systems, on-the-ground and air surveillance of pipelines by security forces. High cost, planning complexity, and lack of proper access routes are major drawbacks for these monitoring methods.

Leak identification is presently being carried out in Nigeria via visit to the site of the spill to determine the cause of the damage to the pipeline. A joint Investigation team pays a visit to the site of the oil spill to determine the volume of oil spilled and the cause of the spill. The investigation team normally comprises of persons from oil companies, the federal government, state governments, regulators, security agencies and communities. The

drawback for this type of method is that severe loss of product and serious environmental damage would have taken place before repair works would commence due to time wastage. As at the end of 2018, 234 oil spill sites in Nigeria remained un-remediated (Shell in Nigeria, 2018). Huge cost of transportation of investigation team is another drawback.

Methods of pipeline line monitoring from previous research have been discussed. The merits and demerits of these methods have also been highlighted. Vibration-based methods for pipeline monitoring have been found to be very effective. The method of time delay between pressure pulse arrivals was discovered to be very successful in locating leaks on a pipeline. Due to unwanted interference noise from traffic, water, wind and other sources, the acoustic method of pipeline monitoring was found to be not very efficient in the determination of a leak in a pipe. Transients inaccurate modelling was found to be a major drawback of the inverse least square method of pipeline monitoring. The pipe network boundary conditions was found to be another major drawback.

Vibration based methods have been used to monitor pipelines in the past through the process of mathematical modelling. These mathematical models have proven to be successful in detecting and locating damages along a pipeline. As is the practice in literatures reviewed, the developed mathematical model is always simulated to predict the accuracy of the model. The simulation is then verified using built test rigs. The size of the built test rigs differ from reviewed literatures. Also, several types of sensors were used in past studies.

Several signal processing techniques were discussed highlighting their uses and strengths. The Fourier transform was found to be very important in revealing hidden information in a

signal. It was also found to be very important in calculating Cross-Correlation Functions (CCFs) between unsymmetrical images to help averaging and comparing them.

To the best of the author's knowledge and from literature reviewed in the course of this study, previous research that has been done on pipeline monitoring has not taken adequate consideration in characterisation of the causes of damages or leaks on pipelines. Potential causes of damages on pipes like drilling, corrosion, use of a heavy mass were rarely taken into consideration. The research works were also based on operational causes of damage. Exhaustive studies have not been carried out mimicking these different damage causes to classify and characterise them in order to determine the cause of a leak on a pipeline even before arrival at the spill site.

Also, to the best of the author's knowledge and from literature reviewed in the course of this study, the reconstruction of the original pressure pulse at the point of damage from a combination the pressure pulses at different damage points along a pipeline has rarely been researched on. Most of the research works on event reconstruction were based on the reconstruction of an original pressure pulse from a single pulse recorded along the pipe. The Fourier transform methods of sampling and reconstruction are proposed to help achieve this. Combination of reconstruction methods to achieve the best possible estimate of the original pulse magnitude was rarely investigated. Research works on pipeline monitoring have focused on achieving event reconstruction using different methods. The use of a factor along with the Fourier transform method of signal reconstruction to solve the problem of under estimation and over estimation of original pulse magnitude is hereby advocated for, for its simplicity and versatility.

Previous research works on pipeline monitoring have rarely focused on real time transmission and monitoring of damage data wirelessly to an Internet-of-Things (IoT) platform to as a means of optimising pipeline monitoring systems. Previous research works have focused on the design/development of intelligence pipeline inspection gauges (pigs) to achieve this. Available in the world today is about 2.5 million km of hydrocarbon pipeline. That is enough to go around the earth's circumference at least 62 times. For different reasons, a good percentage of pipelines across the world are considered impossible to pig (Oil and Gas IQ, 2015). These reasons include:

- Tight bends in the pipe do not allow the rigid exoskeleton of the pig to pass through.
- Blockages caused by sediments and contaminants may act as barriers to the path of a pig.
- Pipes may have a number of different diameters which prohibit the passage of these torpedo-like structures.
- There are valves in the pipeline that permanently obstruct the passage of anything but a gas or fluid.
- There may be no direct entry into a pipe.

The process of pigging is a very expensive one. Estimates have shown that pipeline monitoring and inspection by pig can cost as much as $56,000 per kilometer. Taking into consideration that 25 percent of the world's pipelines fall into the 'un-piggable' class, it can be estimated that companies are spending close to $105 billion on pigging the world's hydrocarbon pipelines. This is more than the annual gross domestic product of a lot of countries. Creating a pigging system for pipeline inspection and monitoring is a very messy

and labor intensive process. A very demanding planning process is required to make sure that operations are maintained within HSE parameters because most pigging processes run whilst pipelines are in service. Trained crew may require hours to correctly load the pig into the pipe after the planning is complete; and the running distance will only stretch to a handful of kilometers. As a result of this, depending on the pipe, before an inspection pattern can emerge, pigs may need to be launched severally. Smart pigs use intelligent technology such as transmitters, sensors, GPS, eddy current, magnetic fields, ultrasonic and acoustics to identify and diagnose potential problems (Oil and Gas IQ, 2015).

The use of an Arduino, Wi-Fi module and the ThingSpeak IoT platform is advocated for to achieve real time monitoring of a pipeline from anywhere in the world.

Therefore, the research gaps that have been filled in this work include:

i. The use of an experimental test consisting of a PVC pipe, air/water as the transportation fluid, piezoelectric sensors and a data logger for the characterisation of damages in pipes
ii. The use of the Fourier transform methods of sampling and reconstruction to achieve the reconstruction of an original pulse at the damage point from two other pulses at different points along the pipe
iii. The use of a factor, along with the Fourier transform technique of signal reconstruction to achieve a best possible estimation of the original pulse magnitude.
iv. Combination of an Arduino, Wi-Fi module and the ThingSpeak IoT platform for continuous wireless transmission of damage data and real time monitoring of pipelines

The development of the petroleum pipeline monitoring system in this research work will therefore be based on the combination of signal processing techniques of sampling and reconstruction using Fourier transform; and the method of time delay between pulse arrivals, a vibration-based technique. The system is expected to detect and locate damage on a pipe with a good level of accuracy. It is also expected to be able to classify various damage events that occur on a pipeline.

CHAPTER THREE

3.0 MATERIALS AND METHODS

3.1 Materials

The major materials that were used for this research work are:

i. Flexible polyethylene hose pipe

ii. Water

iii. Sulphuric Acid (AR H_2So_4 M.W. 98.08) with composition Assay/acid metric: 97-99%; wt. per ml at 20^oC about 1.835g

The major equipment for the research and a brief about their use is presented thus in Table 3.1.

Table 3.1: Major Equipment

S/N	Equipment	Specification	Purpose	Source
1	TCAM piezoelectric sensors	Diameter 15 cm; thickness 0.35 mm; model number: 8QQ0302	For detection of propagated pulses along pipeline	Aliexpress
2	Pulse generator	5 Pa rating	For generation of sharp fronted pressure pulses in the pipe	Fabricated
3	Pico Log 1012	10 bits, 12 channels data acquisition module with serial number pl1000.en r2 10.05.2013	Recording and processing of signals from pressure pulses at specified sampling rate	Pico Technology

4	MATLAB® software package (R2016a)	R2016a	Analysis of generated data; calculation of Fourier functions of generated pulses; for sampling and reconstruction of signals	MathWorks
5	Two LOMVUM hand drill	Model Number: 13T; Rated power: 220 V; Output Power: 750 W and 1000 W respectively; Frequency: 50/60Hz; No load speed: 0-4600 R/Min; Max. Chuck: 13 mm; Size: 19 cm and 28 cm respectively	To generate pulses in the pipe	Mechanical Workshop, Federal University of Technology Minna
6	Hammer	Average size, wooden handle	To generate pulses in the pipe	Mechanical Workshop, Federal University of Technology Minna
7	Wireless communication device	Arduino and WiFi Module-based	For processing and transmission of signal data received from sensors wirelessly to the ThingSpeak	Fabricated

			IoT analytics platform	
8	Laptop computer	HP Pro book 720p	Connected to data acquisition module for processing of and visualisation of data.	Personal copy
9	Water tank	100 Liters rubber tank	For storage of water used in experiments	Gidan Kwano market, Minna

3.1.1 Wireless Communication Device

A wireless communication device was developed to carry out wireless transmission of signal data from the sensors to the ThingSpeak IoT analytics platform. The device consisted of an Arduino, and a Wi-Fi module which were integrated with the sensors that were placed on the pipelines. The Arduino was programmed to communicate with the ThingSpeak IoT analytics platform thereby sending signal data to the platform for visualisation and analysis. A circuit diagram showing how the Arduino and Wi-Fi module were connected is shown in Figure 3.1.

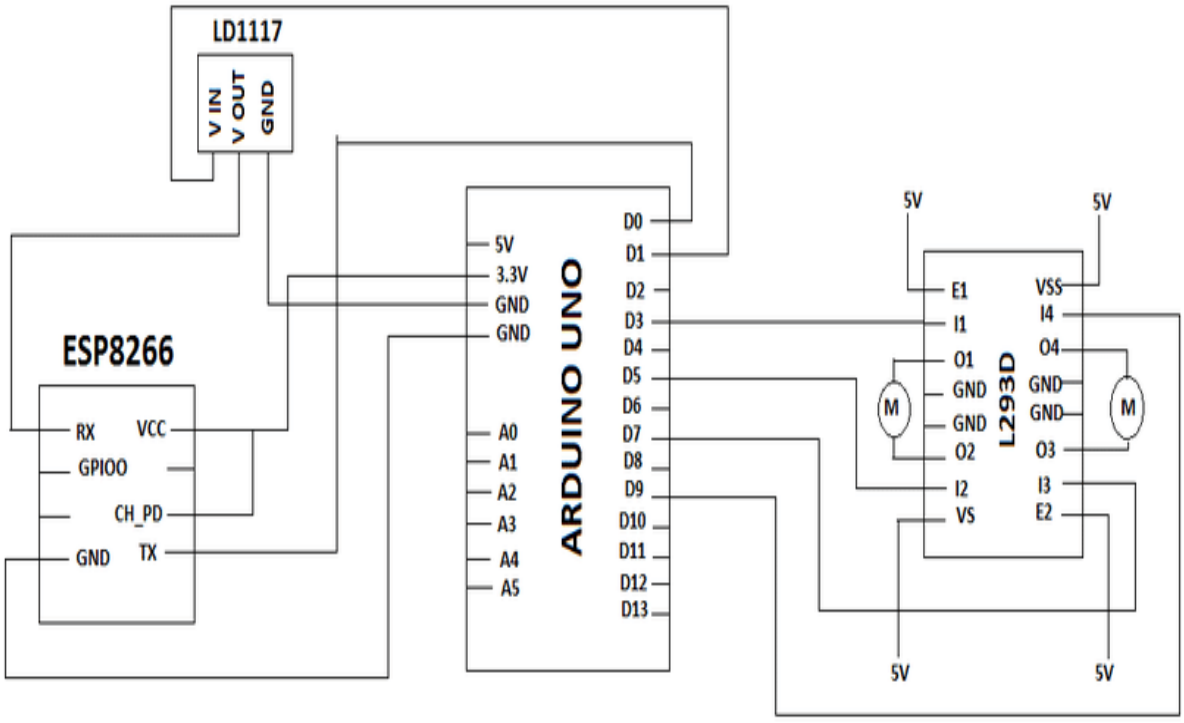

Figure 3.1: Circuit diagram of connection between Arduino and Wi-Fi module (Raj *et. al.*, 2015)

3.1.1.1 The Choice of Micro Controller

An Arduino was selected as the micro controller for this study. An Arduino is an open-source electronics platform based on easy-to-use hardware and software. Inputs like light on a sensor, a finger on a button, or a Twitter message can be read by an Arduino board and then turned it into an output like activating a motor, turning on an LED, publishing something online (Hobby Electronics, 2019).The Hwayeh CH34og +MEGA 328P Arduino shown in Plate I was used in the development of the wireless communication device. It made use of an Atmel 328 microprocessor controller. It was selected because of its good performance, low power consumption, real timer counter having separate oscillator, and is programmable (Hobby Electronics, 2019). The Arduino was programmed to collect data once every 15

seconds and update the channels on the analytics platform once every 2 minutes as seen in Appendix B. The reason for this is that the student version of the IoT platform was used and as such, only permit data collection every 15 seconds, though an Arduino is able to collect data even in milliseconds. The Arduino code is called a ‘sketch’ which is a short program that is run over and over by the device.

Plate I: Hwayeh CH34og +MEGA 328P Arduino (Hobby Electronics, 2019)

3.1.1.2 Wi-Fi Module

The ESP01 ESP8266 Wi-Fi module as shown in Plate II was used for developing the wireless communication device. It was selected for this study due to its powerful storage and on-board processing capabilities, the ESP8266 can be integrated with sensors and other application-specific devices through its GPIOs with minimal development and load during runtime. It allows for minimal external circuitry due to its high degree of on-chip integration, and the entire solution, including the front-end module, is designed to take up as little PCB space as

possible (Gamma, 2019). It was programmed via written codes to communicate with the Arduino. The codes are seen in Appendix C.

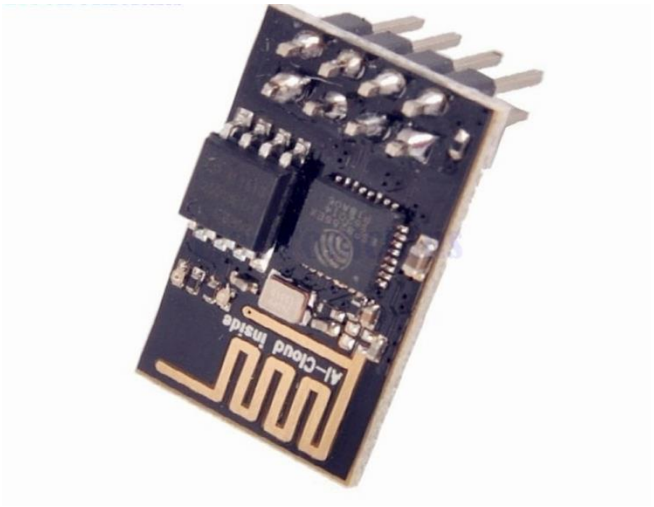

Plate II: ESP01 ESP8266 Wi-Fi module (Gamma, 2019)

3.2 Methods

3.2.1 Damage Location Based on Pressure Pulse Analysis

When a pipe is damaged, the flow of a pressure pulse to and from the point of damage occurs. These pressure pulses are eventually reflected when they reach the boundaries of the pipelines. In this work, the position of a damage on a pipe with air and water as the transport fluids respectively was determined. This required placing sensors at various points along the pipe. The travel times these pressure pulses can be found when a high frequency is used for sampling when pressure measurements are made at several points along the length of the pipe (Olugboji, 2011).

3.2.2 Propagation of Damage-Induced Pressure Pulses in a Water Filled Pipe

The pulse arrival times in a pipe and the sensor positions along the pipe can be used to determine the location of an event along a pipe. These events could either be caused by drilling, impact, explosion, etc. In Figure 3.3, the schematic representation of a pipeline with four sensors placed along it is shown. The sensors: 1, 2, 3, 4 were distances x_1, x_2, x_3, x_4 respectively from some boundary. The arrival times of some generated pulses caused by a damage-inducing impulsive event occurring at an unknown location is recorded as t_1, t_2, t_3, and t_4 by the four sensors.

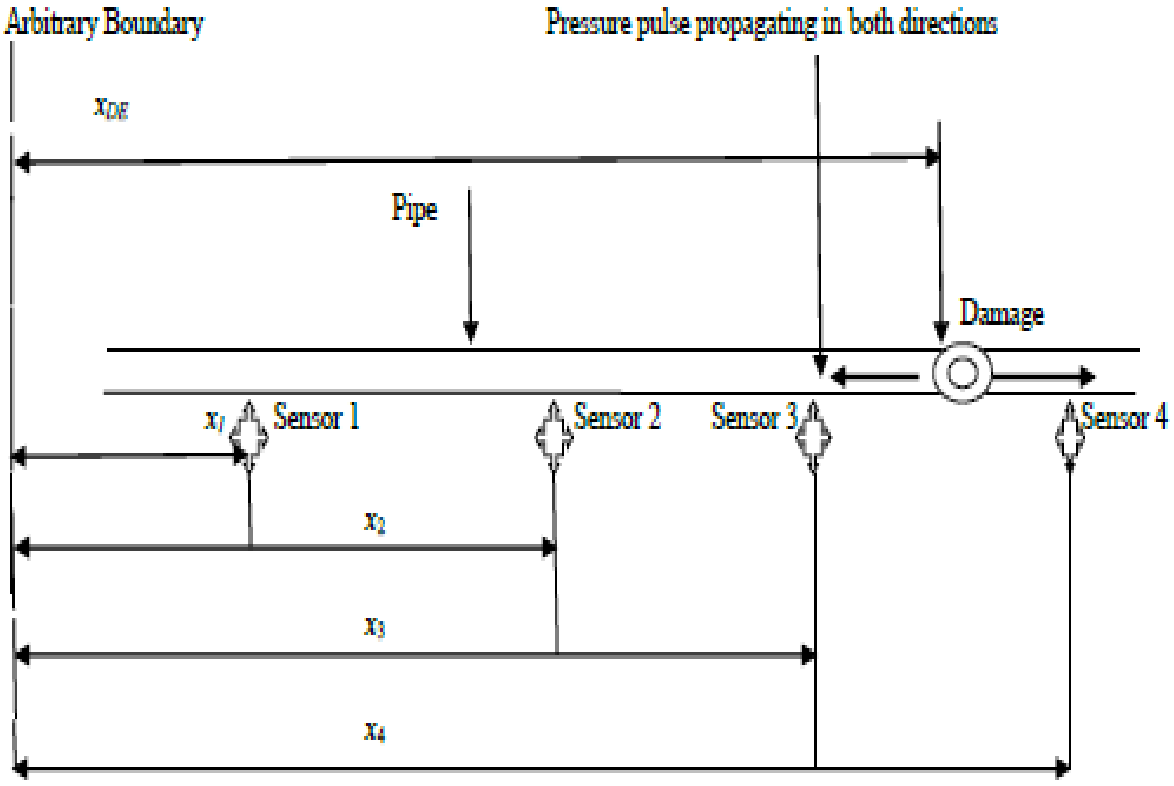

Figure 3.2: Schematic representation of sensors on a pipeline.

In the pipe seen in Figure 3.3, sensor 3 or 4 may be used to determine the event location. The occurrence of a damage event on a pipe leads to a change in pressure within the pipe which eventually leads to the generation of pressure pulses. Any change in pressure moves through the fluid-filled pipe at a velocity c_p which is manifested as a pressure pulse in both directions. The propagation velocity of a pressure pulse in a fluid-filled pipe is determined by the elastic properties of the fluid, as well as the material and the geometry of the pipe. In an elastic pipe, the pulse propagation velocity of the elastic fluid within it is given as (Záruba, 1993; Finnemore and Franzini, 2002):

$$C_p = \sqrt{\frac{1}{\rho\left(\frac{1}{Ev} + \frac{ID}{Wt\,E}\right)}} \qquad 3.1$$

Where ρ = density of water (kg/m^3)

E_v = water bulk modulus (N/m^2)

ID = pipe inner diameter (mm)

W_t = thickness of wall (mm)

E = Pipe modulus of elasticity (N/m^2)

The arrival of pulses at the same side of the event which are sensors 3 and 2, or sensors 2 and 1 as in the case shown in Figure 3.3 can be adopted for determining the velocity of the pulse propagation. Thus, the exact location of the event from sensor 3 is calculated by:

$$x_{DE3} = \frac{(\mathrm{t34Cp} + x43)}{2} \qquad 3.2$$

or from sensor 4,

$$x_{DE4} = \frac{(x43 - Cp\mathrm{t34})}{2} \qquad 3.3$$

To determine the arrival times of signals at each sensor, the cross correlation technique is used.

3.3 Experimental Validation

An experimental flow loop as shown in Figure 3.4 was built for validating the above discussed event location theory. This consisted of an air-filled PVC (Polyvinyl Chloride) pipe of total length 20.11 m and internal diameter of 20 mm for propagation of pressure pulses. A pressure pulse generator was used to introduce sharp-fronted pulses into the water filled pipe. Five piezoelectric sensors were situated at various points on the PVC pipe. Sensor 1 was located at 2 m from one end; while sensor 2 was located at 5 m from one end and sensor 3 was located at 8 m from one end. Sensors 4 and 5 were located at 11 and 17 m from one end respectively. All sensors were connected to a single Pico Log data instrumentation system. This was used for capturing and recording of pulse data in the first experimental test rig. A second test rig was built with all the sensors connected to the wireless communication device. Both test rigs are shown in Plate I and II. The location of the event was 4.32 m from one end of the pipe.

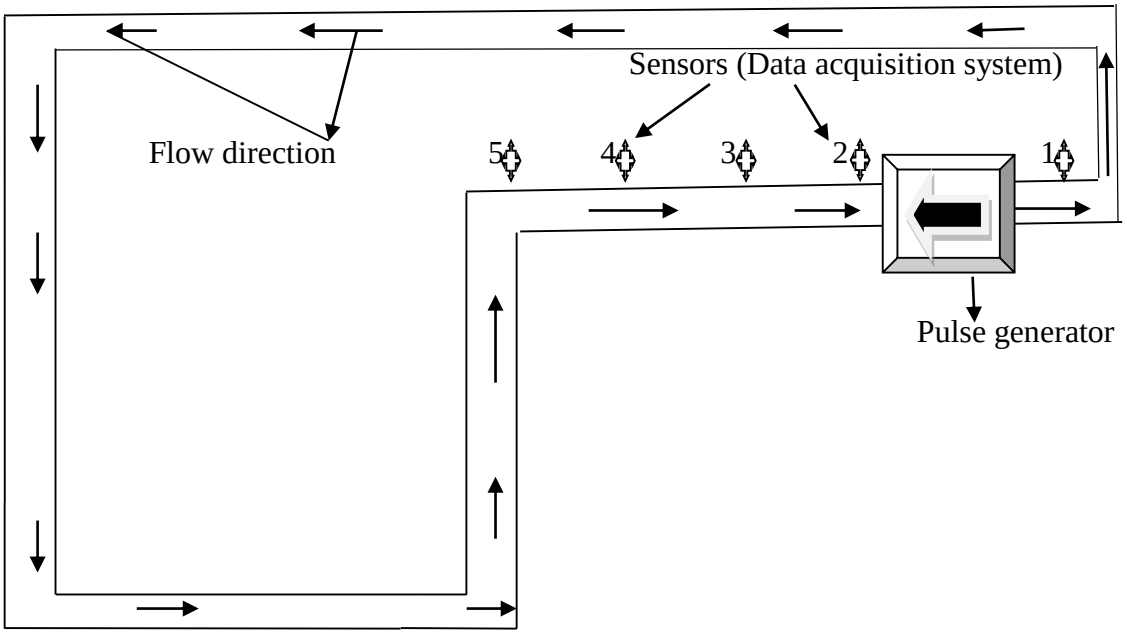

Figure 3.3: Schematic of experimental flow loop

Sensor 1 was located at 2.24 m from the event location; sensor 2 was located at 0.68 m from the event location; sensor 3 was located at 3.68 m from the event location; sensor 4 was located at 6.68 m from the event location, and sensor 5 located at 12.684 m from the event location. Pressure pulses were generated at specific pressures severally and measurements made for each amount of pressure set in the pulse generator. The velocity of the propagated pulse was experimentally determined using the pulse data from any two sensors. The MATLAB® software was used to carry out cross-correlation of the measured pressure pulses to determine time delay in arrival times of these (t_{delay}). The velocity of pulse propagation was calculated as (Olugboji, 2013):

$$C_p = \frac{x_{ab}}{t_{delay\ ab}} \qquad 3.4$$

Where x_{ab} = distance between two selected sensors

$t_{delay\ ab}$ = time delay between arrival times at two selected sensors

This calculation was done severally to examine the repeatability of pressure pulse velocity value. The average value of experimental pulse propagation velocity was determined from plots made graphically and compared to that obtained using Equation 3.1 to validate its accuracy. These experimental values of velocity were then used in Equations 3.2 and 3.3 to calculate the various event locations. An average of these locations was calculated and compared against the actual measured event location to determine the accuracy of the method used.

3.4. Numerical Simulation and Modeling of Pressure Pulse Propagation for Characterisation

In this work, emphasis was based more on outputting, analysing and categorising the wave spectra obtainable from different impulsive or damage causing events. Fourier transform ranks highest when it comes to linking the physics of a signal producing system to the signal characteristics. The majority of physics laws can be expressed using differential equations. This is a major advantage of Fourier transform as it can use differential equations to provide comprehensive and general solutions to physical systems.

3.4.1 Mathematical Model

For this work, the Fourier Transform was employed as the basis to develop the mathematical model needed for the reconstruction of the signals arising from various impulsive events. This is discussed in the following sections.

3.4.1.1 Mathematical Model for Sampling

The following notations were adopted in the course of this work:

x_o (t) = original analog signal

x_r (t) = reconstructed signal (where x_r (t) = x_o (t))

x_{ts} (n) = time-discrete signal or sampled signal $(where\ x_{ts}\ (\mathrm{n}) = x(nTs))$

fs = sampling frequency (samples/second)

T_s = sampling interval (seconds/sample) $\left(where\ T\mathrm{s}\ =\ \left(\frac{1}{f_s}\right)\right)$

ὠ = real angular frequency (radians/second)

ω = digital angular frequency (radians/second) (where ω = ὠTs)

t = continuous time variable

n = discrete integer variable

The inverse Fourier transform of a time discrete signal x_{ts} (n) obtained from sampling x_o (t) every Ts second is given by:

$$x_{\mathrm{ts}}(\mathrm{n}) \quad = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_{ts}\left(e^{i\omega}\right)e^{i\omega n}d\omega \qquad 3.6$$

In simplifying the above, the equation is expressed in terms of real angular frequency, ὠ. This gives:

$$x_{\mathrm{ts}}\ (\mathrm{n}) = \frac{T_S}{2\pi}\int_{-\frac{\pi}{T_S}}^{\frac{\pi}{T_S}} X_{ts}\left(e^{i\varphi T_S}\right)e^{i\varphi T_S n}d\varphi \qquad 3.7$$

For a continuous signal, the inverse Fourier transform given by:

$$x_{\mathrm{o}}\ (\mathrm{t}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_o(i\varphi)e^{i\varphi t}d\varphi \qquad 3.8$$

Replacing t with nTs in the above equation, we obtain

$$x\,(\mathrm{n}T\mathrm{s}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_o(i\varphi)e^{i\varphi nT_s}d\varphi \qquad 3.9$$

Splitting the integration in (3.9) into sub-intervals of length $\frac{2\pi}{T_s}$ and taking the sum over the resulting integrals to obtain the complete area, we have:

$$x\,(\mathrm{n}T\mathrm{s}) = \frac{1}{2\pi}\sum_{k=-\infty}^{\infty} \int_{\frac{(2k-1)\pi}{T_s}}^{\frac{(2k+1)\pi}{T_s}} X_o(i\varphi)e^{i\varphi nT_s}d\varphi \qquad 3.10$$

Changing the integration variable by setting $= \alpha + \frac{2*\pi k}{T_s}$, we obtain:

$$x\,(\mathrm{n}T\mathrm{s}) = \frac{1}{2\pi}\sum_{k=-\infty}^{\infty} \int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}} X_o\left(i\left(\alpha + \frac{2*\pi k}{T_s}\right)\right)e^{i\left(\alpha+\frac{2*\pi k}{T_s}\right)nT_s}d\alpha \qquad 3.11$$

Multiplying (3.11) all through by $\frac{T_s}{T_s}$, with $\alpha = \varphi$ and noting that $e^{i2*\pi kn}$, we obtain:

$$x\,(\mathrm{n}T\mathrm{s}) = \frac{T_s}{2\pi}\int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}}\sum_{k=-\infty}^{\infty} \frac{1}{T_s}X_o\left(i\left(\varphi + \frac{2*\pi k}{T_s}\right)\right)e^{i\varphi nT_s}d\varphi \qquad 3.12$$

In order for x_{ts} (n) to be equal to x (nTs) for all values of the integer n, equations 3.7 and 3.12 must agree as given below:

$$X_{ts}\left(e^{i\varphi T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X\left(i\left(\varphi + \frac{2\pi k}{T_s}\right)\right) \qquad 3.13$$

The above is the mathematical model for sampling of signals to obtain their digital spectrum.

3.4.1.2 Mathematical Model for Signal Reconstruction

The inverse Fourier transform for a band-limited signal is given by:

$$x_o(t) = \frac{1}{2\pi}\int_{\frac{-\pi}{T_S}}^{\frac{\pi}{T_S}} X\ (i\varphi)e^{i\varphi t}d\varphi \qquad 3.14$$

$X_{ts}\left(e^{i\varphi T_s}\right) = \frac{X(I\varphi)}{T_S}$ for the interval being integrated and when this is substituted into equation 3.14, we obtain:

$$x_o(t) = \frac{T_s}{2\pi}\int_{-\frac{\pi}{T_S}}^{\frac{\pi}{T_S}} X_{ts}\left(e^{i\varphi T_s}\right)e^{i\varphi t}d\varphi \quad 3.15$$

When the DTFT expression for $X_{ts}\left(e^{i\varphi T_s}\right)$is applied, we obtain:

$$x_o(t) = \frac{T_s}{2\pi}\int_{-\frac{\pi}{T_S}}^{\frac{\pi}{T_S}} \sum_{k=-\infty}^{\infty} \quad x_{ts}(n)e^{(-i\varphi nT_s)}e^{i\varphi t}d\varphi \quad 3.16$$

With summation and integration interchanged, we obtain:

$$x_o(t) = \frac{T_s}{2\pi}\sum_{k=-\infty}^{\infty} x_{ts}(n)\int_{-\frac{\pi}{T_S}}^{\frac{\pi}{T_S}} e^{i\varphi\,(t-nT_s)}d\varphi \quad 3.17$$

Integrating, we obtain:

$$x_r(t) = x_o(t) = \sum_{k=-\infty}^{\infty} \quad x_{ts}(n)\frac{\sin\left(\frac{\pi}{T_s}(t-nT_s)\right)}{\frac{\pi}{T_s}(t-nT_s)} \quad 3.18$$

The above is the reconstruction model that was used to recover the original signal x_o (t) from the time-discrete sequence or sampled signal x_{ts} (n).

To obtain a more accurate reconstruction of the original signal x_o (t), we obtain:

$$x_r(t) = \sum_{k=-\infty}^{\infty} x_{ts}(n)\frac{\sin\left(\frac{\pi}{T_s}(t-nT_s)\right)}{\frac{\pi}{T_s}(t-nT_s)} * K \quad 3.19$$

Where K is an approximation factor obtained by dividing the maximum value of $x_o(t)$, by the maximum value of $x_r(t)$. Equation 3.19 is the mathematical model that was developed to solve the problem of overestimation of a pulse that was reconstructed from the combination of two other pulses along the pipeline. K is obtained by dividing the maximum amplitude of the reconstructed pulse at a particular sensor by the maximum amplitude of the original pulse at the same sensor. These are obtained from plots in the MATLAB® environment.

3.5. Simulation of Mathematical Model

The simulation of these mathematical models developed for both the sampling and reconstruction of signals obtained from pressure pulses' propagating within a pipe was carried out using the MATLAB® software. This was done to ascertain the workability of the model and its ability to perform proper event reconstruction.

A mathematical model was derived to determine the location of a damage event along a water-filled pipe. Another mathematical model was developed to determine the velocity of propagated pulse from experimentally measured pulse pressure at any two sensors using the principle of delay in the arrival times of pressure pulses at sensors.

Mathematical techniques were also derived to carry out signal sampling and reconstruction. Preliminary testing of these mathematical models was carried out with the model being applied to three sample sets of data obtained from a different experimental setup carried out on pipeline monitoring. The sampling for each data sample was done after every 10 data sets to obtain a total of 250 samples. This simulation was carried out using the MATLAB® software. The results obtained show that this model can exactly reconstruct a signal from from values sampled at discrete, uniform intervals as long as the signal frequency is less than half the sampling frequency, fulfilling the Nyquist frequency criterion. The simulation was done in accordance with the procedure carried out by Olugboji, (2011).

3.6. Experimentation for Validation of Simulation

For the validation of the proposed methods for damage characterisation in pipes, four different experiments were carried out. These four experiments were carried out to mimic damage events in pipes caused by explosion, drilling, vehicular movement, and corrosion respectively.

3.6.1 Experimentation to Mimic Damage Event Caused by Explosion

The experimental set up was basically the same as that shown in Figure 3.3 and Plates III and IV. A pulse generator was used to mimic an explosion in the pipe. Piezoelectric sensors that were linked to one data logger were placed at points along the pipe, and the data logger was used to capture pulse data and saved in a tab delimited data file. Analysis of the captured data was done offline in the MATLAB® environment with written codes.

These experiments were carried out with air and water as the transport fluid respectively within the pipe, and different pressure rates were used in both experiments. A pulse generator with a hand pump connected to it was used to generate sharp fronted pulses into the pipe. This was done repeatedly at pressure readings from 0.2 bar to 1 bar within the pulse generator. A total of 50 tests were carried out and all the measured pressure pulses had similar shapes.

The pressure pulse with the highest amplitude was obtained using a pressure reading of 1.0 bar in the pulse generator, while the pressure pulse with the least amplitude was obtained using a pressure reading of 0.2 bar. 50 measurements were made in total because 10 measurements each were made for each pressure rating in the pulse generator. An average of the time delay between pulse arrivals was obtained after cross correlation in MATLAB®

3.6.2 Experimentation to Mimic Damage Event Caused by Drilling

The experimental setup was the same with that of explosion. To mimic a drilling operation, sharp fronted pressure pulses were introduced into the PVC pipe by using a hand drill to drill holes of 4 mm diameter each at different points along the surface of the pipe. The signals from the sensors were recorded using the data acquisition model and saved. The saved data was analysed in the MATLAB® environment with written codes.

With the same experimental test rig, damage from a drilling event was mimicked. Two different hand drills were used to generate sharp fronted pulses into the pipe. The drills had power ratings 750 Watts and 1000 Watts respectively. This experiment was carried out with both air and water as the transport fluid in the pipe respectively. The drills were used to make holes along the pipe

3.6.3 Experimentation to Mimic a Potential Damage Event Caused by Vehicular Movement

With a similar experimental setup as that of explosion and drilling, this experiment was carried out by using a hammer to strike at points along the pipe. The hammer struck the pipe at an average of 445 N per strike. The solid mass was meant to mimic a sudden weight like that obtained from a vehicle. The signals from the sensors were also recorded using the data acquisition device and saved. The saved data was then analysed in the MATLAB® environment with written codes.

To mimic potential damage to a pipe caused by movement of the tires of a vehicle over the pipe, a hammer was used to impart a mass on the pipe of the experimental test rig thereby introducing sharp-fronted pulses into the pipe. Air and water were also used respectively as the transport medium in the pipes

3.6.4 Experimentation to Mimic a Potential Damage Event Caused by Corrosion

The experimental setup is as illustrated in Figure 3.5. To mimic a corrosive environment in a pipe, sharp fronted pressure pulses were generated into the water filled PVC pipe by carefully introducing 97-99% concentrated Sulphuric acid at different points along the length of the pipe. The signals from the sensors were recorded using the data acquisition model and

saved. Analysis of the saved data was carried out using the MATLAB® software through written codes.

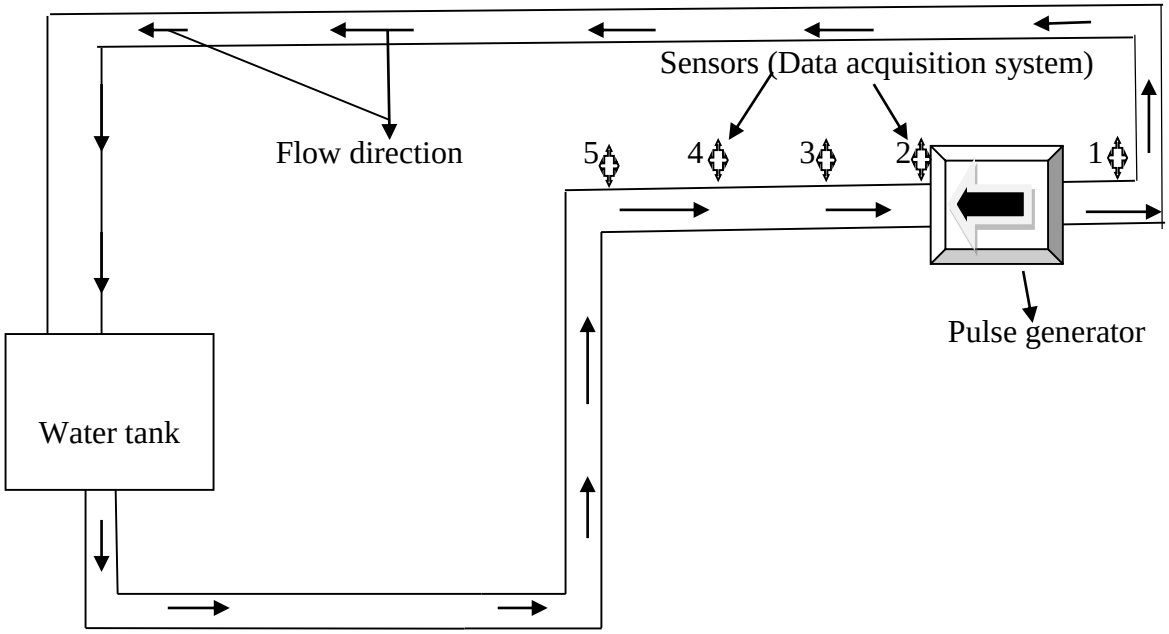

Figure 3.4: Schematic of experimental test rig for corrosion

The setup in Figure 3.4 was also adopted in mimicking damage due to drilling, vehicular movement and explosion in order to compare results between scenarios where air and water are the transport fluids within the PVC pipe and to see what is obtainable in both cases. This test rig is shown in Plate III.

In all the four experiments, a voltage rating of 100 mV on the Pico Log data logger was used. This voltage was used because it was the maximum voltage rating on the data logger that

permitted the outputted pulse spectrums not to be distorted. An algorithm for FFT was written in the MATLAB® software and used to carry out sampling and reconstruction on the collected signal data. These signals in the frequency domain contained noise frequencies. An FFT algorithm for the Sinc filter was also be written in MATLAB® to perform the noise cancellation process. The experiments for each impulsive event were carried out repeatedly. Those to mimic explosion, drilling and vehicular motion were carried out 50 times each, while those to mimic explosion was carried out 5 times. The various wave spectra were outputted after processing and then analysed for pattern recognition.

3.7 ThingSpeak IoT Analytics Platform

ThingSpeak is an IoT platform service that enables live data streams to be viewed and analysed from sensor devices in the cloud (ThingSpeak, 2019). This platform enables you to perform data analysis on data collected from remote devices with MATLAB® codes in real-time. The platform was signed up to, and channels were created on the platform. These channels were configured via written codes to communicate with the desired sensors.

On the ThingSpeak network, five channels were generated to collect data from the five sensors used in the experimental test rig. Every 15 seconds, signal data was obtained, and every 2 minutes, all channels were changed. This was due to the fact that the student license of the platform was used. One second transfer of data, and immediate updating of the platform is only possible using the paid license. ThingSpeak visualised data posted by wireless networking devices in real time. With the ability to execute MATLAB® codes in ThingSpeak, online analysis and processing of the data was performed as it came in. Figure 3.6 illustrates the Internet of Things process that was carried out in this work.

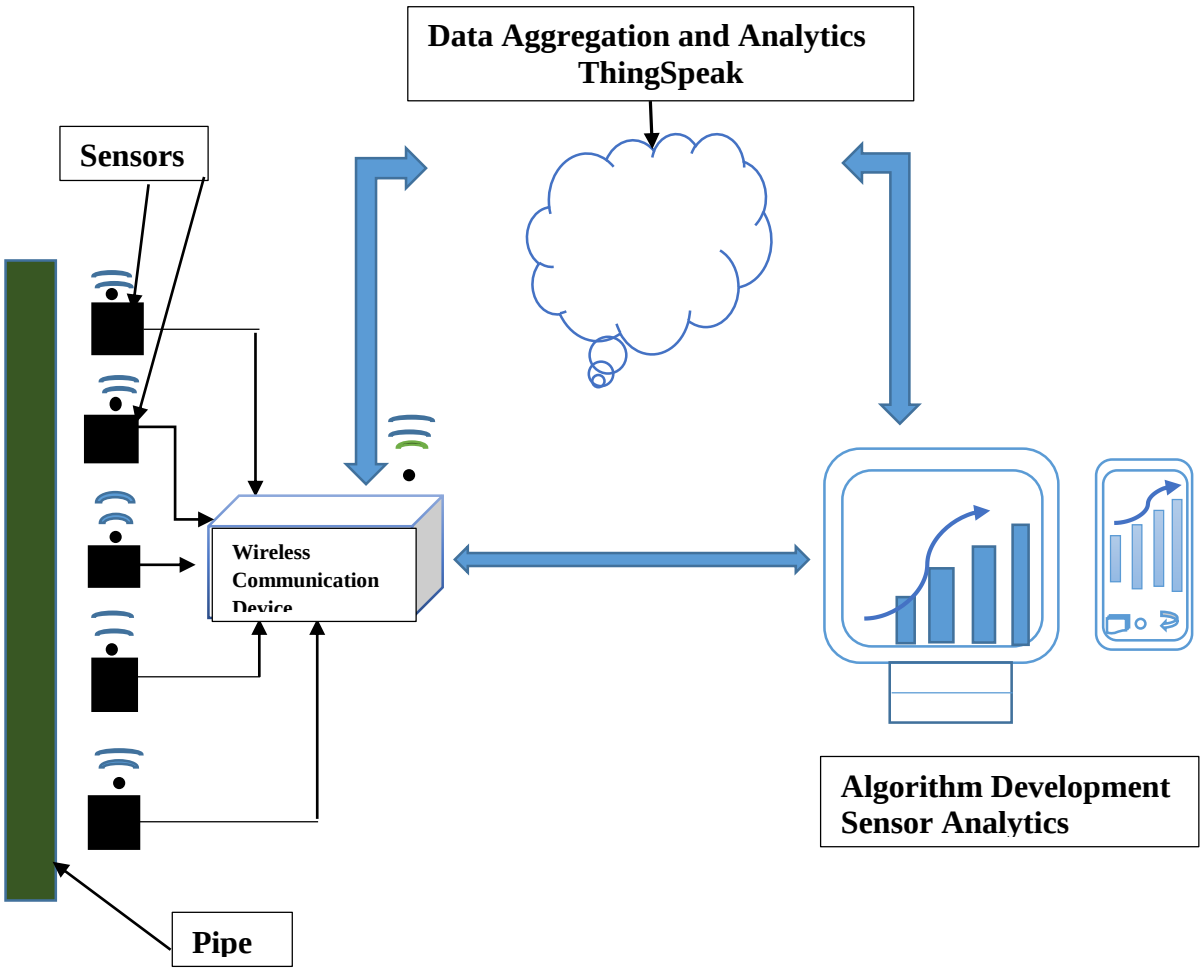

Figure 3.5: IoT system adopted for pipeline monitoring (ThingSpeak, 2019)

Seen on the left of the figure are the smart devices (the "things" in IoT) that live at the edge of the network which is the wireless communication device used in this work. These device collected pressure pulse data from the piezoelectric sensors. In the middle is the cloud where data from the sensors was aggregated and analysed in real time, in this case was by the ThingSpeak IoT analytics platform.

The right side of the diagram depicts the visual display associated with the IoT application. By conducting historical analysis on the data, insight into the collected data was gained.

3.7.1 Optimisation of Developed Detection, Location and Characterisation System

A wireless device was incorporated into the experimental test rig as shown in Plate IV. The Arduino and Wi-Fi module were powered by a computer system through the use of two USB cables. This was so due to proximity to the computer system as they could be powered by being connected to an electric source. The Wi-Fi module of the device was activated by the internet connection from an android phone. This device replaced the data logger that was used in the setup in Plate I. Wire piezoelectric sensors were used in these experiments and the device was connected to the sensors via wires. The sensors captured the pressure pulses from the various damage events and the device transmitted these pulse data to the ThingSpeak analytics platform. Processing of these pulse data was done in real time on the ThingSpeak platform and the output of the measured pressure pulses was also displayed in real time.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

4.1 Simulation of Derived Mathematical Models

Results of the simulation of sampling and reconstruction carried out with this developed model are thus presented in Figures 4.1, 4.2, 4.3, 4.4, 4.5,4.6,4.7,4.8, and 4.9 respectively.

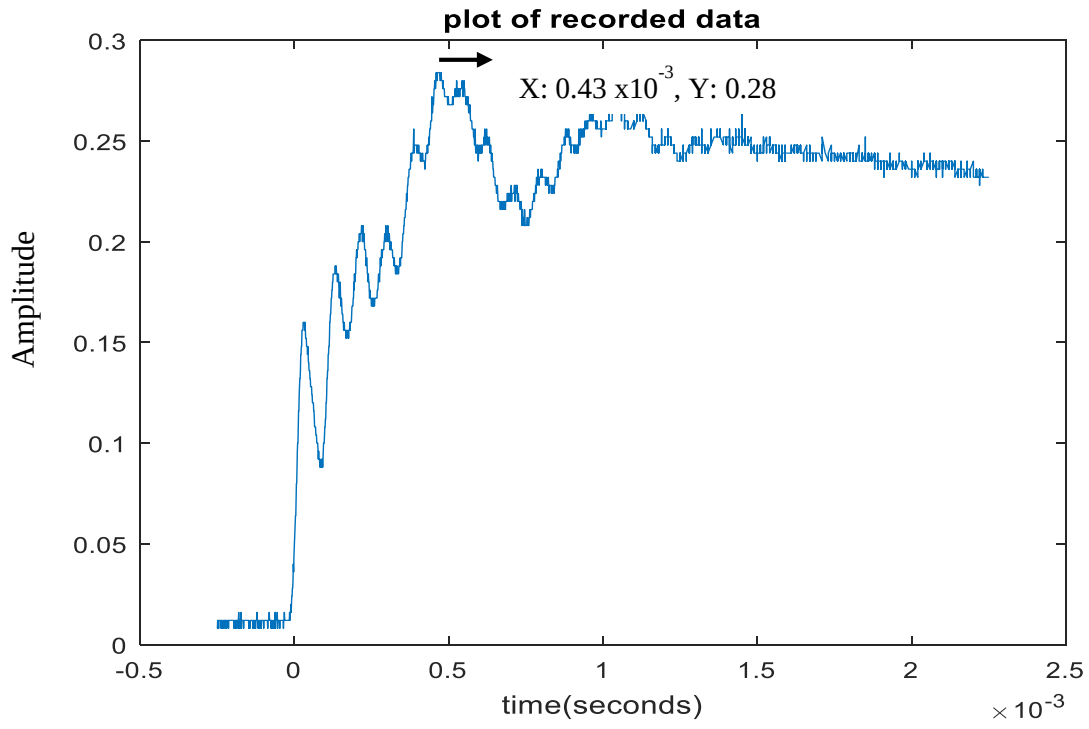

Figure 4.1: Original simulated pulse for data set 1

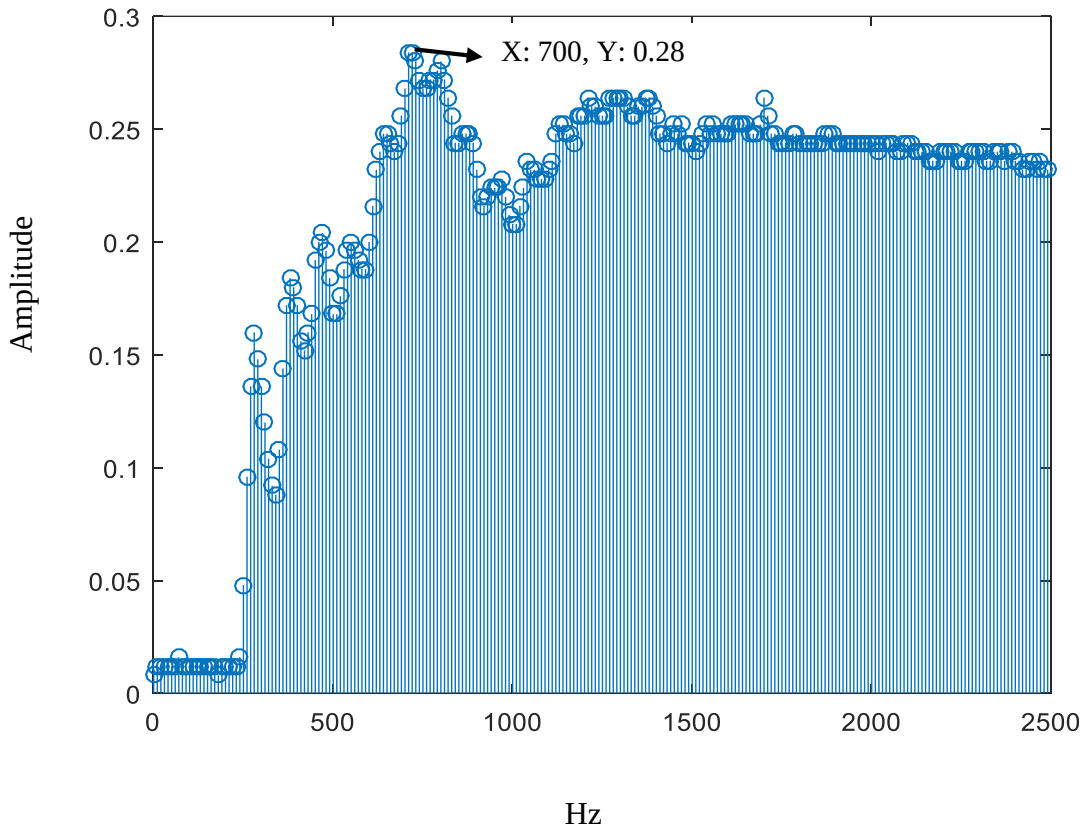

Figure 4.2: Sampled simulated pulse for data set 2

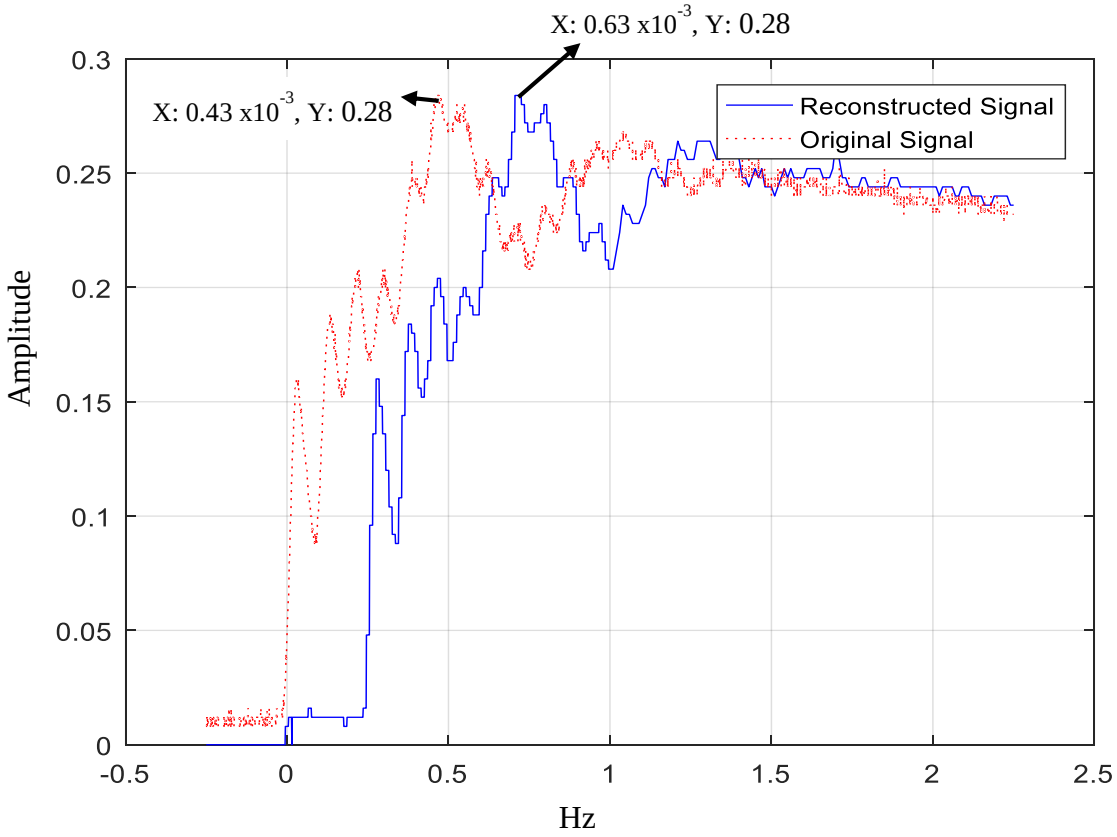

Figure 4.3: Reconstructed and original simulated pulse of data set 1

In Figure 4.1, 4.2, and 4.3, the value of the sampling period was 0.00001 s, and the data contained a single frequency component of 100 KHz. Figure 4.1 shows the plot of the original pulse. Figure 4.2 shows the sampling of the pulse using the Fourier transform method. The sampling was done after every 10 data sets to obtain a total of 250 samples. Figure 4.3 shows the reconstruction of the pulse using the Fourier transform method. The shapes of the reconstructed and measured original pulse in Figure 4.3 agree quite well as both have basically the same contour, and also, the magnitude of both pulse in Figure 4.3 can be seen to be the same with a value of 0.28m. The original measured pulse started at a negative time while the reconstructed pulse started from zero. This is because the reconstruction method does not take into consideration the negative time data values (Blackledge, 2006).

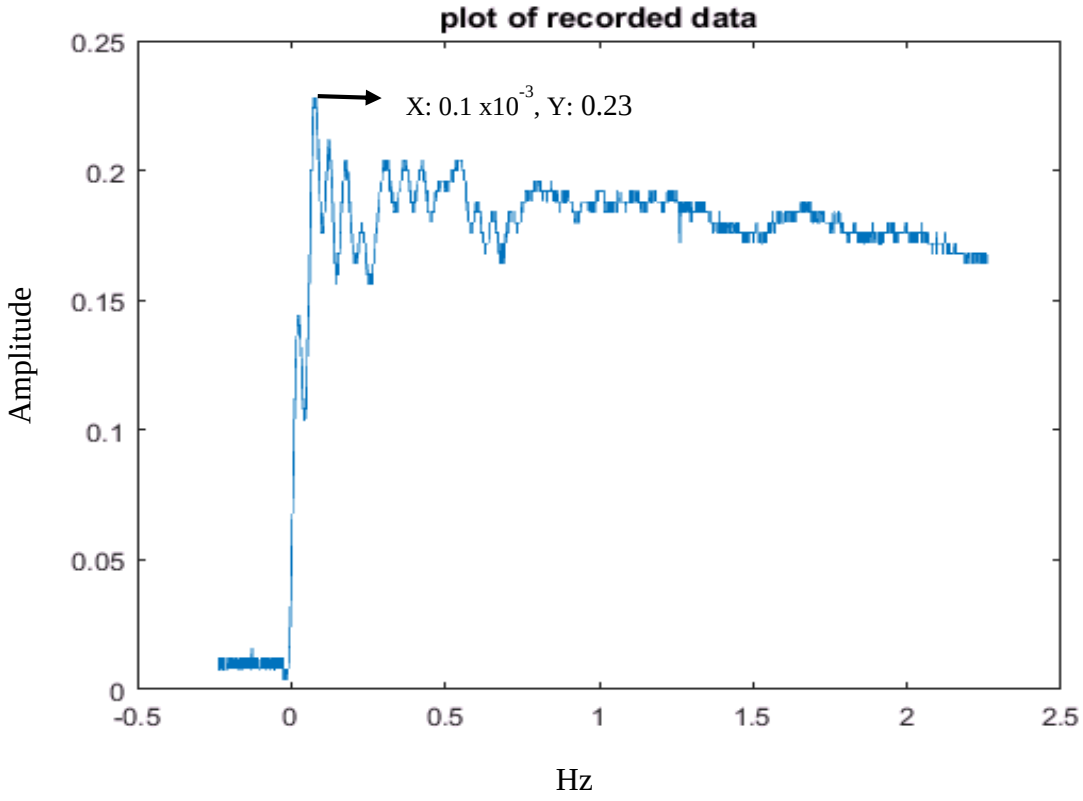

Figure 4.4: Original simulated pulse of data set 2

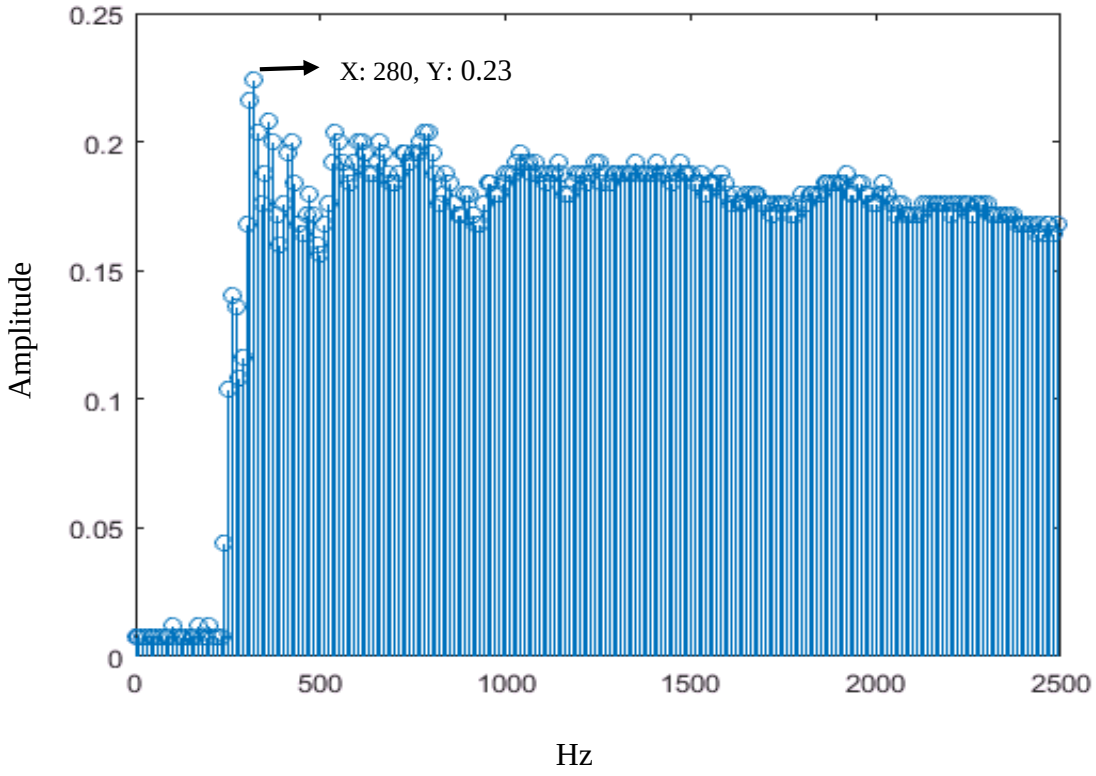

Figure 4.5: Sampled simulated pulse of data set 2

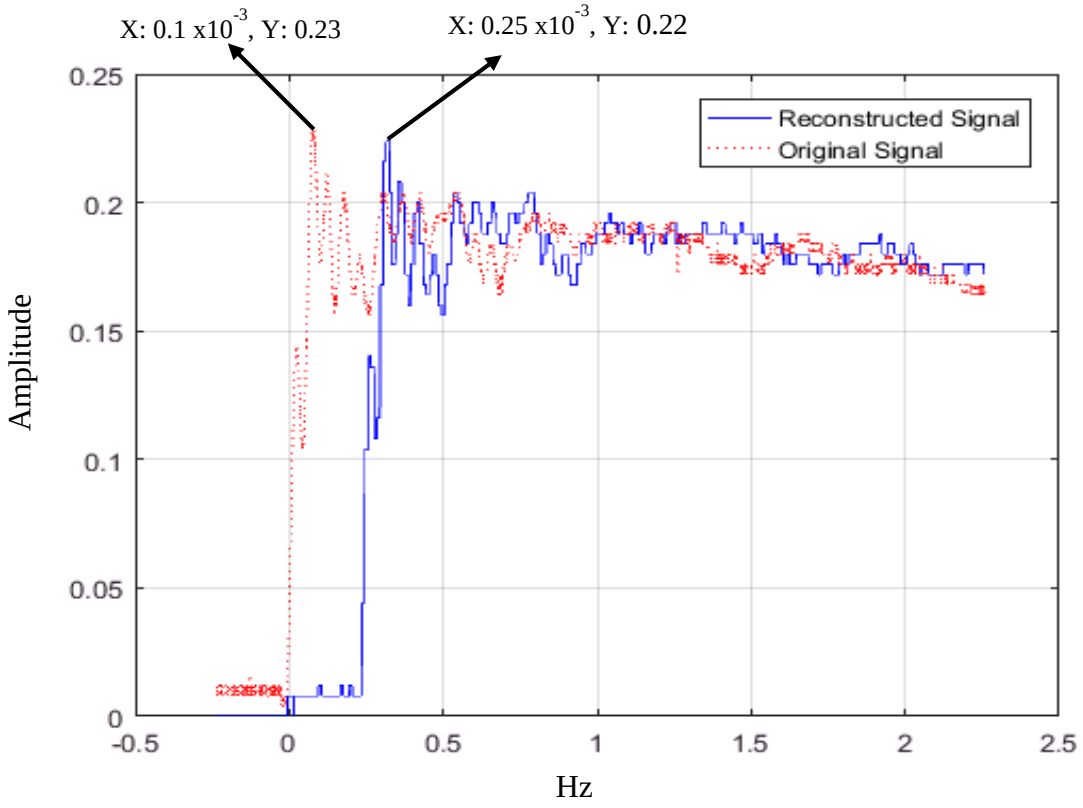

Figure 4.6: Reconstructed and original simulated pulse of data set 2

A sampling period value of 0.00001 s was used in data set 2 and the data contained a single frequency component of 100 KHz. Figure 4.4 shows the plot of the original pulse. Figure 4.5 shows the sampling of the pulse using the Fourier transform method. Figure 4.6 shows the reconstruction of the pulse using the Fourier transform method. The shapes of the reconstructed and measured original pulse in Figure 4.6 agree quite well as both have similar contours. Here, the magnitude of the reconstructed pulse in Figure 4.6 can be seen to be slightly underestimated compared to that of the original measured pulse. The original measured pulse started at a negative time while the reconstructed pulse started from zero. This is because the reconstruction method does not take into consideration the negative time data values (Blackledge, 2006).

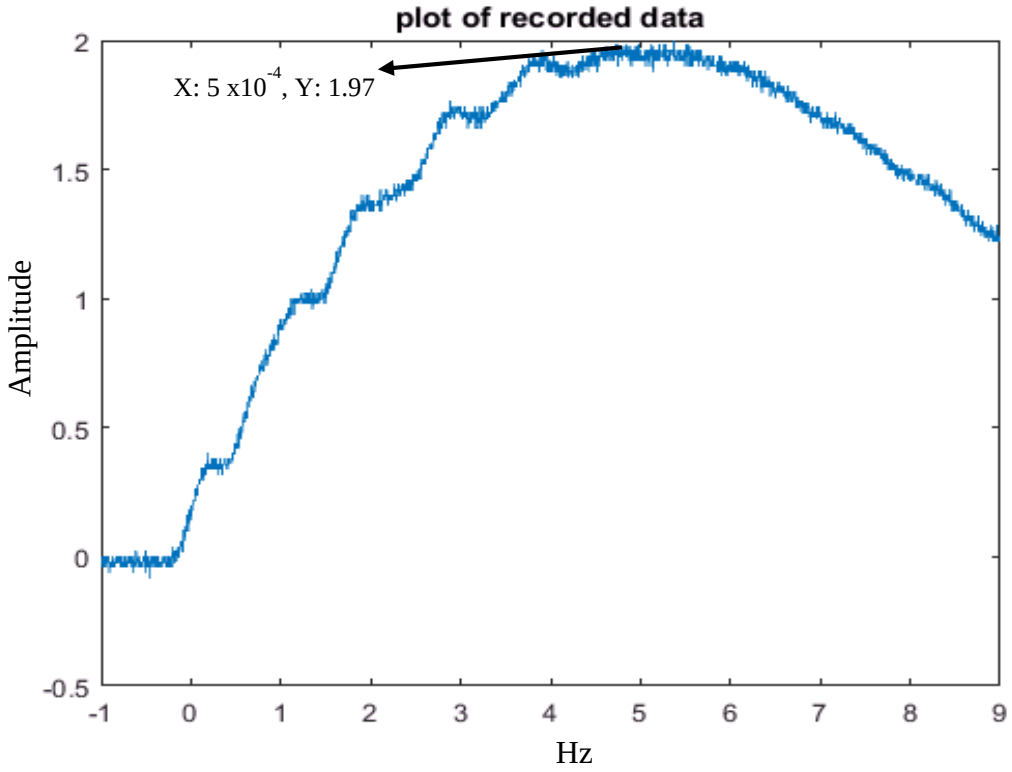

Figure 4.7: Original simulated pulse of data set 3

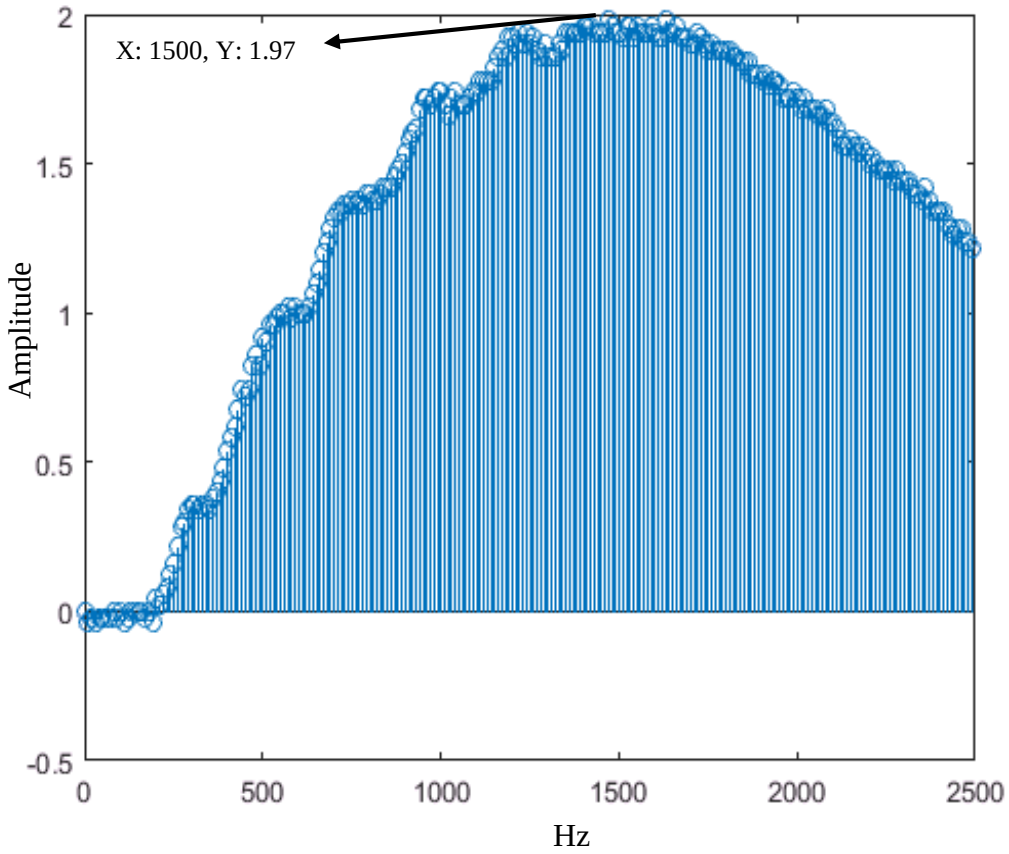

Figure 4.8: Sampled simulated pulse of data set 3

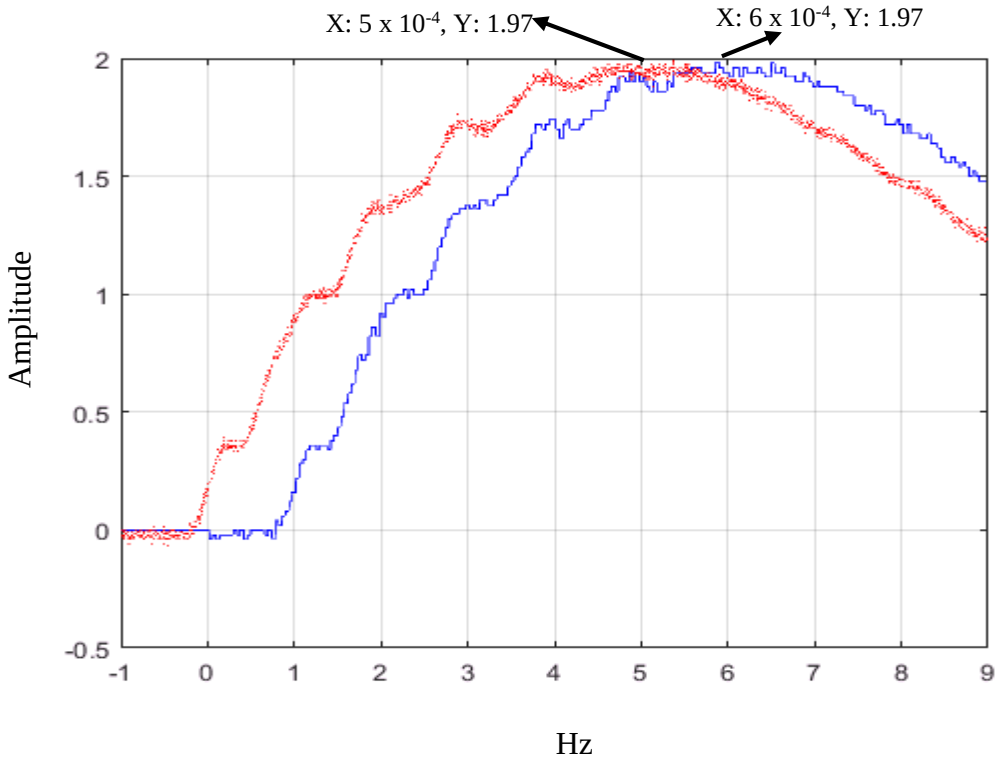

Figure 4.9: Reconstructed and original simulated pulse of data set 3

For data set 3, a sampling period value of 0.000004 s was used and the data contained a single frequency component of 250 KHz. Figure 4.7 shows the plot of the original pulse. Figure 4.8 shows the sampling of the pulse using the Fourier transform method. Figure 4.9 shows the reconstruction of the pulse using the Fourier transform method. The shapes of the reconstructed and measured original pulse in Figure 4.9 also agree quite well as the contours of both pulses are similar. Here, the magnitude of the reconstructed pulse in Figure 4.9 can be seen to be the same as that of the original measured pulse. The original measured pulse here also starts at a negative time while the reconstructed pulse starts from zero. This is because the reconstruction method does not take into consideration the negative time data values (Blackledge, 2006).

From the results of the simulation, it is clear that the developed mathematical model can carry out good event reconstruction.

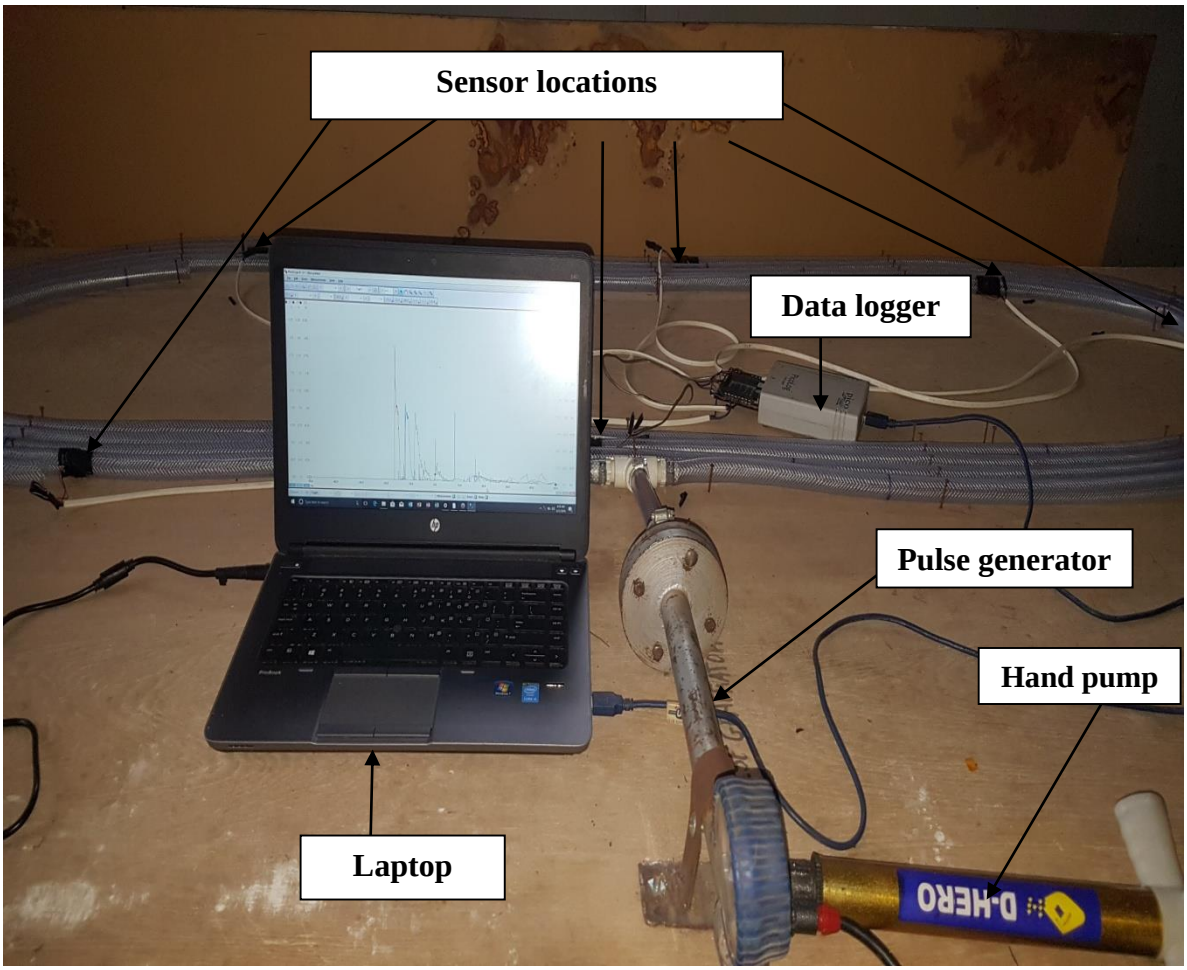

Plate III: Experimental test rig1 showing various components of the rig with air as the transport fluid

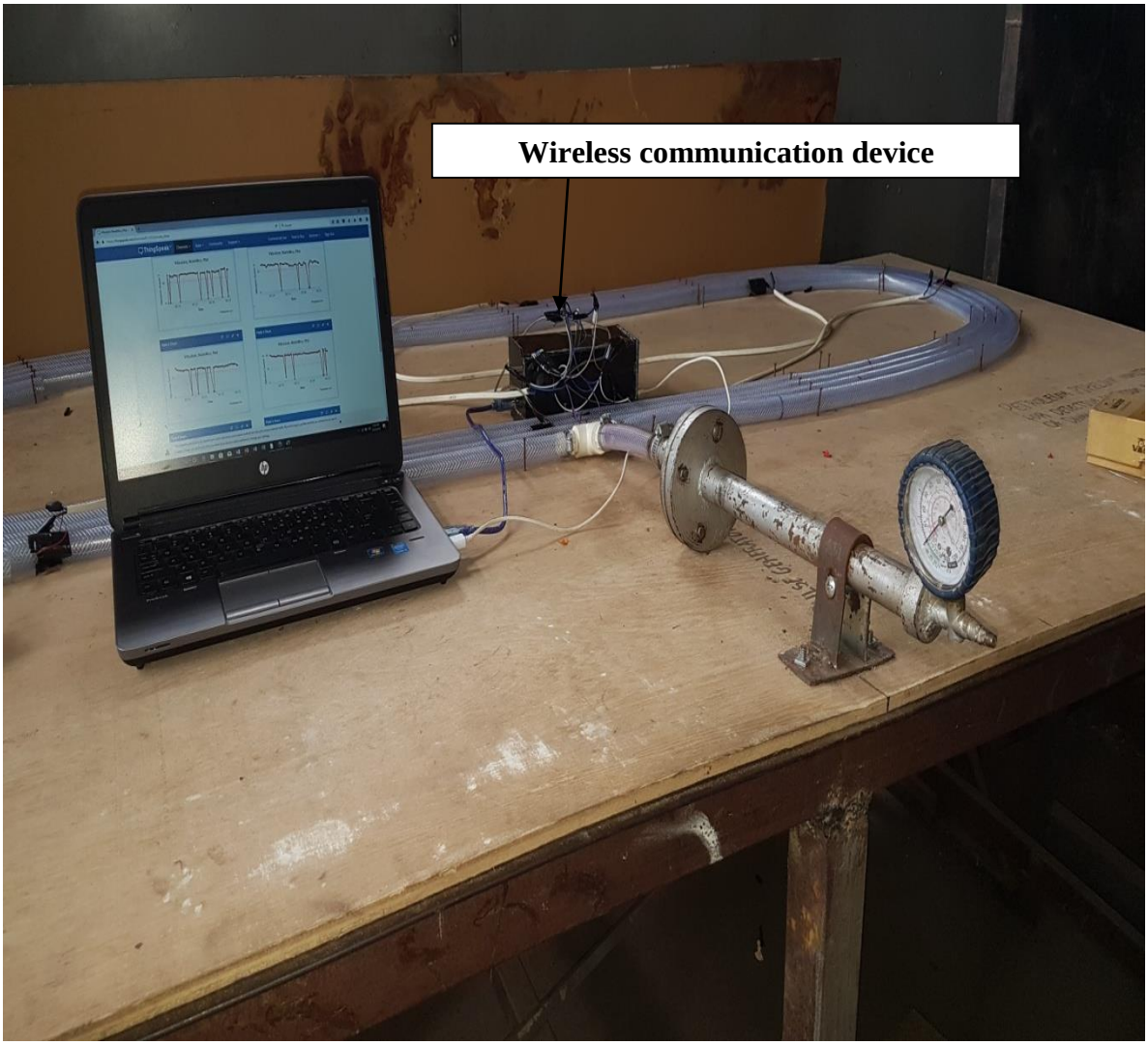

Plate IV: Experimental test rig 2 with wireless communication device and air as the transport fluid

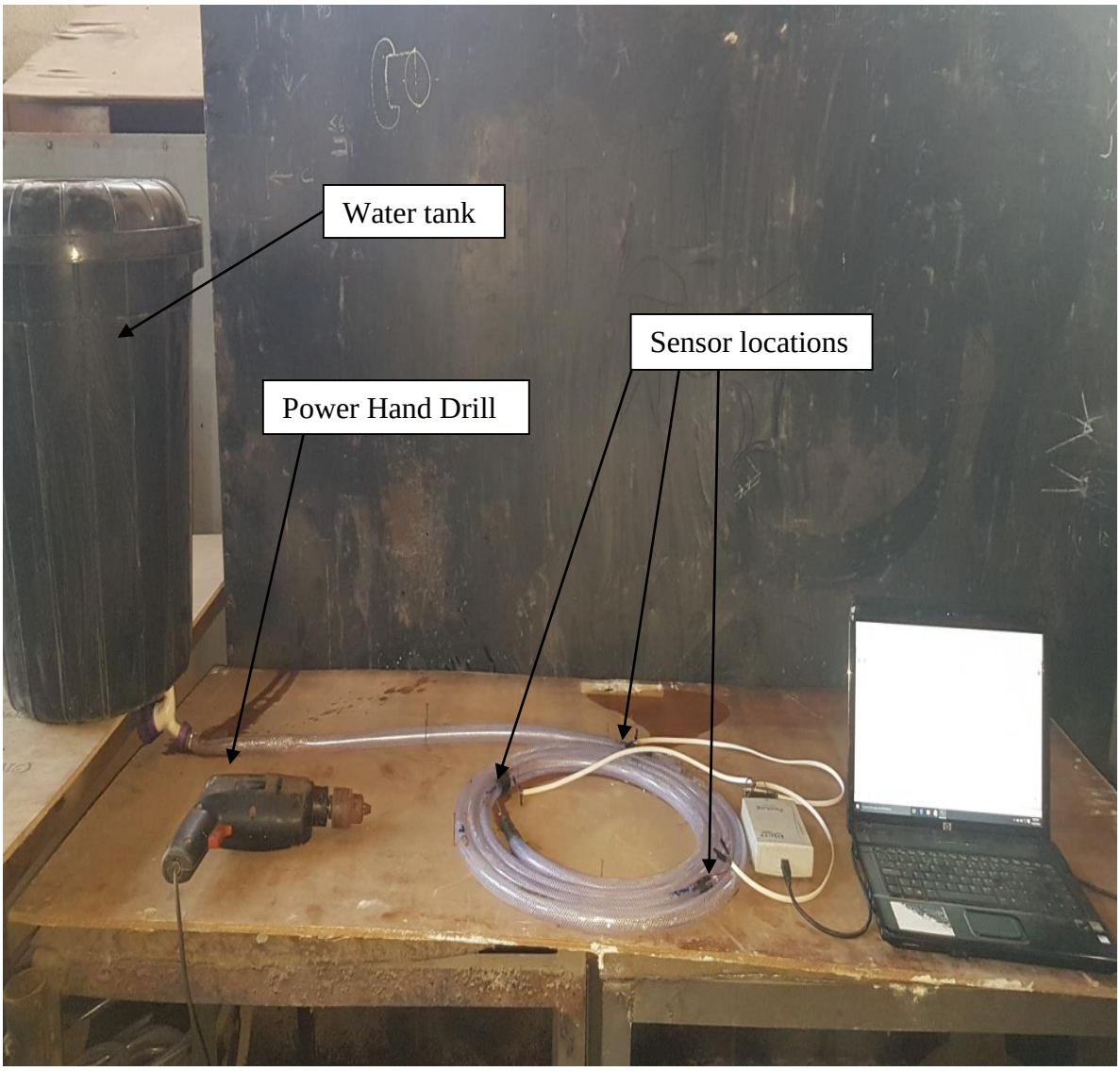

Plate V: Experimental test rig 3 with water as the transport fluid

4.2 Experimental Results

The first set of experiments were carried out using the test rig shown in Plate I with air as the transport fluid. A Pico Log view of the pressure pulses captured along the pipe at all five sensors used is shown Figure 4.10, Figure 4.11 and Figure 4.12 respectively. A sampling rate of 13.16Ks/s was used in measuring and recording the pulse signals at the four sensors. This was twice the value of the frequency of the signals with a value of 6.54Ks/s. Using the sampling rate of 13.16Ks/s, the Niquist criterion was satisfied, and this prevented aliasing of the signals.

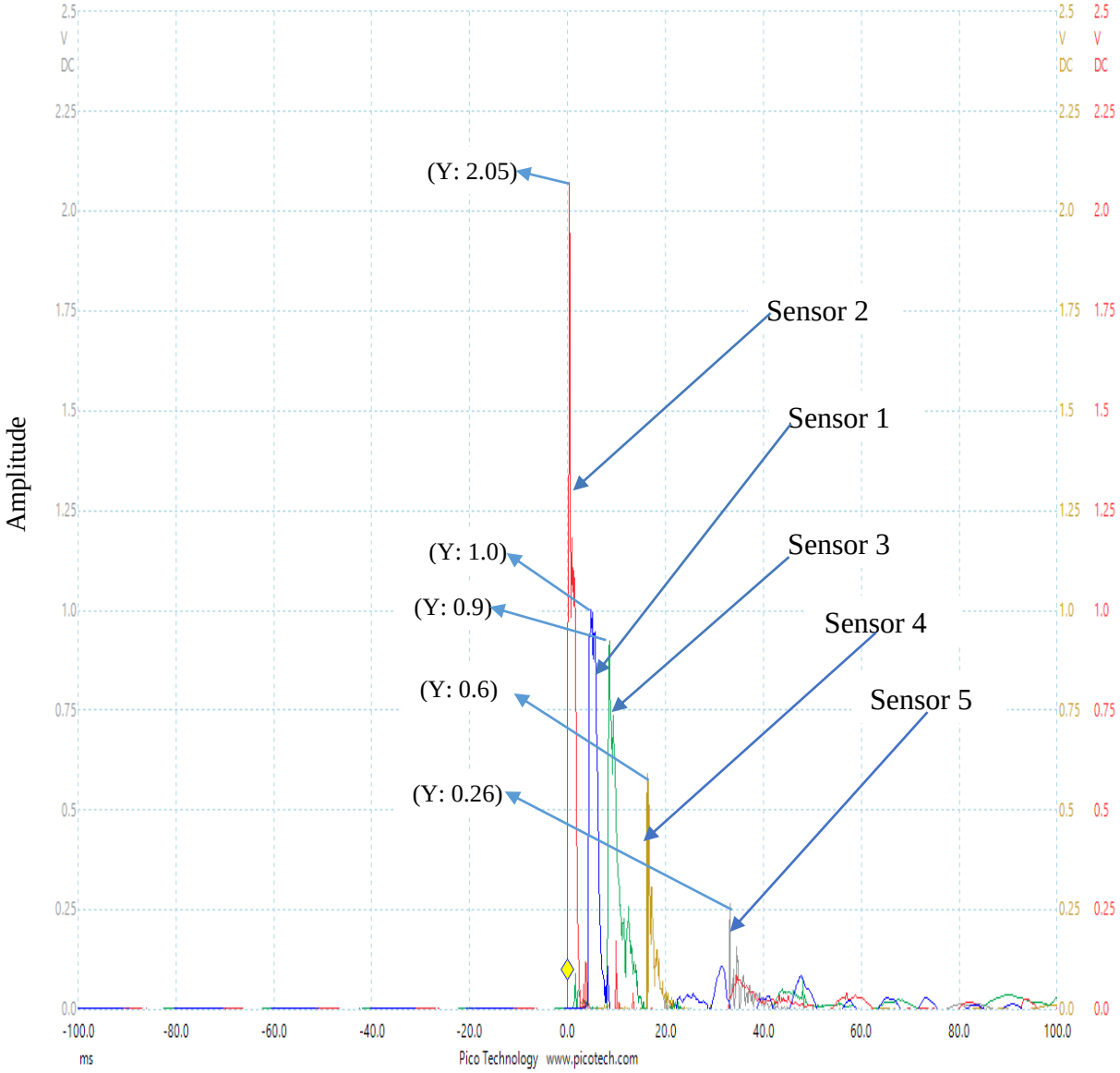

Figure 4.10: Measured pressure pulses from experimental rig sensors at pressure of 1 bar with air as transport fluid

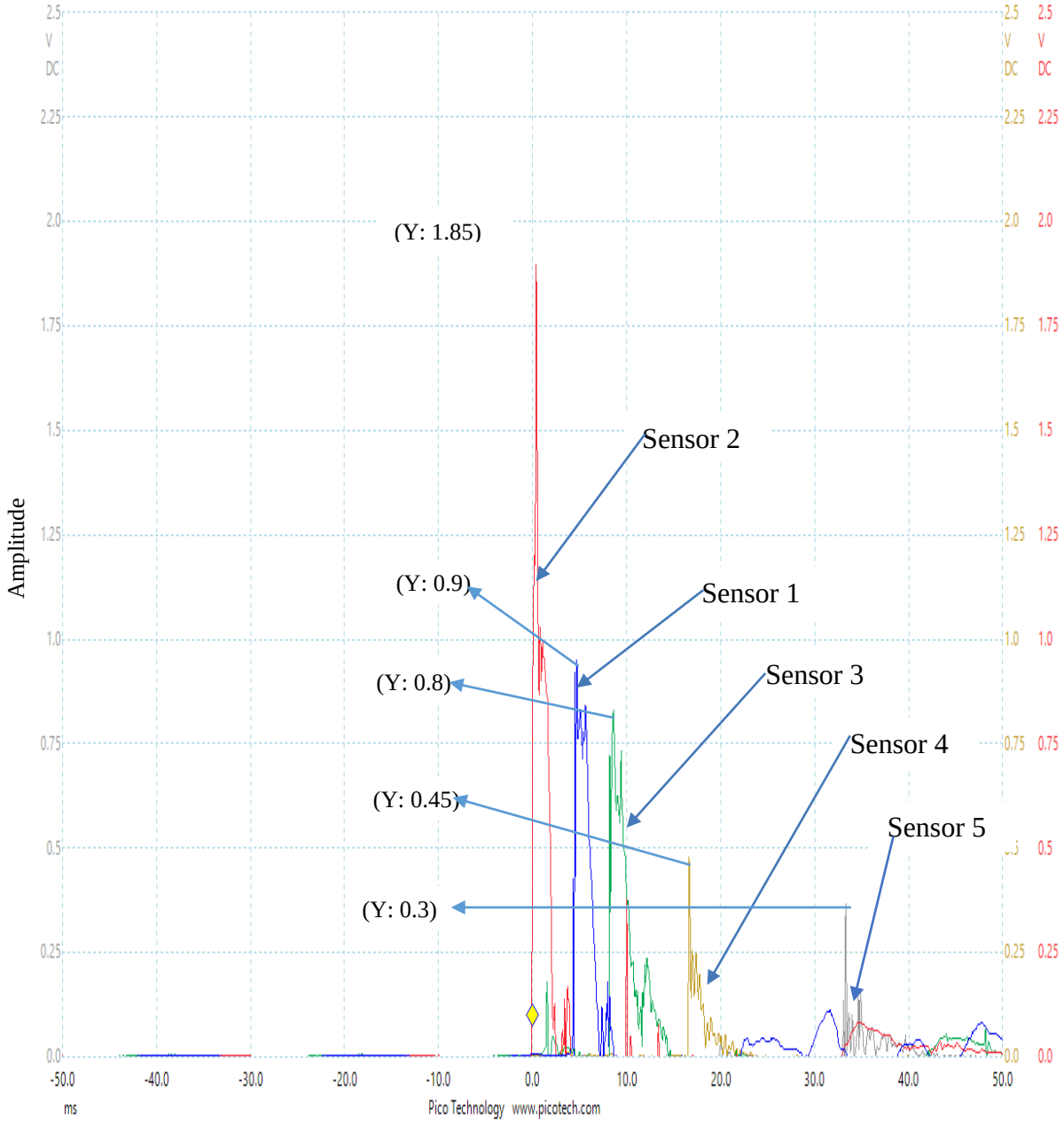

Figure 4.11: Measured pressure pulses from experimental rig sensors at pressure of 0.8 bar with air as transport fluid

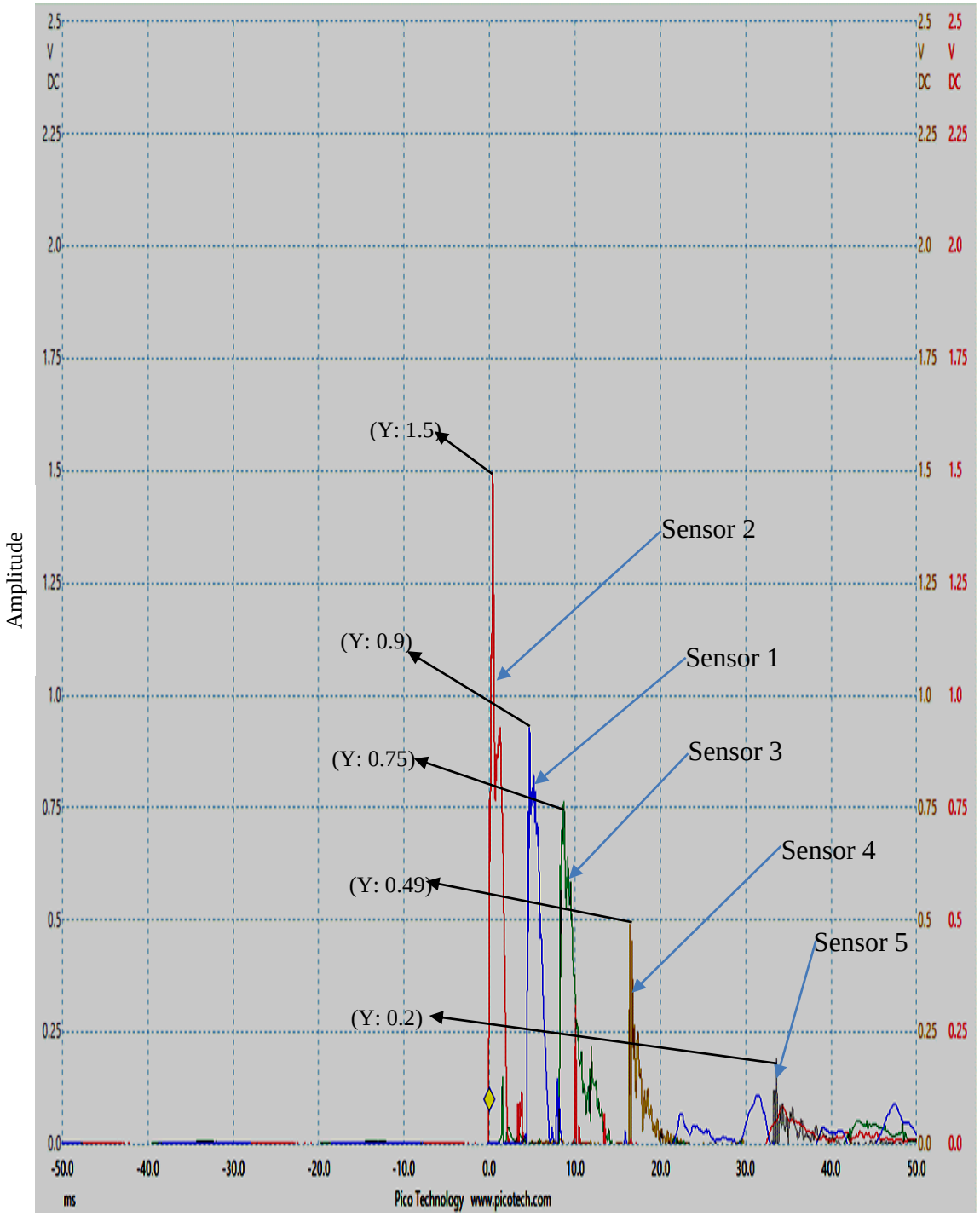

Figure 4.12: Measured pressure pulses from experimental rig sensors at pressure of 0.6 bar with air as transport fluid

Figures 4.10, 4.11 and 4.12 show pressure pulses from three experimental results outputted from all five sensors placed on the test rig at pressure readings of 1bar, 0.8bar, and 0.6 bar respectively with air as the transport fluid. Located closest to the tee connection was sensor 2, meaning it was closest to the point on the pipe where the pulse first arrived entered. This therefore defines the event location. The pulse enters at this point and then propagates in opposite directions, showing up first at sensor 2, then at sensor 1, then at sensor 3, next at sensor 4, and lastly at sensor 5. This is confirmed by the various peaks of the pulses as observed in Figures 4.10, 4.11 and 4.12. In Figure 4.10, the pulse at sensor 2 had a peak value or amplitude of 2.05 m. Also, the peaks at sensors 1, 3, 4 and 5 were 1 m; 0.9 m; 0.6 m and 0.26 m respectively. This shows that the pulse from the pulse generator got to sensor 2 first before getting to sensors 1, 3, 4 and 5. This was the same for all the experiments carried out. The sensors' distance from the point where the produced pulse enters the pipe was directly related to the time it took for these pulses to arrive.

In Figure 4.11, the pulse peak at sensors 2, 1, 3, 4, and 5 were 1.85 m; 0.9 m; 0.8 m; 0.45 m and 0.3 m respectively. In Figure 4.12, the pulse peak at sensors 2, 1, 3, 4, and 5 were 1.5 m; 0.9 m; 0.75 m; 0.49 m and 0.2 m respectively. The pulse peaks in Figures 4. 11 and 4.12 were lower compared to those in Figure 4.10. This was because a pressure of 1 bar was used to obtain the peaks in Figure 4.10, while a pressure of 0.8 bar and 0.6 bar was used to obtain the peaks in Figure 4.11 and 4.12 respectively.

The most ideal autonomous estimation of the occasion as it enters the pipe is sensor 2. The reason being that the pulse generator was closest to sensor 2. This is because attenuation or distortion of the pulse propagating from the tee connection will be very little before it reaches sensor 2 as a result of its proximity to the tee connection. The remaining four sensors are located at different points along the pipe to aid the location of the event. It is observed from

Figures 4.10, 4.11 and 4.12 that a reflection of the pressure pulse generated originally back into the pulse generator occurs. This reflection is observed to go back and forth from the pulse into the pipe. In carrying out experimental calculations, the negative pulse was not utilised. This is because the Fourier transform model does not take into consideration negative pulse data values.

The experiments were also carried out using the test rig shown in Plate III with water as the transport fluid. A Pico Log view of the pressure pulses captured along the pipe at all five sensors used is shown Figure 4.13, Figure 4.14 and Figure 4.15. A sampling rate of 13.16Ks/s was also used in measuring and recording the pulse signals at the four sensors.

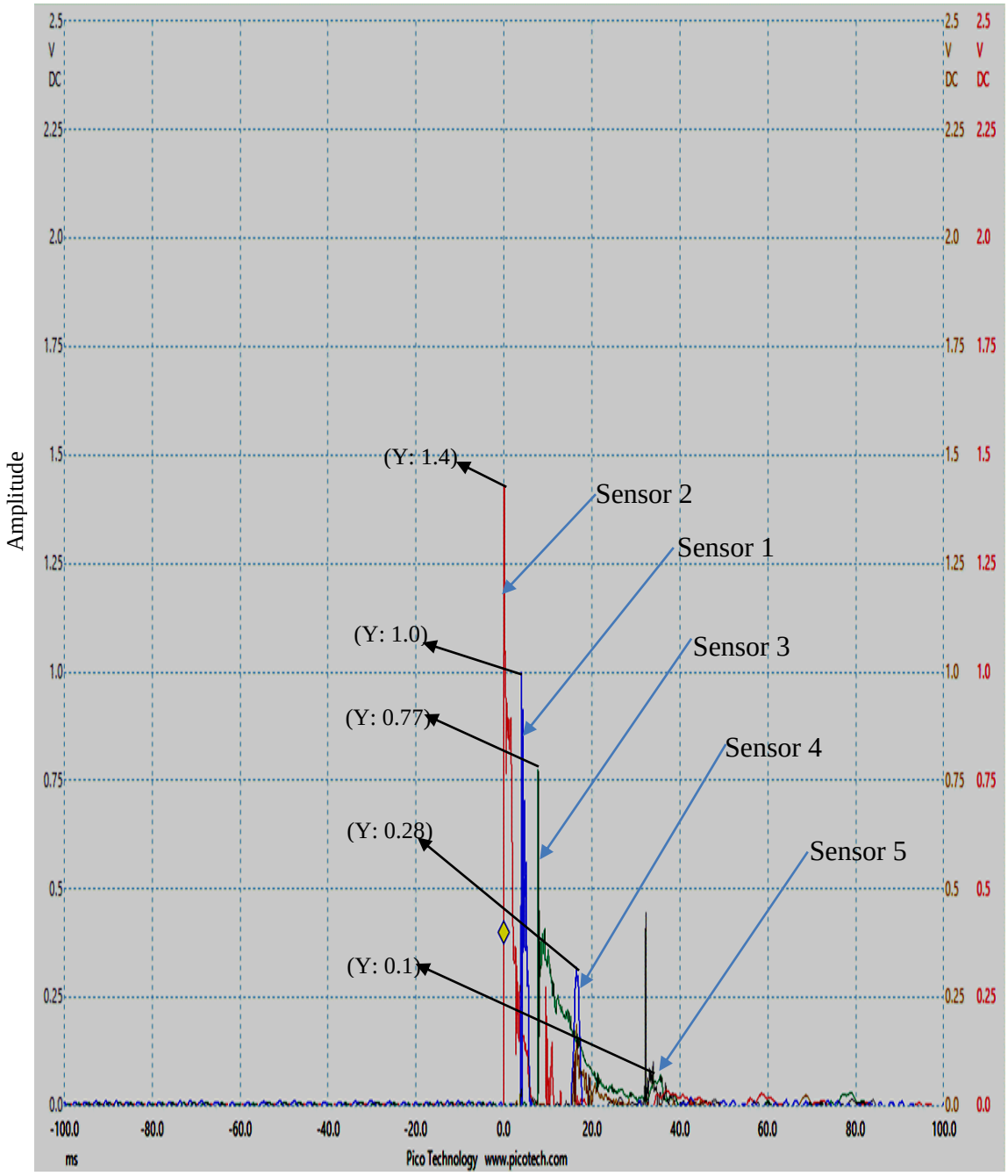

Figure 4.13: Measured pressure pulses from experimental rig sensors at pressure of 1 bar with water as transport fluid

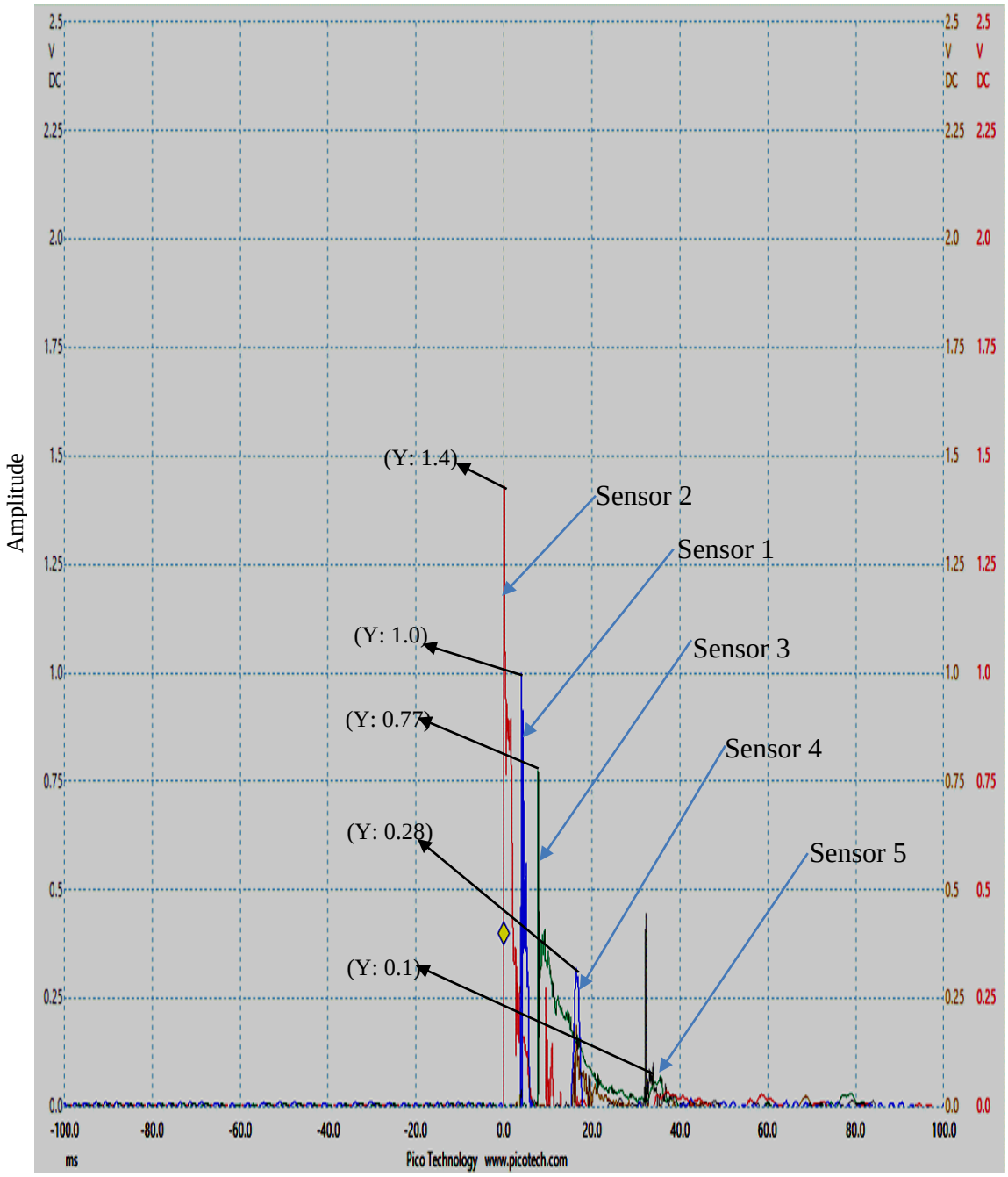

Figure 4.14: Measured pressure pulses from experimental rig sensors at pressure of 0.8 bar with water as transport fluid

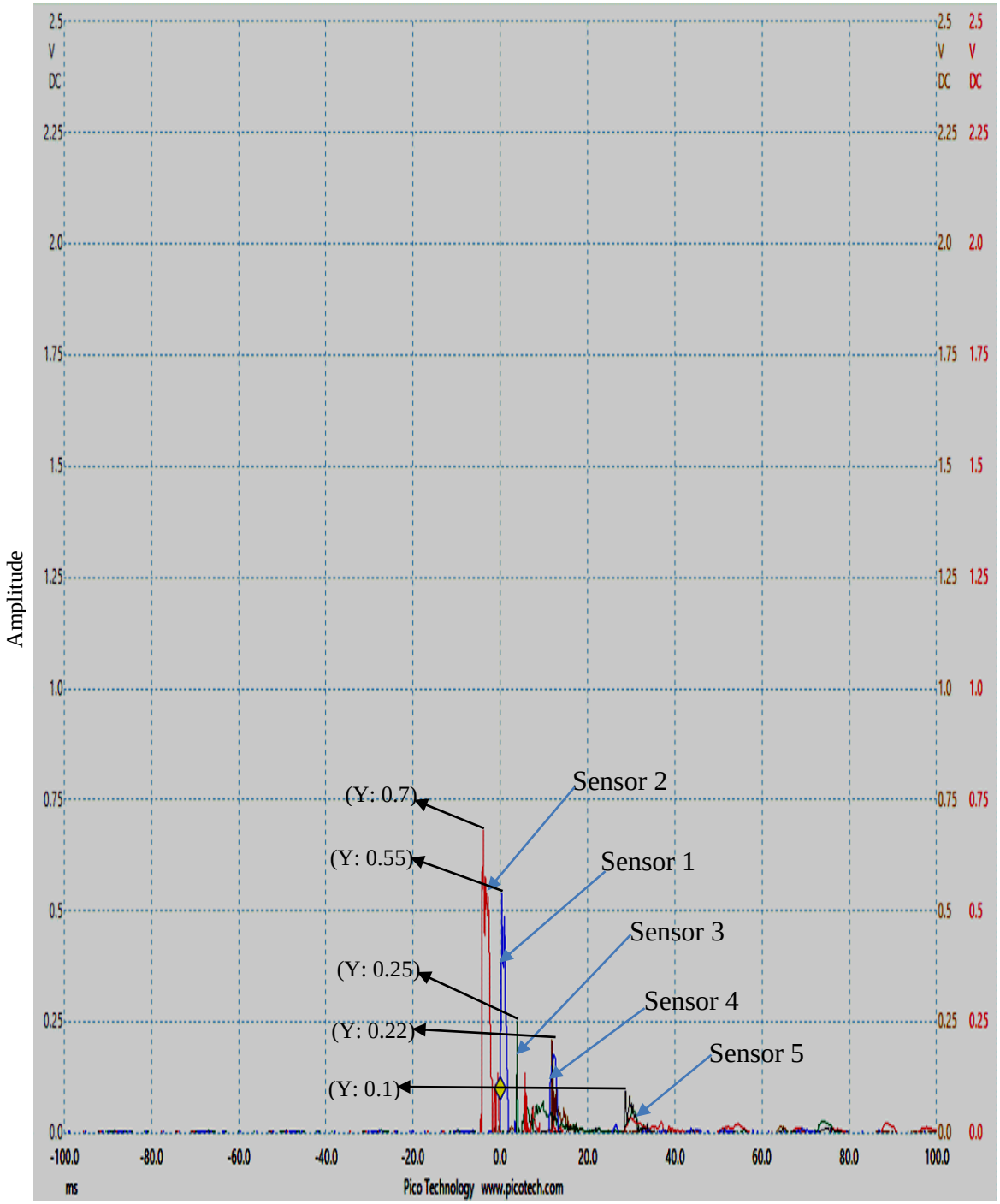

Figure 4.15: Measured pressure pulses from experimental rig sensors at pressure of 0.6 bar with water as transport fluid

Figures 4.13, 4.14 and 4.15 show pressure pulses from three experimental results outputted from all five sensors placed on the test rig at pressure readings of 1bar, 0.8bar, and 0.6 bar respectively with water as the transport. Located closest to the tee connection was sensor 2, meaning it was closest to the point on the pipe where the pulse first arrived entered. Also, this therefore defines the event location. The pulse enters at this point and then propagates in opposite directions, showing up first at sensor 2, then at sensor 1, then at sensor 3, next at sensor 4, and lastly at sensor 5. Here also, the time of arrival of these pulses were directly related to the respective distance of the sensors from the point at which the generated pulse enters the pipe.

In Figure 4.13, the pulse peak at sensors 2, 1, 3, 4, and 5 were 1.4 m; 1.0 m; 0.77 m; 0.28 m and 0.1 m respectively. In Figure 4.14, the pulse peak at sensors 2, 1, 3, 4, and 5 were 1.4 m; 1.0 m; 0.77 m; 0.28 m and 0.1 m respectively. In Figure 4.15, the pulse peak at sensors 2, 1, 3, 4, and 5 were 0.7 m; 0.55 m; 0.22 m; 0.25 m and 0.1 m respectively. The pulse peaks in Figures 4.14 and 4.15 were lower compared to those in Figure 4.13. This was because a pressure of 1 bar was used to obtain the peaks in Figure 4.13, while a pressure of 0.8 bar and 0.6 bar was used to obtain the peaks in Figure 4.14 and 4.15 respectively.

The most ideal autonomous estimation of the occasion as it enters the pipe is also at sensor 2. This is because attenuation or distortion of the pulse propagating from the tee connection will be very little before it reaches sensor 2 as a result of its proximity to the tee connection. The remaining four sensors are located at different points along the pipe to aid the location of the event. It is observed from Figures 4.13, 4.14 and 4.15 that a reflection of the pressure pulse generated originally back into the pulse generator occurs. This reflection is observed to go back and forth from the pulse into the pipe. In carrying out experimental calculations,

the negative pulse was also not utilised. This is because the Fourier transform model does not take into consideration negative pulse data values.

4.3 Velocity of Pressure Pulse Propagation in Static Air

Pulse data obtained from sensors 4 and 3 were used adopted for determining the velocity of the pulse propagation across the pipe based on the configuration in Plate I, with air as the transport fluid. The MATLAB® software was used to cross-correlate these measured pulses to determine arrival time delay between the pulses. The pressure pulse propagation velocity was determined using Equation 3.4 (Olugboji, 2013):

$$C_p = \frac{x_{34}}{\text{tdelay}_{34}} \qquad 4.1$$

Where, x_{34} = distance between sensors 4 and 3.

x_{34} = 3m

tdelay_{34} = 0.008297s

$$C_{p\ air} = \frac{3}{0.00845}$$

$C_{p\ air}$ = 355m/s

The velocity of the measured pressure pulse propagation was computed to be 355m/s. There was a clear difference between this value and 343 m/s, which is the velocity of sound in air (NASA, 2018). Olugboji, (2011) investigated the reason for the discrepancy between the value of calculated pressure pulse propagation velocity and that of nominal velocity of sound propagation in his work since it was systematic and repeatable. It was suggested in his work that a localised temperature rise might have led to the pressure increase within the pipe. The

necessary rise in temperature was calculated to investigate the hypothesis. From his calculations, a temperature rise of 10^{o}C and a pressure rise of 34 mbar above ambient was obtained. This led to his work confirming the high measured pulse velocities were as a result of the temperature change within in the pulses. To confirm this, calculations were carried out to see if they were a rise in the temperature of the measured pulses as thus (Olugboji, 2013):

$$\left(\frac{Cn}{Cp}\right)^2 = \frac{Tn}{Tp} \qquad 4.2$$

$$Tp = Tn\left(\frac{Cp}{Cn}\right)^2 \qquad 4.3$$

Where Tp = air temperature within the pressure pulse

Tn = ambient temperature

Cn = nominal sound propagation velocity

Cp = measured pulse propagation velocity

With a measured pulse propagation velocity of 355 m/s, and a sound propagation velocity of 343 m/s, at 25.3^{o}C, Tp was calculated using Equation 4.3 as:

$$Tp = 298.3 * \left(\frac{355}{343}\right)^2 = 319.5K,$$

This represents a temperature rise of 21.2^{o}C. This value goes to confirm the proposal made by Olugboji, (2011) and thus explains the discrepancy in the values of the measured pressure pulse velocity and the nominal sound velocity in air.

4.4 Velocity of Pressure Pulse Propagation in Flowing Water

Pulse data obtained from sensors 4 and 3 were also used to determine the pulse propagation velocity across the pipe based on the configuration in Plate III, with water as the transport fluid. The MATLAB® software was used to cross-correlate these measured pulses to determine arrival time delay between the pulses. The pressure pulse propagation velocity was determined using Equation 4.1. The pressure pulse with the highest amplitude was obtained using a pressure reading of 1.0 bar in the pulse generator, while the pressure pulse with the least amplitude was obtained using a pressure reading of 0.2 bar. Only 10 measurements were made in total because 2 measurements each were made for each pressure rating in the pulse generator. This was due to the sensors losing their sensitivity a short while after coming in contact with water. An average of the time delay between pulse arrivals was obtained after cross correlation in MATLAB® and the result of the average value used in Equation 4.1 to obtain the velocity of pulse propagation as:

$x_{34} = 3m$

$t_{delay34} = 0.00195s$

$C_{p\ water} = \frac{3}{0.00195}$

$C_{p\ water} = 1{,}538 m/s$

The velocity of the measured pressure pulse propagation was computed to be 1,538 m/s. There was a clear difference between this value and the value of the velocity of sound in water at a temperature of 26°C and normal pressure which is 1500 m/s (DOSITS, 2019). The reason for the discrepancy between the value of calculated pressure pulse propagation velocity and that of nominal velocity of sound propagation in water was also investigated to

infer if the rise in velocity was caused by the pressure and localised temperature rise within the pulse. The necessary rise in temperature if any, was calculated to investigate the hypothesis using Equation 4.2 and 4.3.

With a measured pulse propagation velocity of 1538 m/s, and a sound propagation velocity of 1500 m/s, at 25.3ºC, Tp was calculated using Equation 4.3 as:

$$\mathrm{Tp} = 298.3 * \left(\frac{1538}{1500}\right)^2 = 313.6\mathrm{K},$$

This represents a temperature rise of 15.3ºC. This thus explains the discrepancy in the values of the measured pressure pulse velocity and the nominal sound velocity in water.

4.5 Event Location in Static Air

The m-code language in MATLAB® was used in the calculation of the event location of the test rig's pipe with air as the transport fluid. Figure 4.16 shows an estimate of the location of the real event. It shows a spread of the location calculations against the values of the pressure in the pulse generator for a total of fifty tests that were carried out. These number of tests were carried out to confirm the accuracy of the experiments that were carried out.

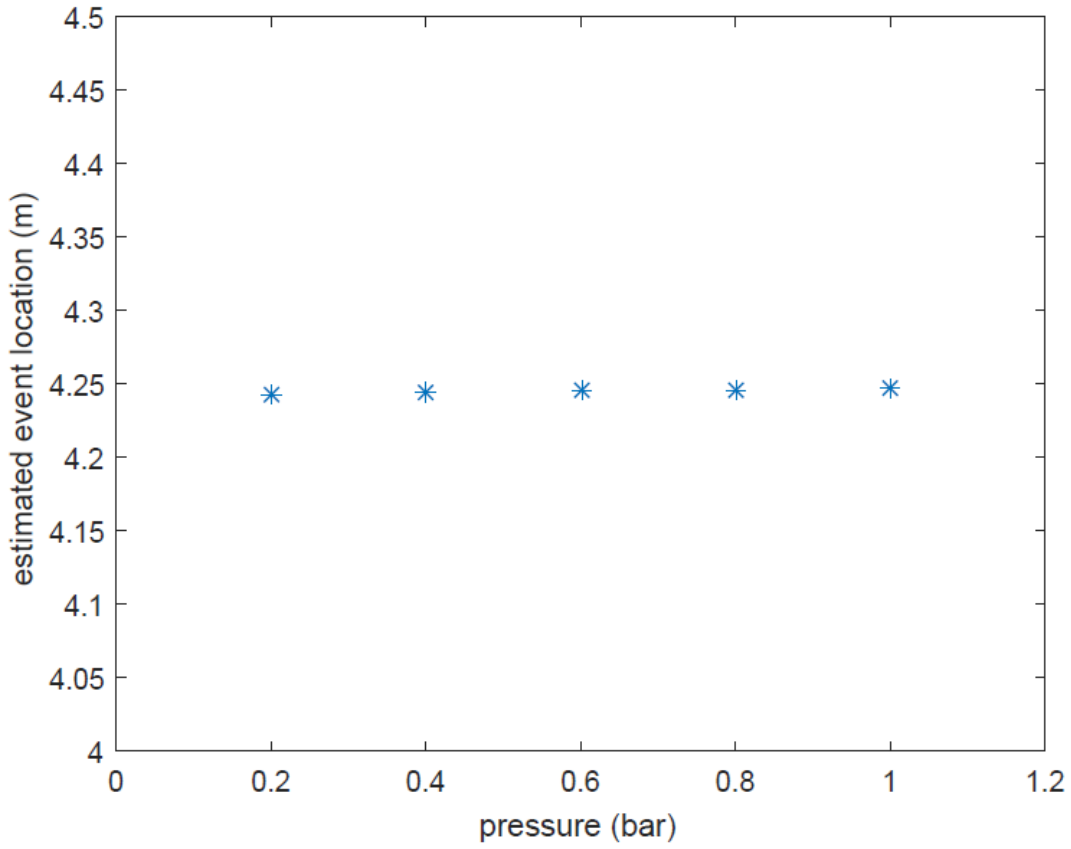

Figure 4.16: MATLAB® presentation of computed event location estimates using air filled pipe

The location of the real event is the point of entrance of the original pressure pulse into the main pipe and it was determined from measurements made at sensors 1 and 2. The tee connection is the actual source of the pulses and formed the basis of all calculations about the location of the event. Equation 3.2 was used to calculate the actual event location which is the event occurring at sensor 2. The equation was slightly modified because as opposed to Figure 3.1, in the actual experimental test rig, the tee connection was located between sensors 1 and 2, but closest to sensor 2 (the event location). Therefore Equation 3.2 was modified as (Olugboji, 2011):

$$x_{DE1} = \frac{(t12Cp + x21)}{2} \quad 4.4$$

Equation 4.4 (the distance of the event location from sensor 1) plus the distance between the tee connection (x_{offset}) and sensor 2 gave the location of the event. x_{21} is the known distance between sensor 1 and 2, while t_{12} is the measured time delay between pulse arrivals at sensors 1 and 2 which was obtained using the cross-correlation technique. With the aid of the written MATLAB® code, a consistency in the computed event location estimates was observed as show in Figure 4.16 confirming the accuracy of the experimental setup to validate the simulation carried out earlier. The computed estimates were in the range of between 4.243 m ad 4.246 m, a scatter of just 3mm, while the calculated measured location of the event was 4.226 m and the measured location of the event on the test rig was 4.23 m.

4.6 Event Location in Flowing Water

The MATLAB® m-code language was then used in the calculation of the event location of the test rig's pipe with water as the transport fluid. Figure 4.17 shows an estimate of the location of the real event. It shows a spread of the location calculations against the values of the pressure in the pulse generator for all the ten tests that were performed. Only ten were carried out as compared to the 50 carried out when air was the transport fluid because the sensors gradually lost their sensitivity when they came in contact with water.

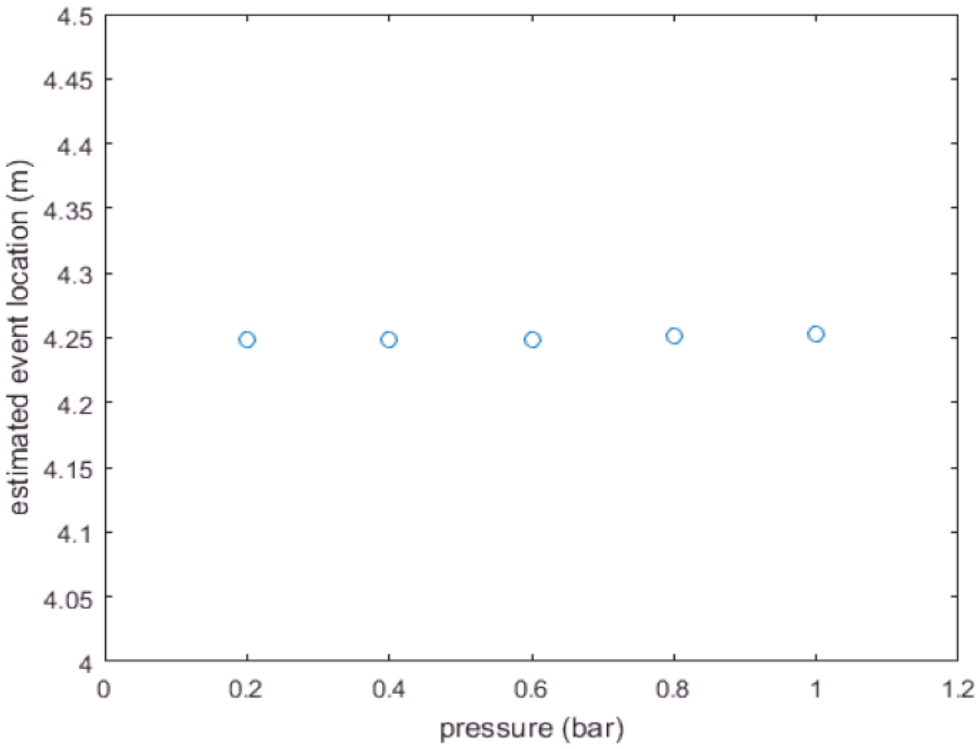

Figure 4.17: MATLAB® presentation of computed event location estimates using water filled pipe

Equation 4.4 was also used to calculate the actual event location which is the event occurring at sensor 2. With the aid of the written MATLAB® code, a consistency in the computed event location estimates was observed as show in Figure 4.17. The computed estimates had a scatter of just 5 mm, and were in the range of between 4.248 m ad 4.253 m, while the calculated location of the event was 4.3 m and the measured location of the event on the test rig was 4.23 m.

4.7 Characterisation of Damage

4.7.1 Event Reconstruction

The mathematical models for sampling in Equation 3.13 and the mathematical model for reconstruction in Equation 3.18 were applied through codes written in MATLAB® to the data

obtained from one of the many experiments made using the pulse generator and static air in the pipe as shown in Figure 4.18. The codes can be seen in Appendix A. The five sensors were connected to five different channels on the data logger and the original pressure pulses captured at all sensors were labelled as s_1, s_2, s_3, s_4, and s_5 respectively, all located at various distances as discussed earlier. The tee connection was located between sensors 1 (s_1) and 2 (s_2). Sensor 2 (s_2) was closest to the tee connection and was taken to be the actual event location. The set up in Plate I was used for these experiments.

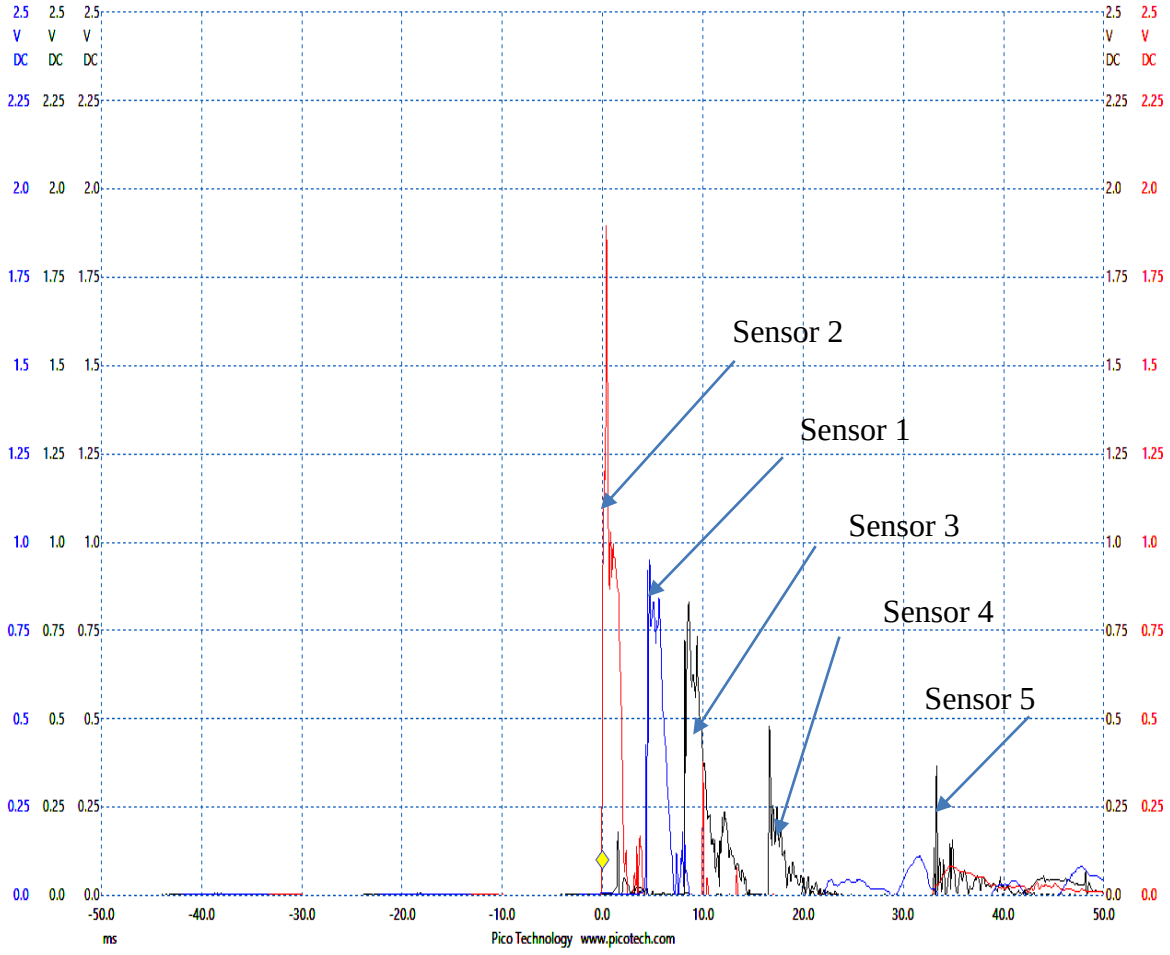

Figure 4.18: Measured pressure pulse at all five sensors

The MATLAB® plots of the pressure pulse at sensors s_1 to s_5 are shown in Figure 4.19, Figure 4.20, Figure 4.21, Figure 4.22, and Figure 4.23 respectively.

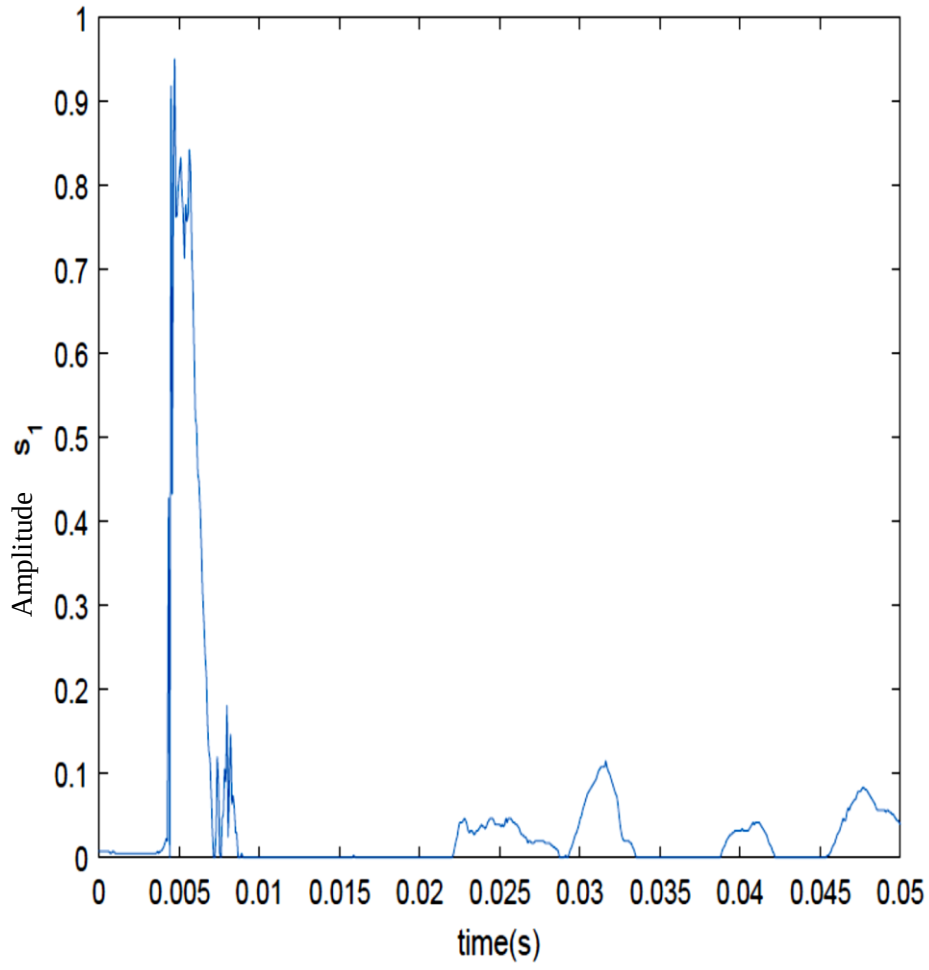

Figure 4.19: MATLAB® plot of pressure pulse at sensor 1

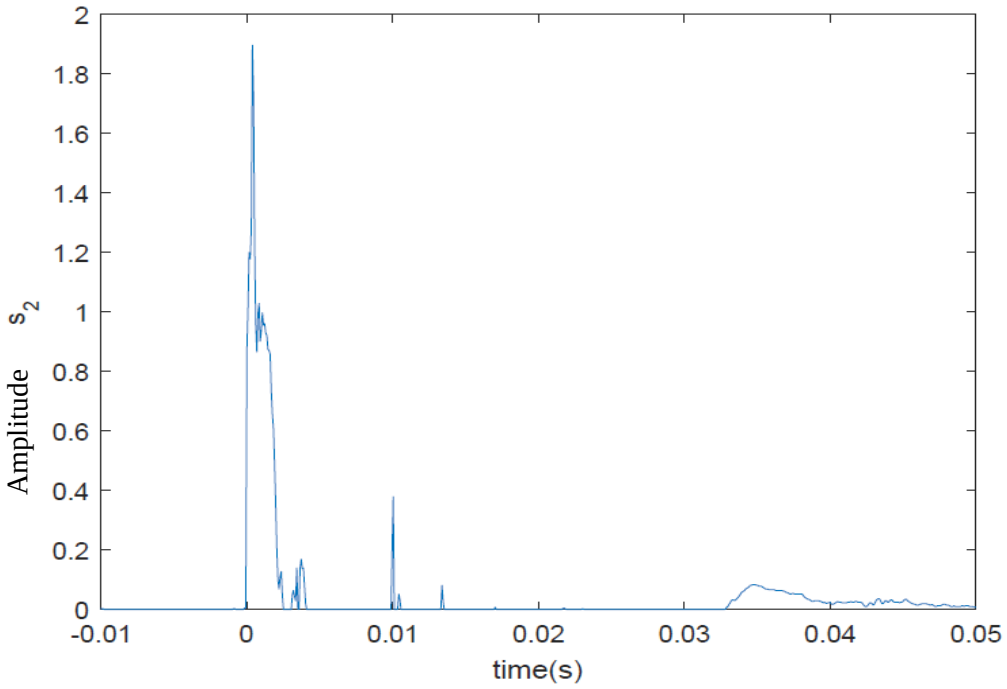

Figure 4.20: MATLAB® plot of pressure pulse at sensor 2

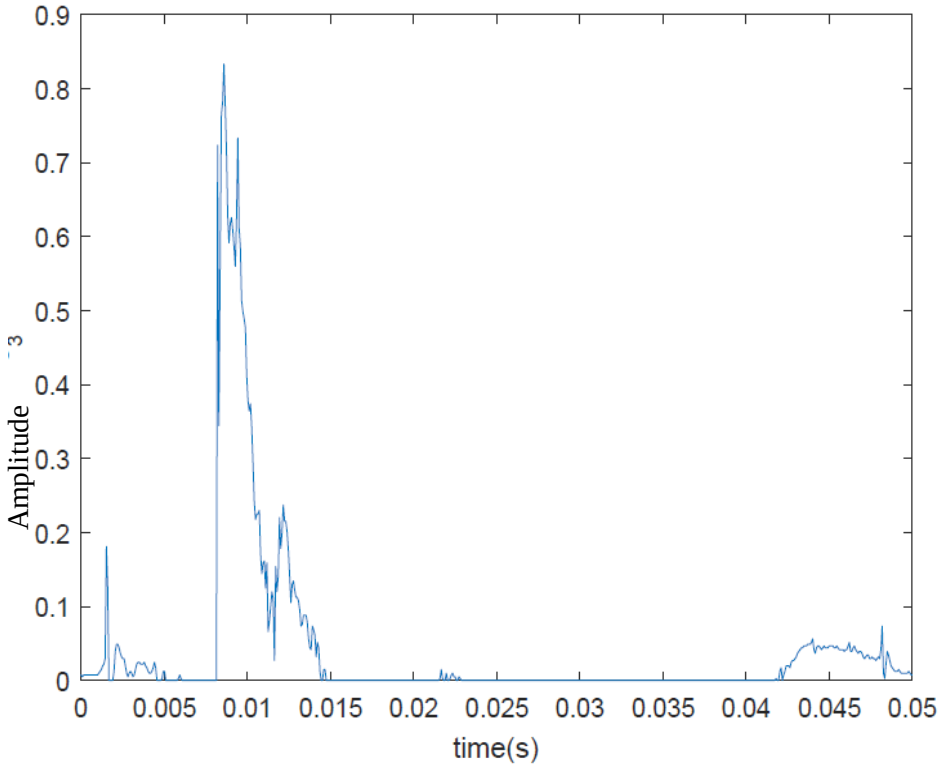

Figure 4.21: MATLAB® plot of pressure pulse at sensor 3

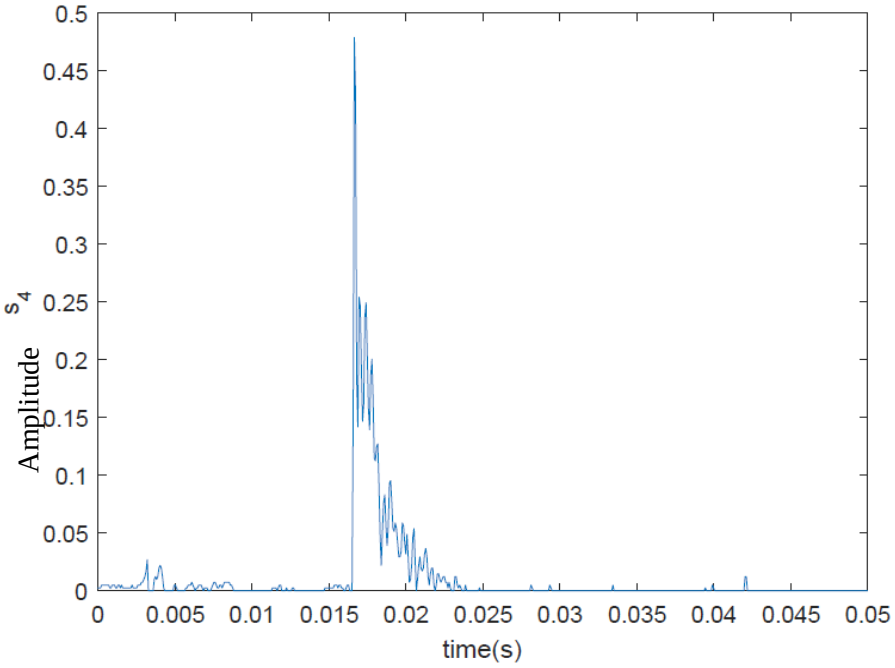

Figure 4.22: MATLAB® plot of pressure pulse at sensor 4

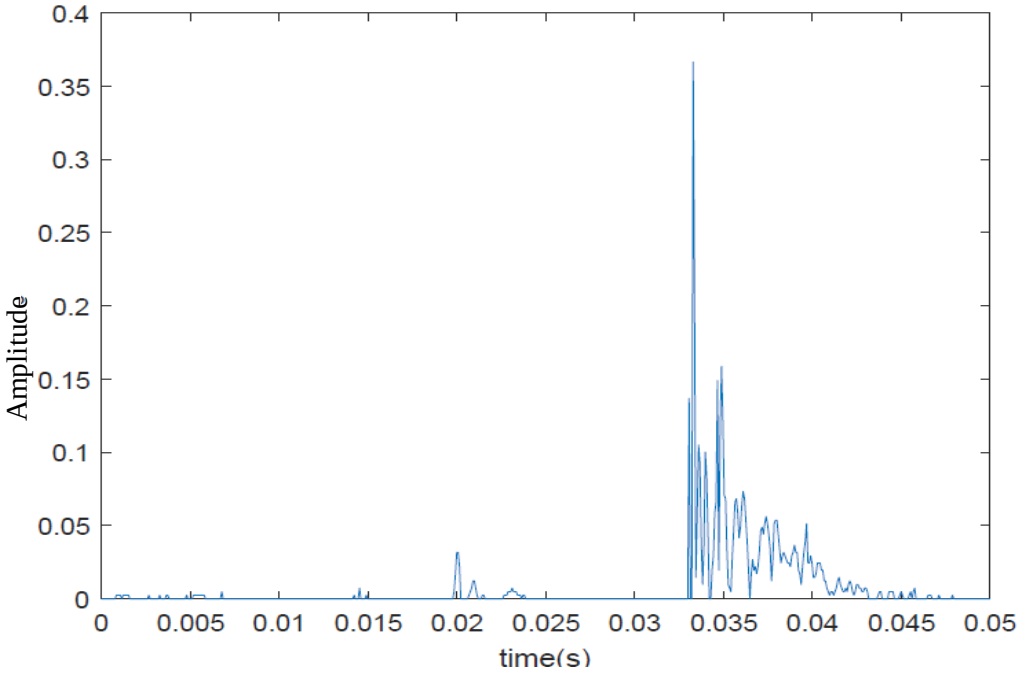

Figure 4.23: MATLAB® plot of pressure pulse at sensor 5

Two major reconstructions of the pressure pulse at s_2, the event location, were carried out. In the first reconstruction process, the original pressure pulse, s_2 was sampled and the obtained samples used to reconstruct the event at s_2. The original pulse s_2 was made up of a total of 1320 samples and the sampling was carried out for 300 samples of s_2 as shown in Figure 4.24.

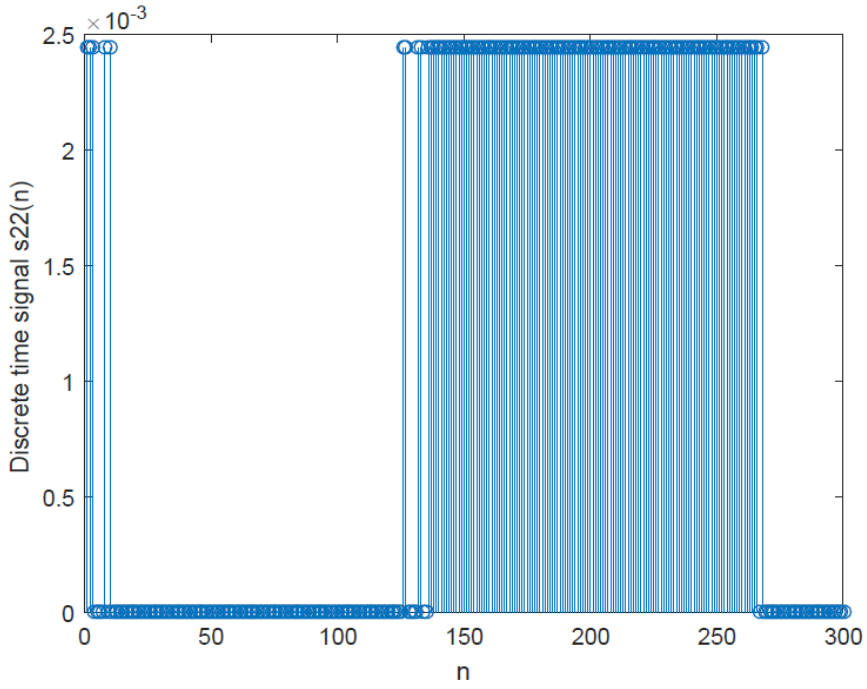

Figure 4.24: Sampling of original pressure pulse s_2 with MATLAB®

The sampling process converted the original pressure pulse s_2 to its discrete form. The original pressure pulse was then recovered from the discrete form of s_2 as shown in Figure 4.25. The original pressure pulse was captured with a single sampling frequency of 13.16 Ks/s. A single sampling frequency of 26.32 Ks/s was used in order to achieve proper reconstruction of event, satisfying the Nyquist criterion. This was done to prevent the occurrence of aliasing of the reconstructed pulse.

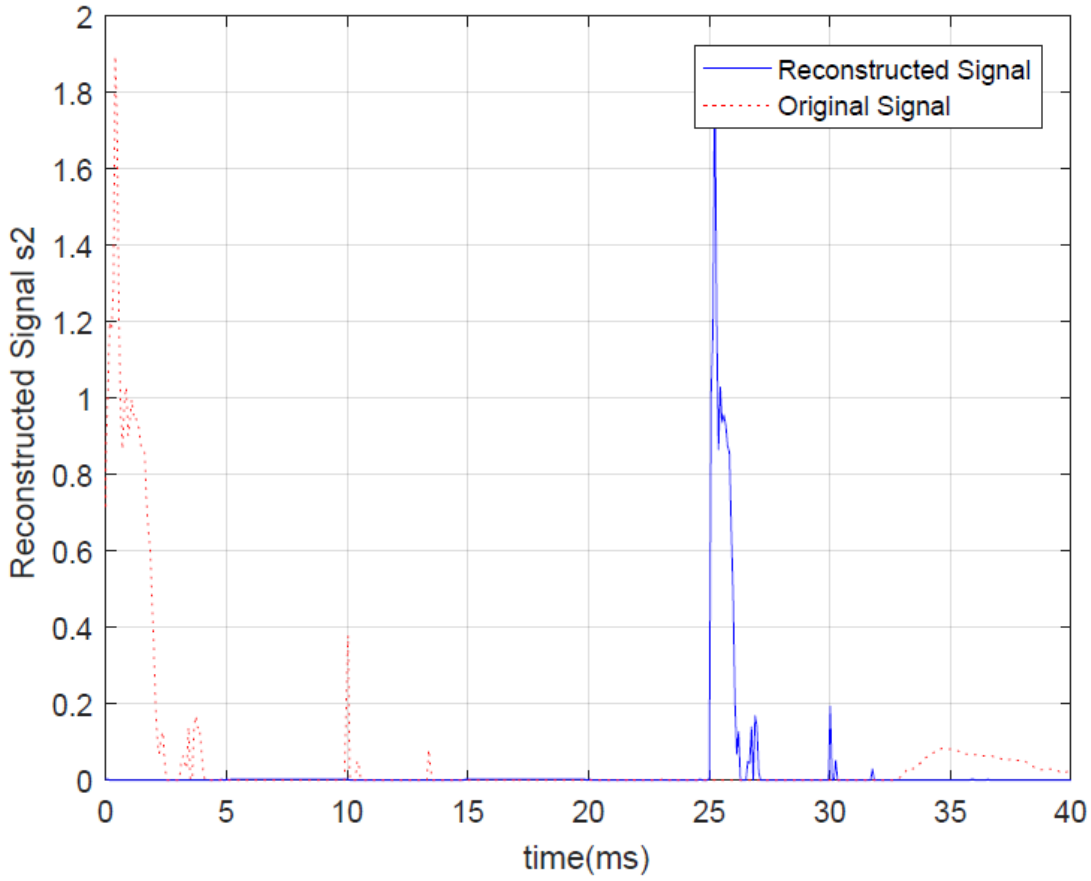

Figure 4.25: Reconstruction of original pressure pulse s_2 from samples of s_2

Figure 4.25 shows a fairly good reconstruction of the actual event as the original pressure pulse and the reconstructed pressure pulse have similar shapes. The amplitude peak of the reconstructed pressure pulse is a little bit underestimated compared to the original pressure pulse. This was consistent with all other reconstructions carried out on other data readings obtained from the several experimental results.

A second reconstruction of the actual event, s_2 was carried out. In this case, an attempt was made to recover the original pressure pulse at the event location from the two closest pressure pulses at the locations closest to the event location. This was done in other to provide a

solution to real life situations where the signal from the sensor at the location of an event was lost or not captured at all. If the other sensors closest to the event captured the signals from the event, these signals can then be used to provide a clue as to what originally happened at the site of the event. In this case, sensors 1 and sensors 3 were closest to sensor 2 at the event location. Therefore, the pressure pulses s_1 and s_3 were sampled respectively as shown in Figure 4.26 and Figure 4.27. Both pressure pulses were made up of a total of 1320 samples and they were both sampled for 300 points each.

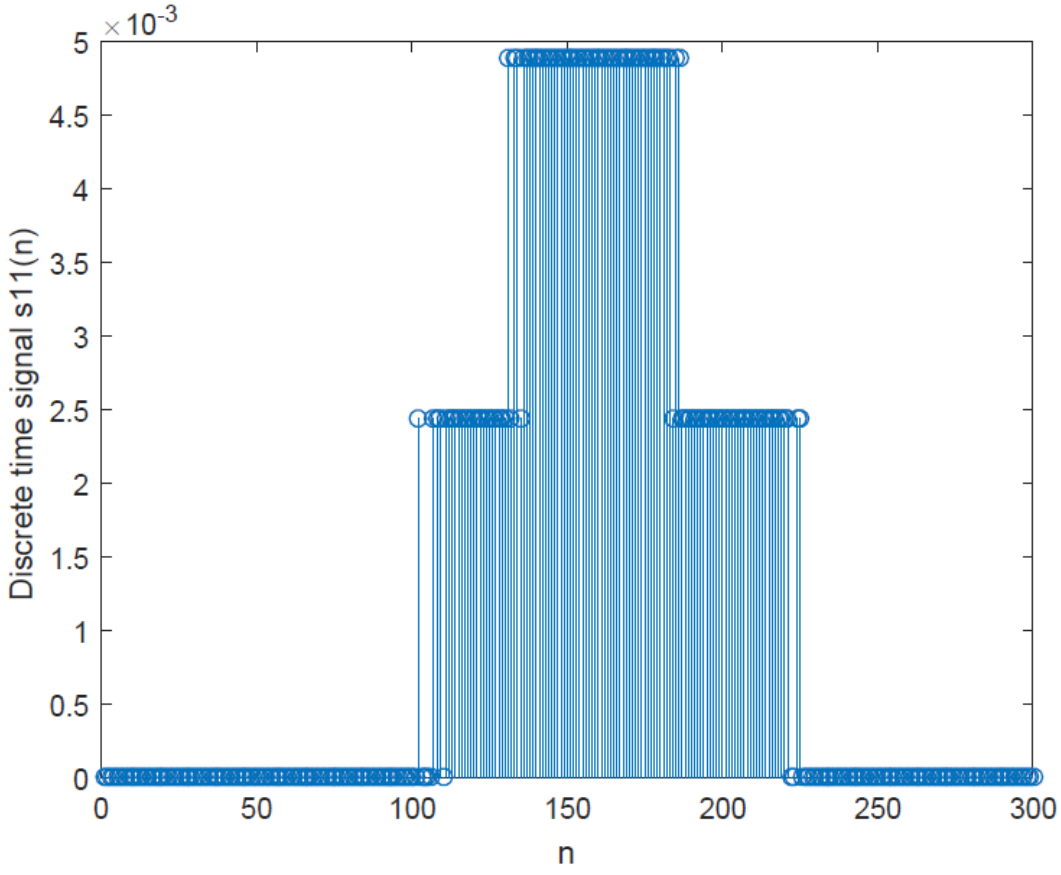

Figure 4.26: Sampling of original pressure pulse s_1 with MATLAB®

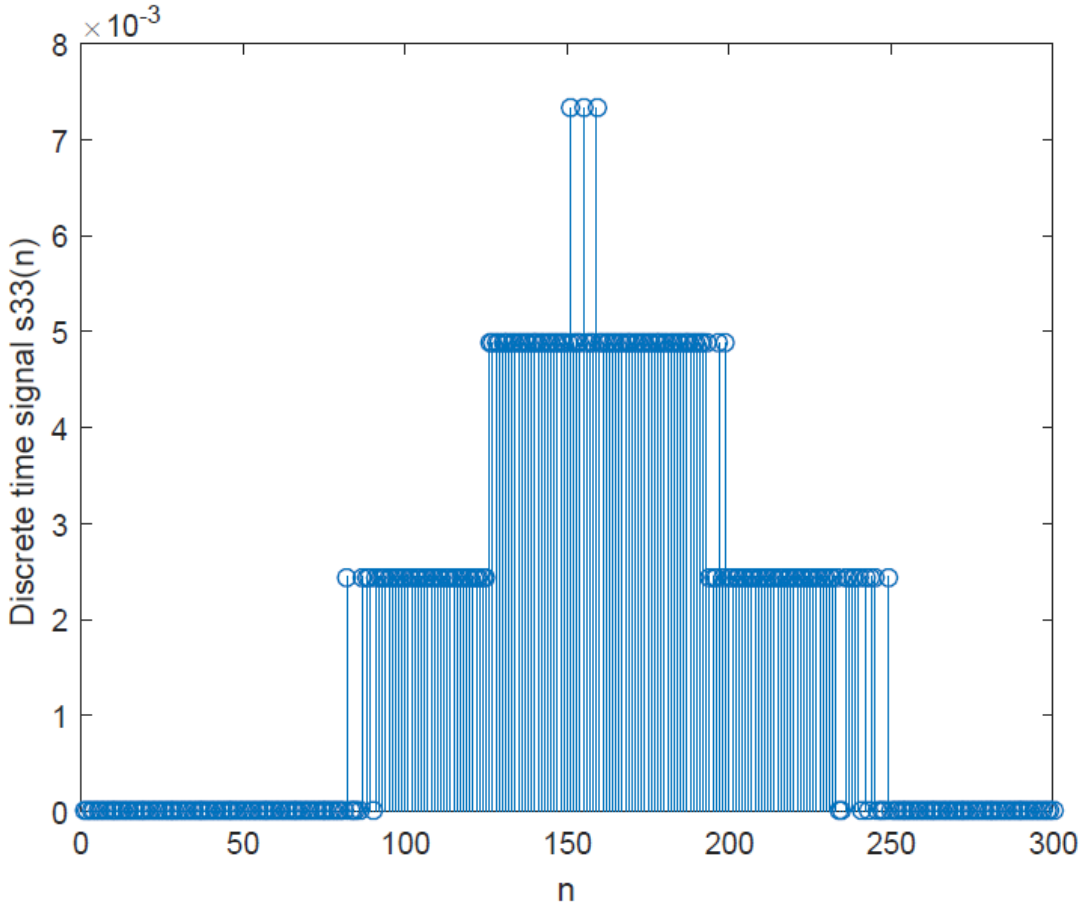

Figure 4.27: Sampling of original pressure pulse s_3 with MATLAB®

The original pressure pulse s_2 was then recovered from the combination of the discrete forms of both s_1 and s_3. Using the reconstruction model in Equation 3.18 in the MATLAB® environment, a combination of these discrete forms gave a reconstruction of s_2 as shown in Figure 4.28. A single sampling frequency of 26.32 Ks/s was used. This was double the value of the sampling frequency that was used in the experiment so as to satisfy the Nyquist criterion and ensure proper reconstruction. A good reconstruction of the actual event was achieved as the reconstructed original pressure pulse s_2 in Figure 4.28 had a similar shape with that of s_2 in Figure 4.20. As before, it was observed that they was a difference in the

magnitude of s_2, as the reconstructed pulse was underestimated compared to that in Figure 4.20 showing the plot of s_2.

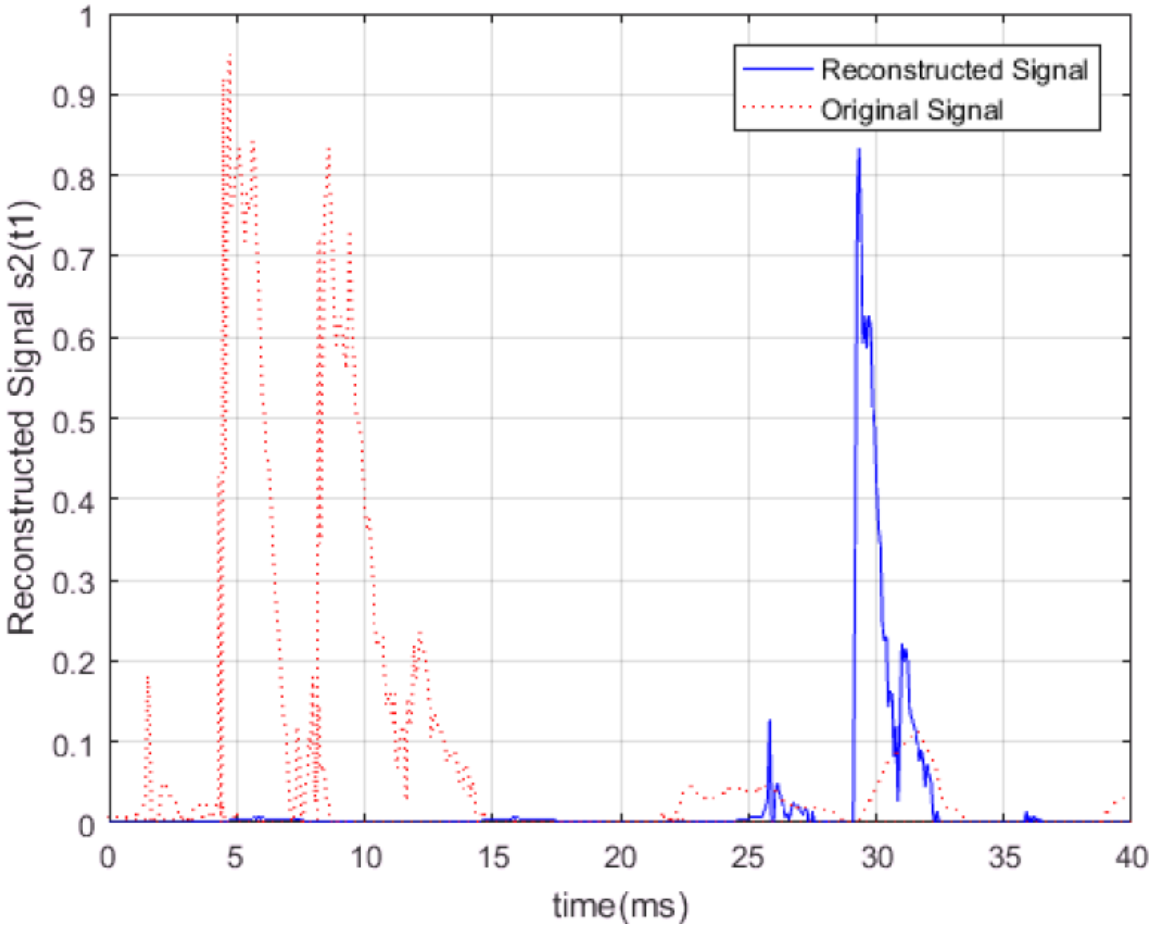

Figure 4.28: Reconstruction of original pressure pulse s_2 from pressure pulses s_2 and s_3

The magnitude of s_2 in Figure 4.28 differed by 45 % compared to its magnitude in Figure 4.20. This was consistent for all the repeated tests and the underestimation ranged between 40 to 45 %. To solve the problem of underestimation of the reconstructed pulse as had been observed with several experimental data processed, a reconstruction factor, K was introduced as seen in Equation 3.19. K is an approximation factor obtained by dividing the maximum value or amplitudes of the original pressure pulse, $x_o(t)$ by the maximum value of the

reconstructed pressure pulse, $x_r(t)$. Comparing Figure 4.28 and Figure 4.20, it is observed that the original pressure pulse in Figure 4.20 has amplitude of 1.9 while the reconstructed pulse in Figure 4.28 has amplitude of 0.85. Therefore K is calculated as:

$$K = \frac{1.9}{0.85} = 2.235$$

Using this obtained K value in Equation 3.19 in MATLAB®, a better reconstruction is obtained as shown in Figure 4.29.

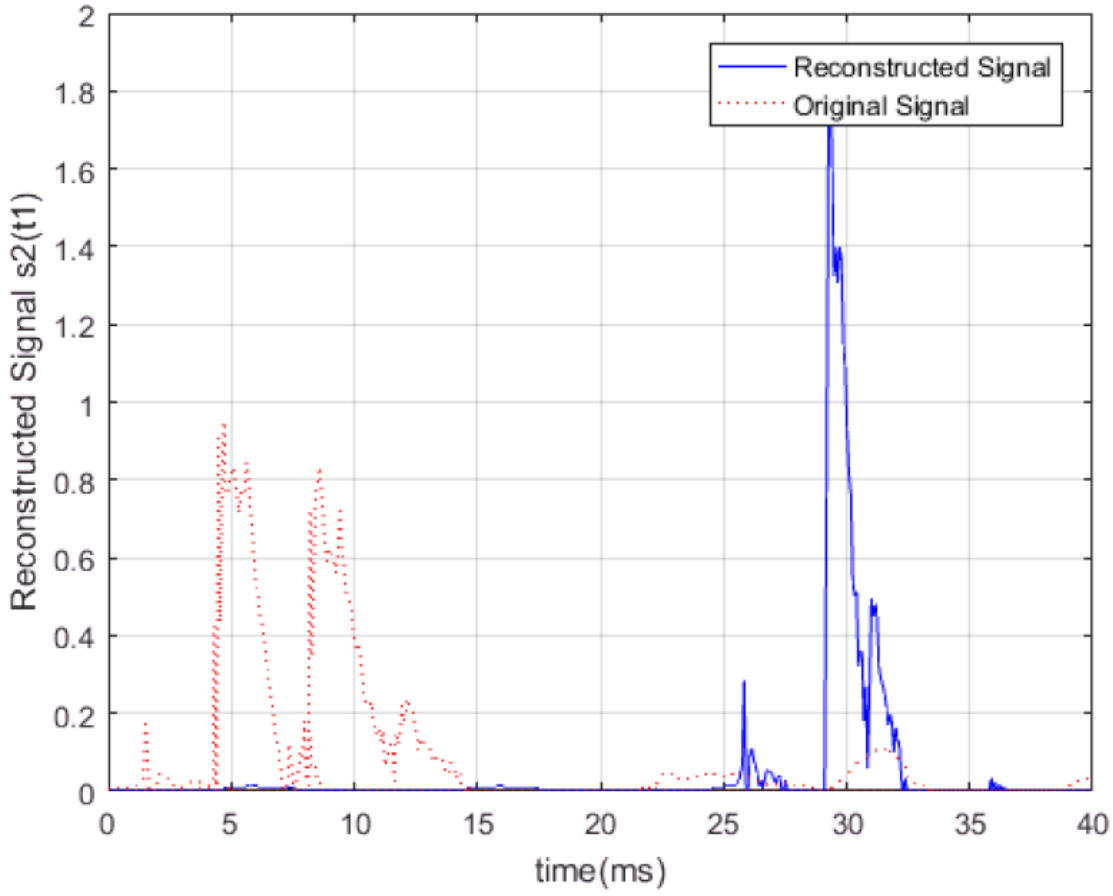

Figure 4.29: Reconstruction of original pressure pulse s_2 from pressure pulses s_1 and s_3 using reconstruction factor

Figure 4.29 shows that a good reconstruction of an event is better achieved by applying a factor K to the developed Fourier transform based model. This helped to solve the problem of underestimation of the nodes of the reconstructed signal that was observed with all the data samples that were processed.

4.7.2 Damage Pattern Recognition

One very important aspect of this work that was obtained from the experimental data is that a broad range of damages can be detected by this system. It was observed that each damage scenario generated its own pressure pulse that was peculiar to it when compared to the pressure pulse from another damage event. The four different experiments to simulate damage caused by explosion, drilling, vehicular motion and corrosion confirmed this.

4.7.2.1 Damage from Explosion

Figures 4.30 and 4.31 show the results from experiments to mimic explosion at two different readings. The results from sensor 1 in both readings are thus displayed.

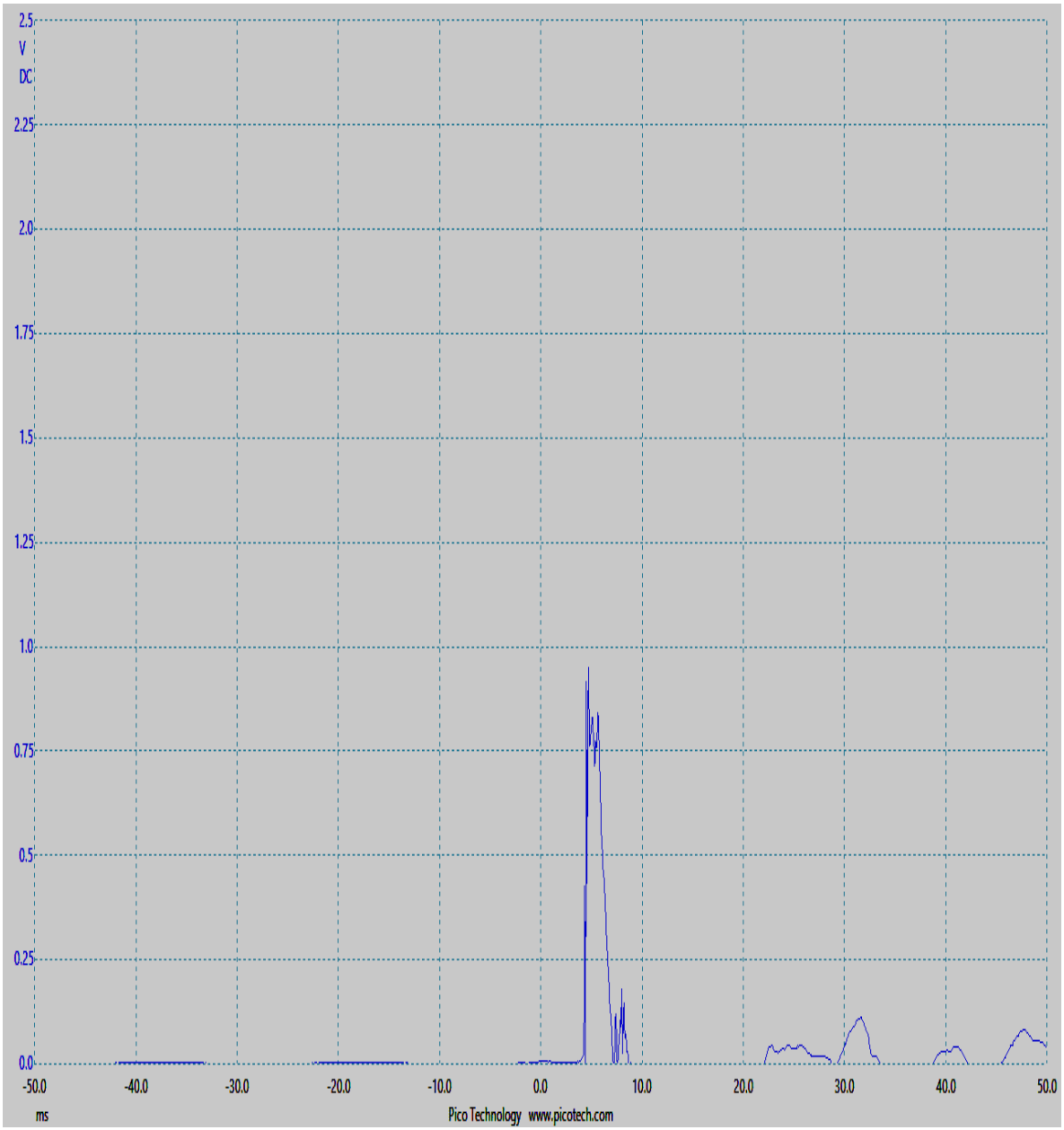

Figure 4.30: Typical pressure pulse measured at experimental rig to mimic damage due to explosion (0.8 bar pressure reading at single sensor)

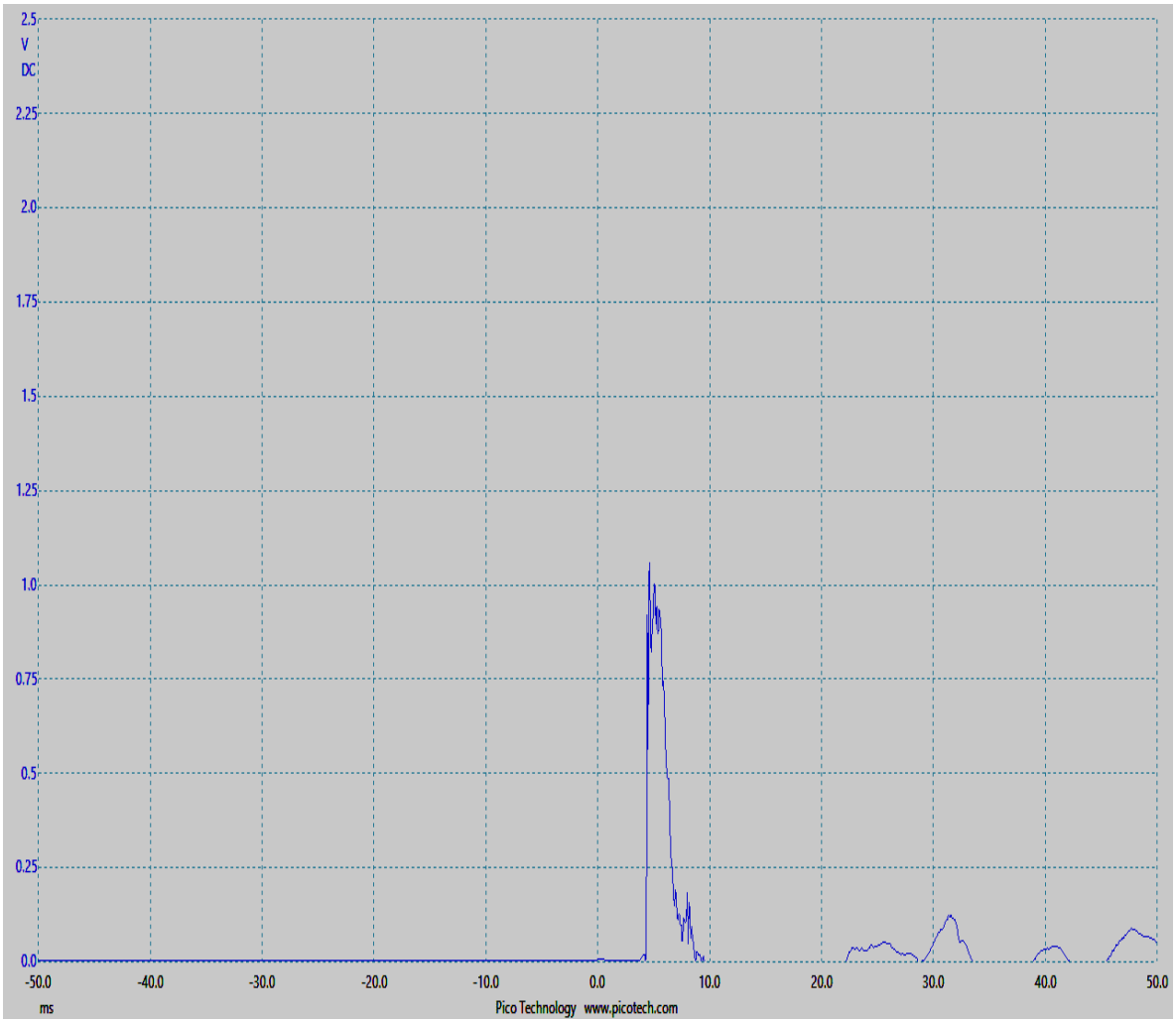

Figure 4.31: Typical pressure pulse measured at experimental rig to mimic damage due to explosion (1 bar pressure reading at single sensor)

Figure 4.30 and Figure 4.31 show the typical pressure pulse that was obtained from the experimental test rig for tests to mimic damage caused by explosion. Only the amplitudes of the pulses differed due to the different pressure readings used as was obtainable in the study by Olugboji, (2011).

4.7.2.2 Damage from Drilling

The captured pressure pulses from the experiments to mimic drilling are are shown in Figures 4.32, 4.33, 4.34 and 4.35 respectively.

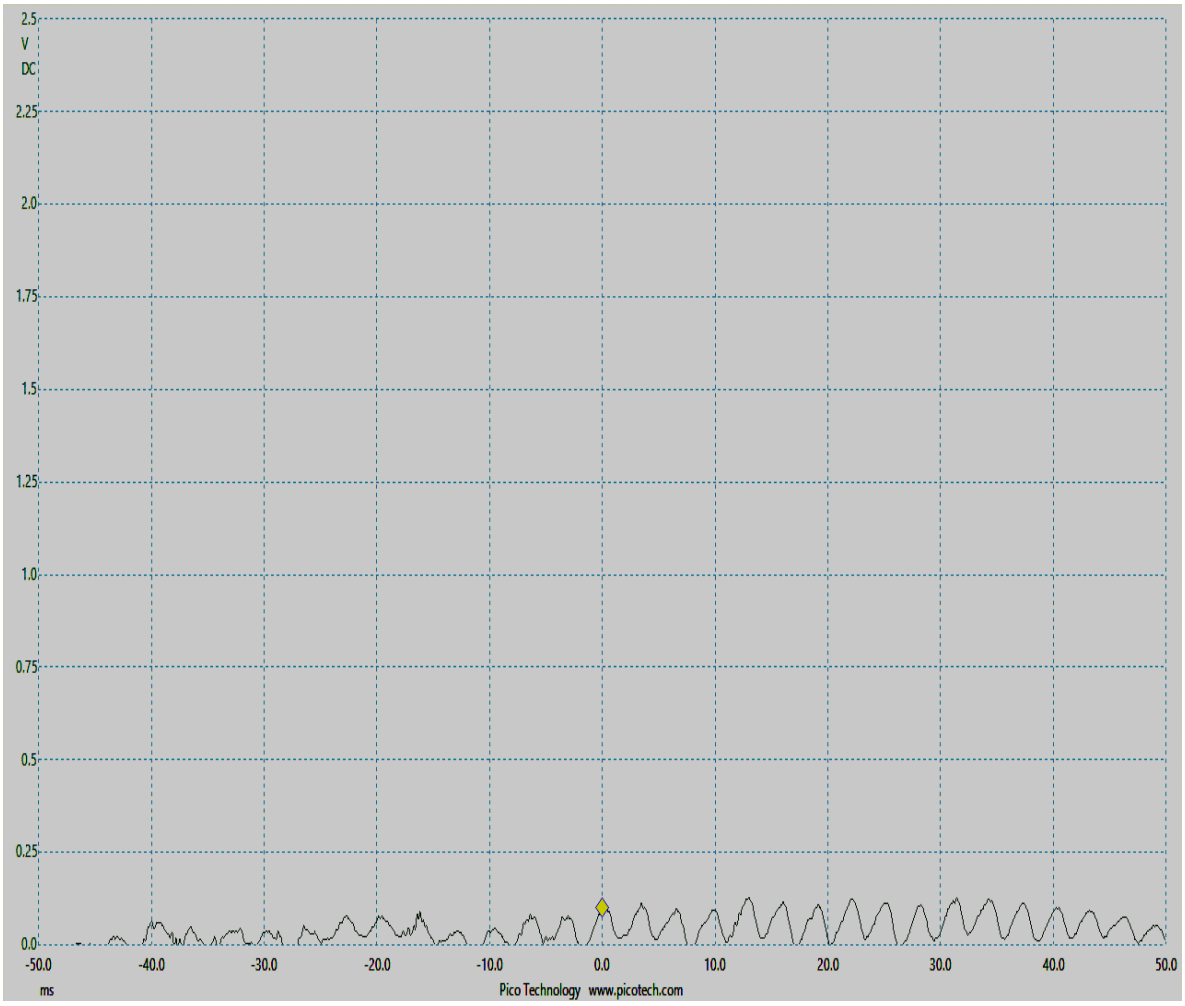

Figure 4.32: Typical pressure pulse measured at experimental rig to mimic damage due to drilling with air as transport fluid (1000 Watts drill input power at single sensor)

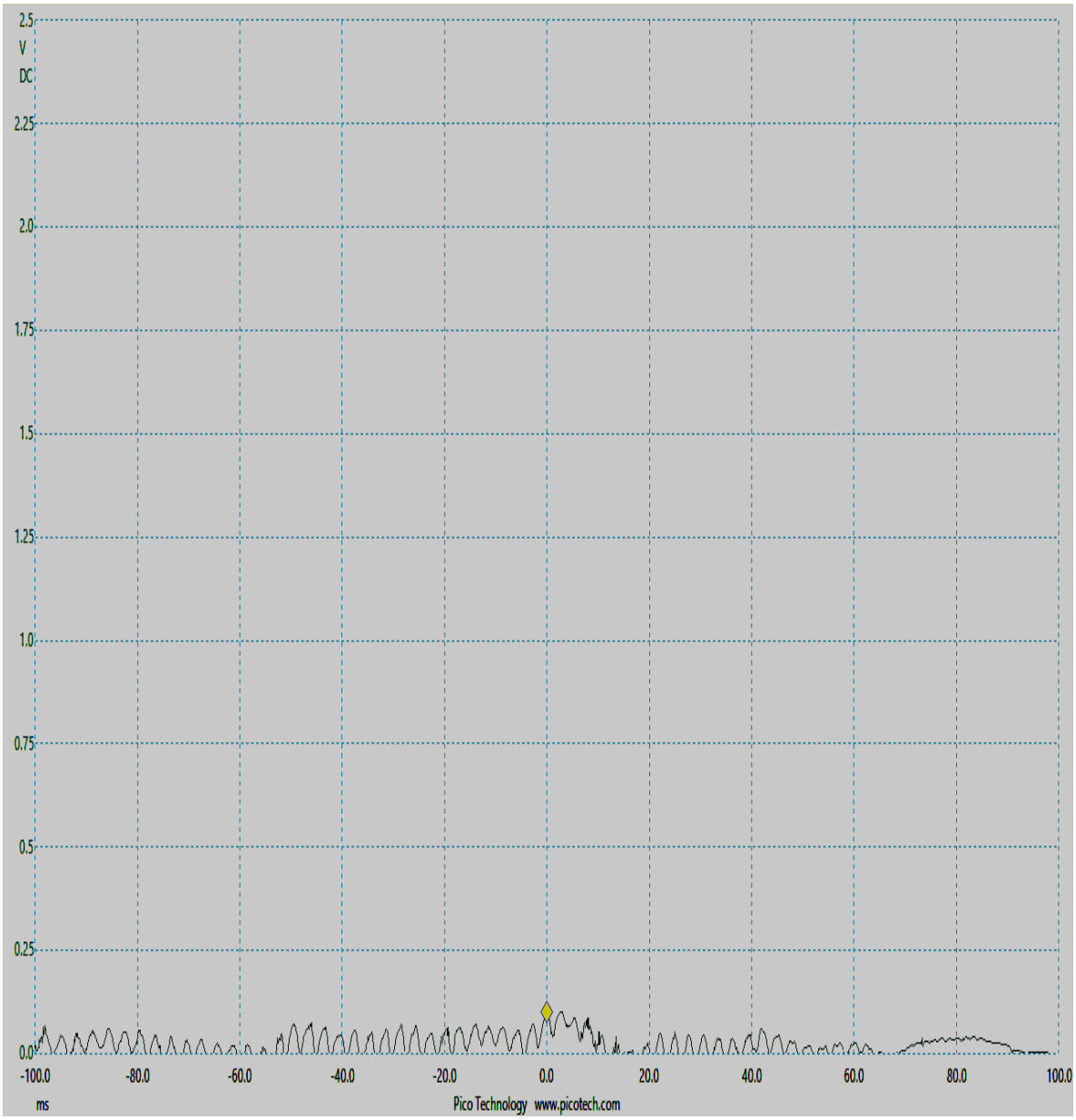

Figure 4.33: Typical pressure pulse measured at experimental rig to mimic damage due to drilling with air as transport fluid (750 Watts drill input power at single sensor)

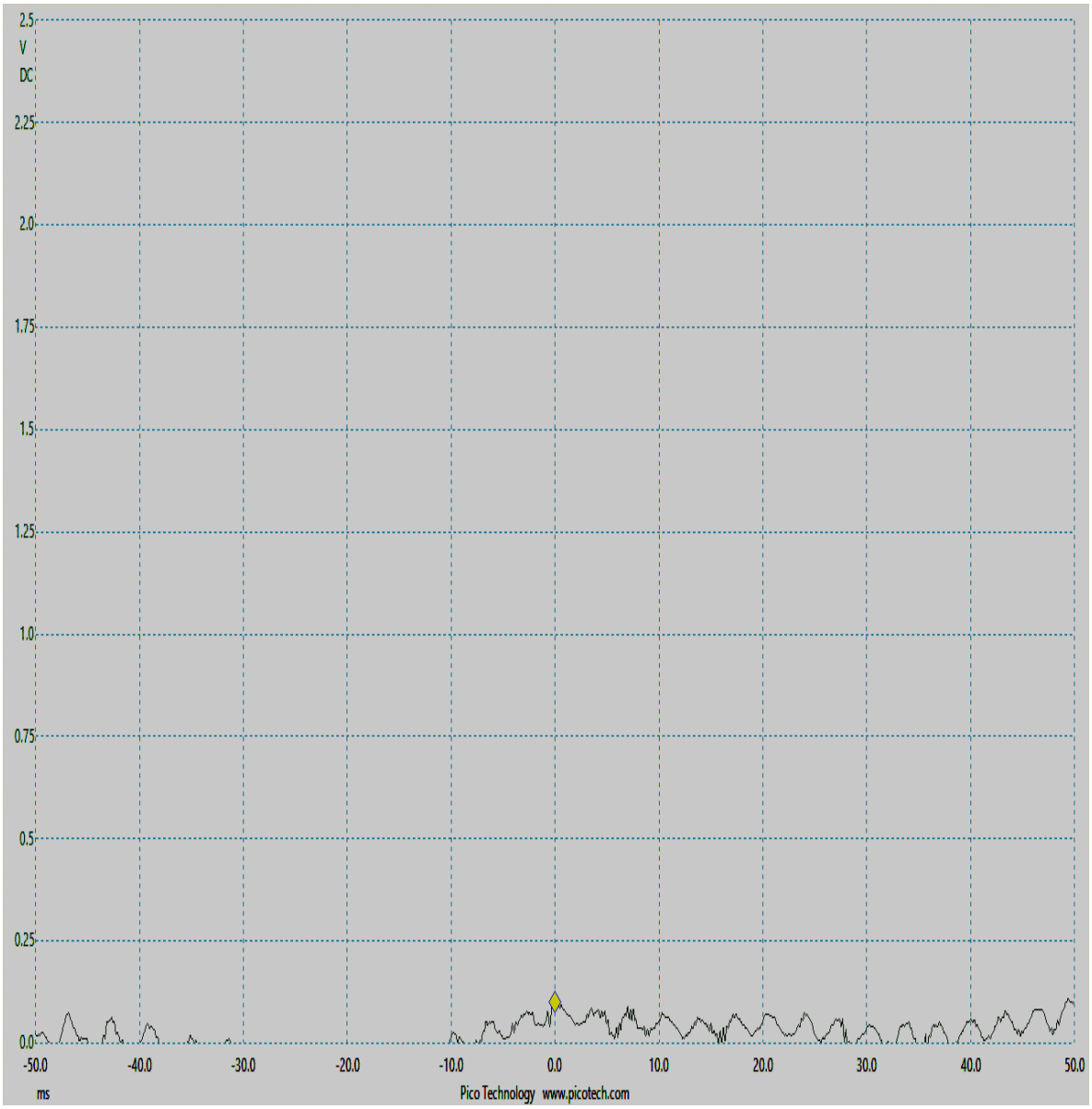

Figure 4.34: Typical pressure pulse measured at experimental rig to mimic damage due to drilling with water as transport fluid (1000 Watts drill input power at single sensor)

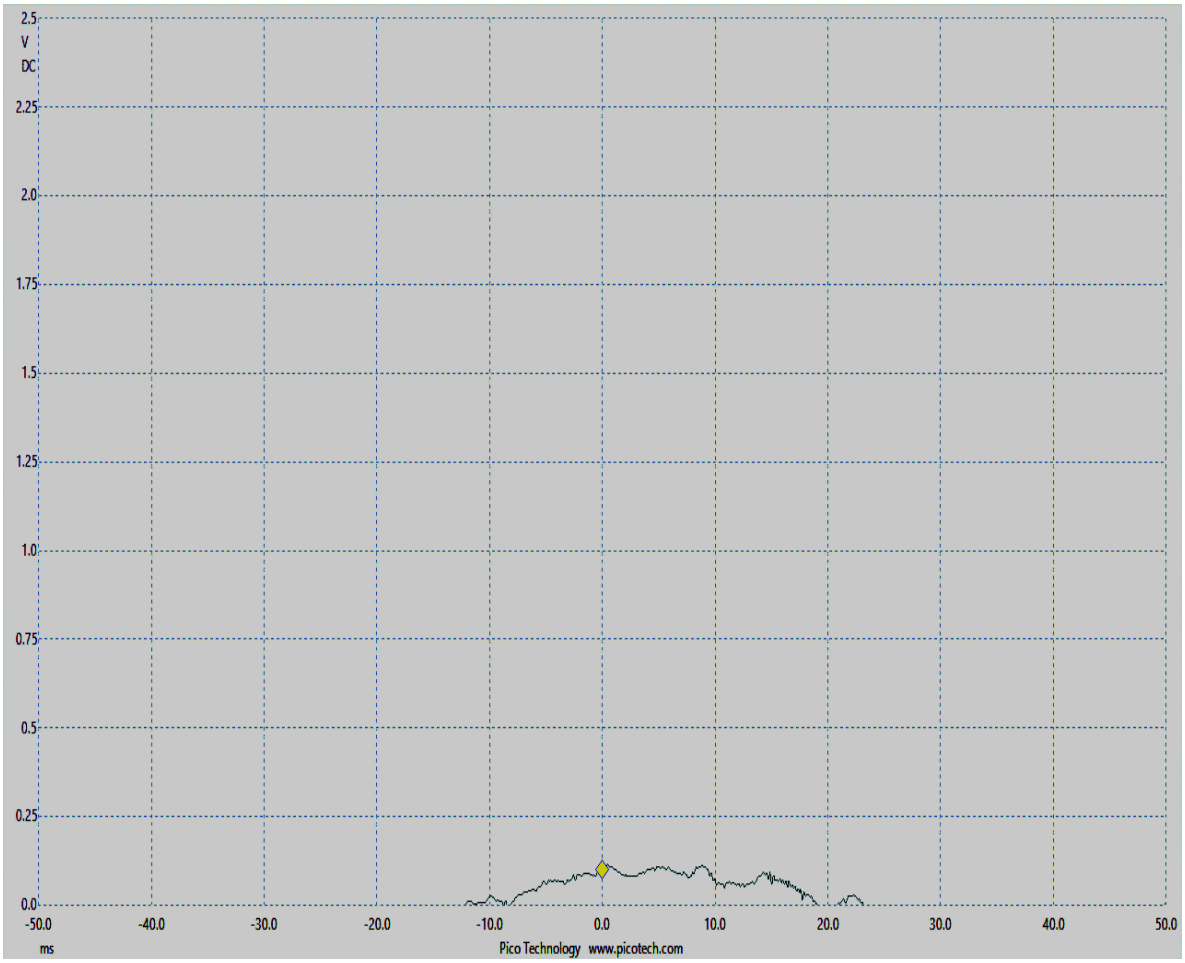

Figure 4.35: Typical pressure pulse measured at experimental rig to mimic damage due to drilling with water as transport fluid (750 Watts drill input power at single sensor)

Figure 4.32 and Figure 4.33 show the typical pressure pulse for a test to mimic a drilling operation on an air-filled pipe while Figures 4.34 and 4.45 show the typical pressure pulse for a test to mimic a drilling operation on a water-filled pipe. These pressure pulses were consistent for all the 30 repeated tests with 5 tests carried out at each sensor using either the 750 Watts or the 1000 Watts input power hand drill for both types of transport media. The results show that the pressure pulses from a drilling operation were different from that of an explosion as evident in the shape and amplitudes of the measured pressure pulses. This

further proved that damage events on pipes can be characterised according to what caused the damage.

4.7.2.3 Damage from Vehicular Motion

The results of these experiments are thus shown in Figures 4.36, 4.37, 4.38 and 4.39.

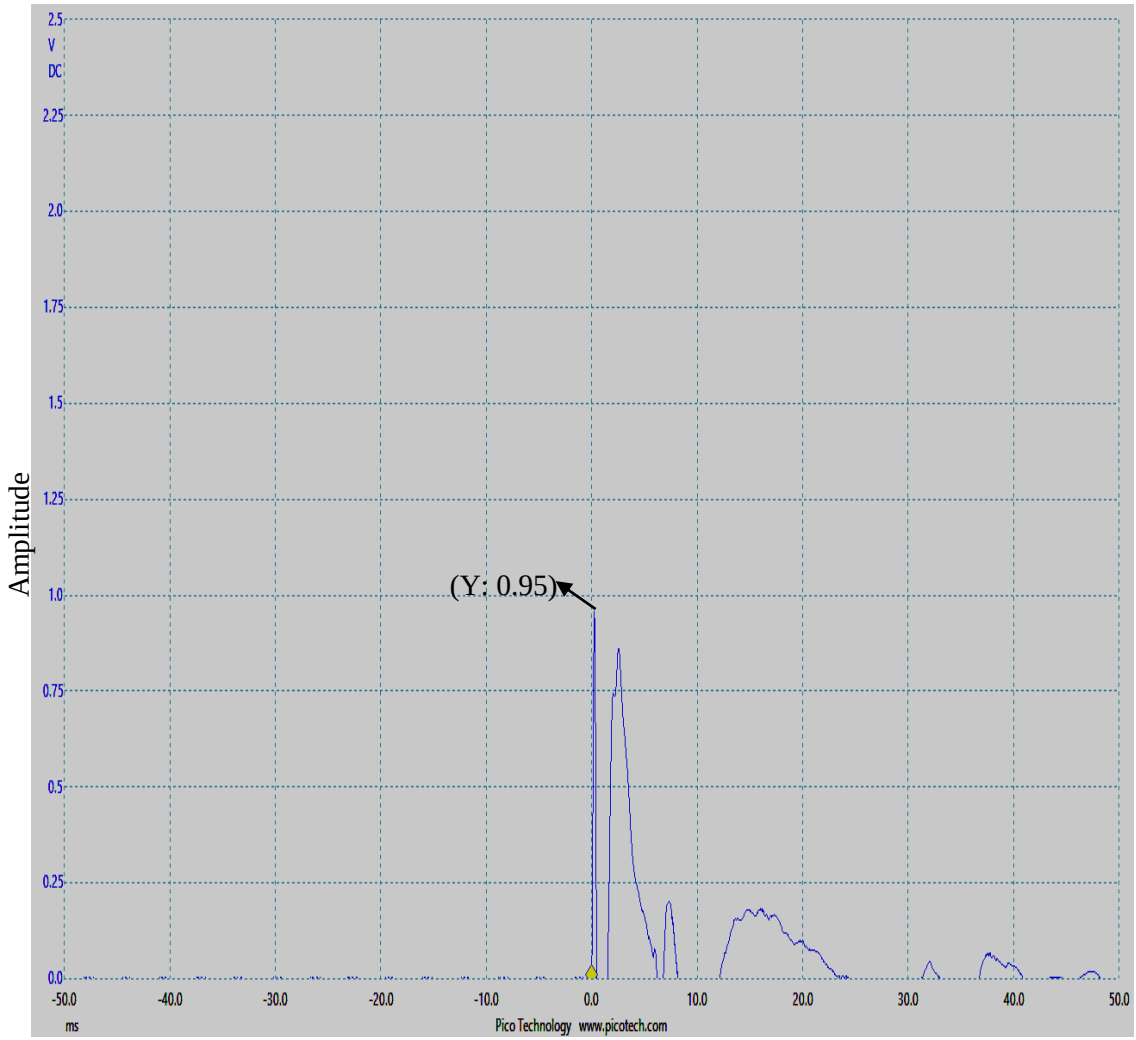

Figure 4.36: Typical pressure pulse measured at experimental rig to mimic damage due to vehicular motion with air as transport fluid (measurement at sensor 2)

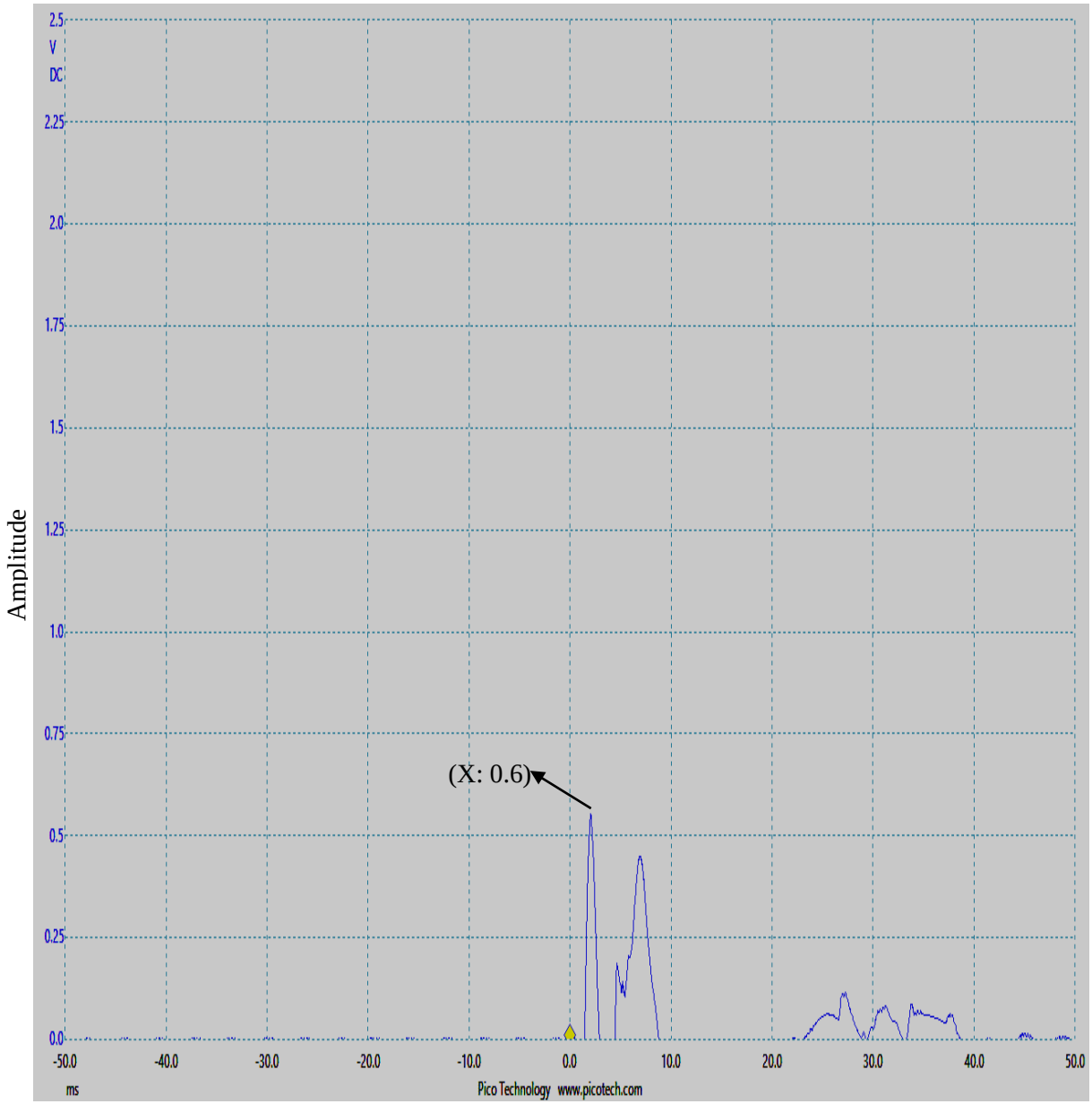

Figure 4.37: Typical pressure pulse measured at experimental rig to mimic damage due to vehicular motion with air as transport fluid (measurement at sensor 3)

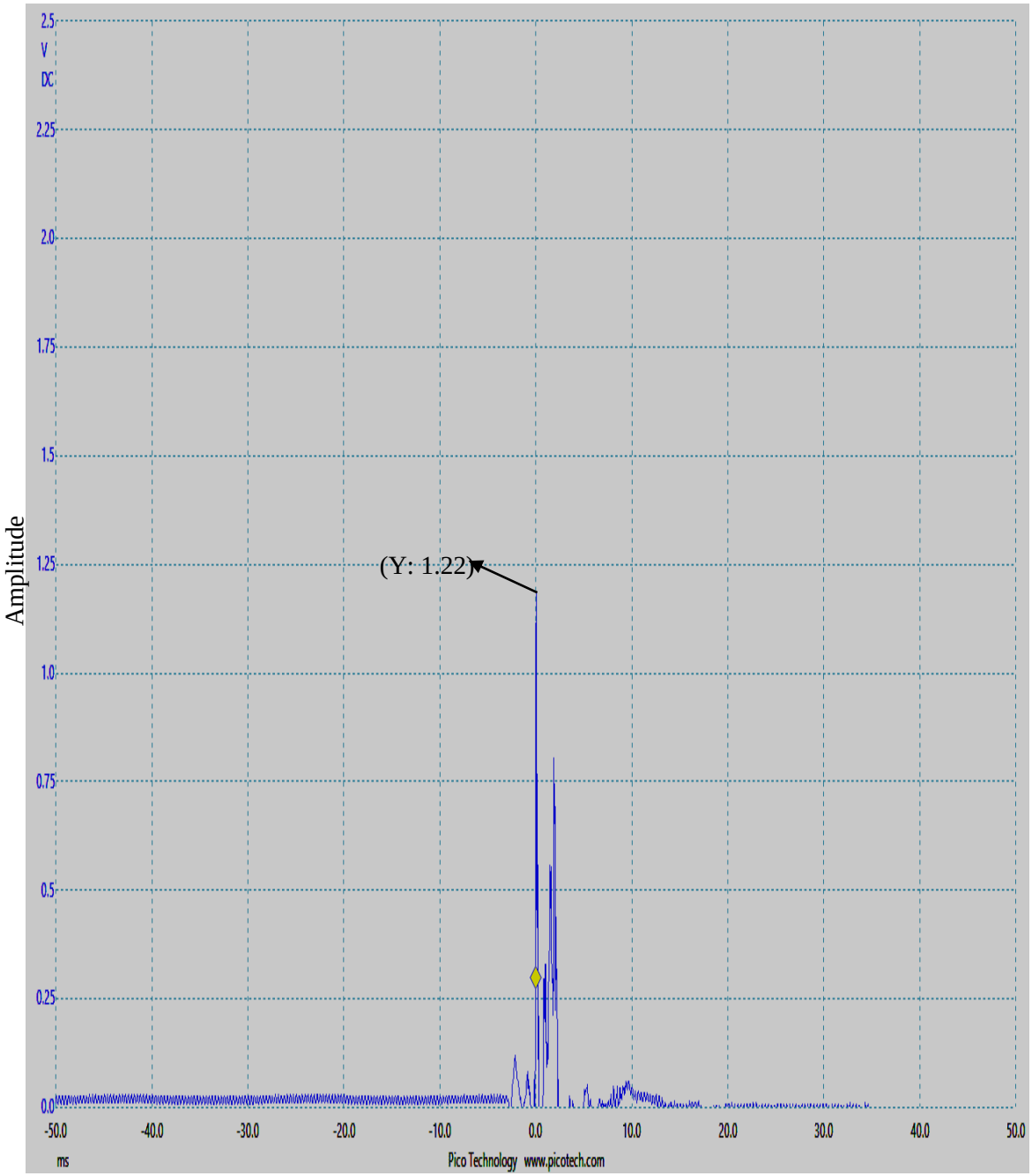

Figure 4.38: Typical pressure pulse measured at experimental rig to mimic damage due to vehicular motion with water as transport fluid (measurement at sensor 2)

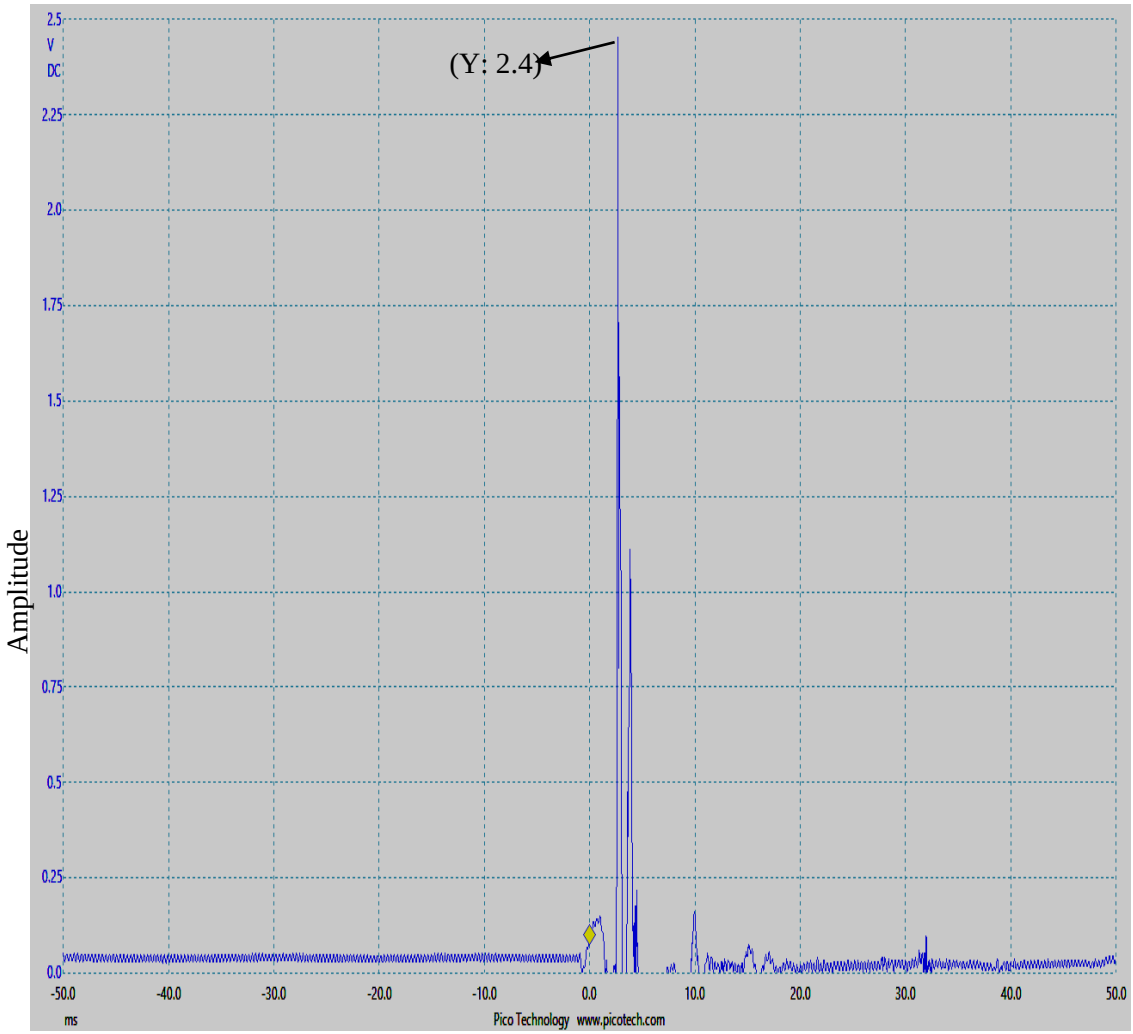

Figure 4.39: Typical pressure pulse measured at experimental rig to mimic damage due to vehicular motion with water as transport fluid (measurement at sensor 3)

Figure 4.36 and Figure 4.37 show the typical pressure pulse for a test to mimic the movement of a vehicle over an air-filled pipe while Figures 4.38 and 4.39 show the typical pressure pulse for a test to mimic the movement of a vehicle over a water-filled pipe. The measured pressure pulses were also consistent in shape for all the 30 repeated tests with 5 tests carried out at each sensor using a small wooden-handle hammer in both transport fluid types. All the outputted pulses were similar in their spectra, and quickly attenuated. The results show that the pressure pulses from a heavy mass as that obtainable from the mass of a vehicles' tyres

are different from that of an explosion or drilling operation as evident in the shape and amplitudes of the measured pressure pulses.

4.7.2.4 Damage from Corrosion

To mimic damage from corrosion, a corrosive environment was created within the pipe. The pipe was connected to a water tank and water was allowed to flow through the pipe. A pipette was used to introduce acid in varying quantities into the water filled pipe. At first, 2 ml of acid was introduced into the pipe; then 5 ml, 10ml and 25 ml respectively. The acid was poured first in small amounts, with the volume poured later increased. This was done this way because no changes were observed at 2 ml and 5 ml of acid poured, so the quantity poured had to be scaled up. 98.08% concentrated Sulphuric acid was used and it was intended that the introduction of the acid on the surface of the pipe would create an environment corrosive enough to generate sharp fronted pulses within the pipe in order to characterise damage on a pipe due to corrosion . Substantial pressure pulses were noticed only when 10 ml and 25 ml of the acid were introduced. This could have been as a result of the increase in the resistivity of the piezoelectric sensors after it came in contact with water. Also repeated tests could not be carried out because the piezoelectric sensors lost their conductivity after sensing as they came in contact with the acid and got corroded. The results of these experiments are thus displayed in Figures 4.40 and 4.41 below.

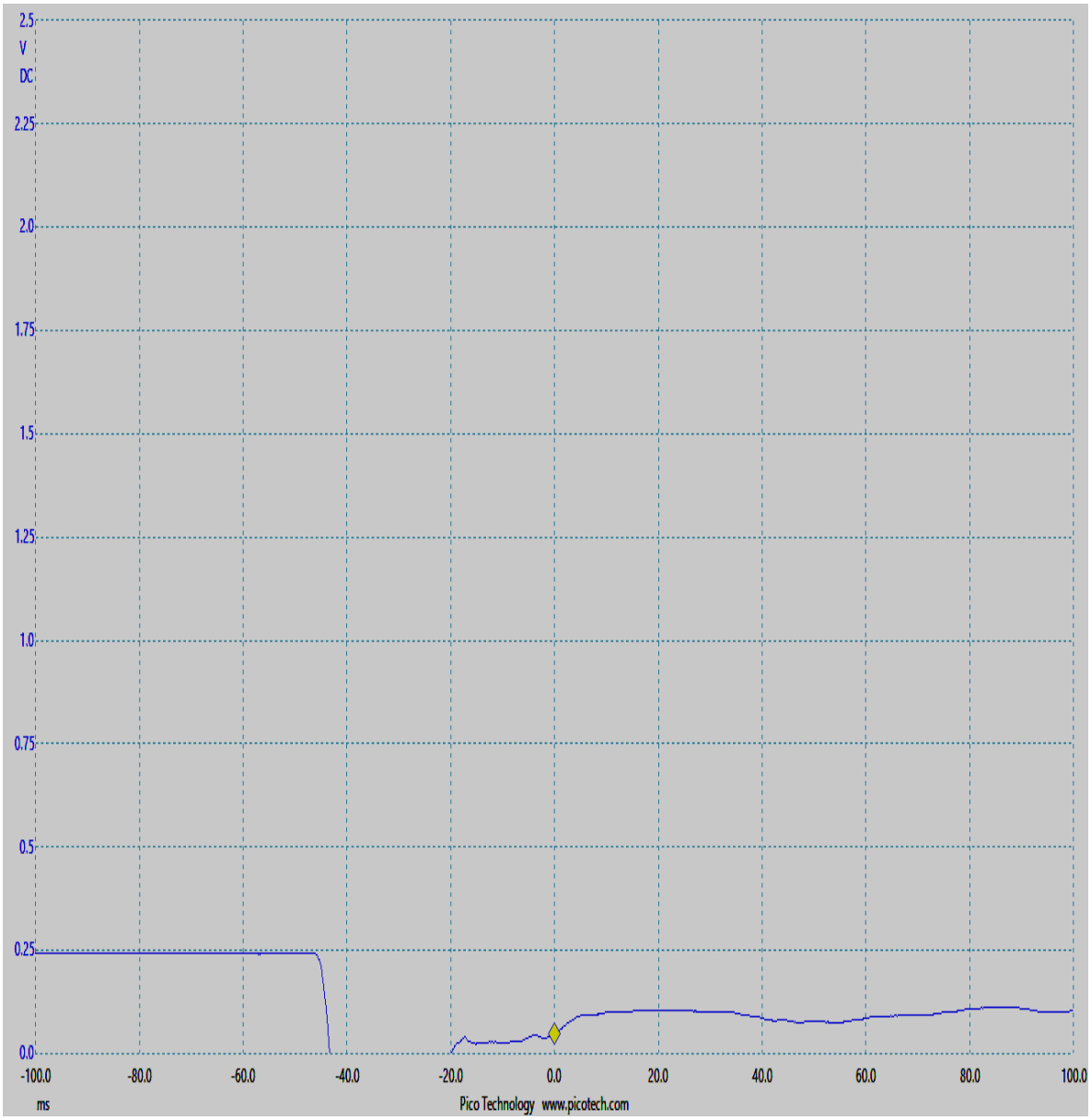

Figure 4.40: Typical pressure pulse measured at experimental rig to mimic damage due to acid (25 ml of acid)

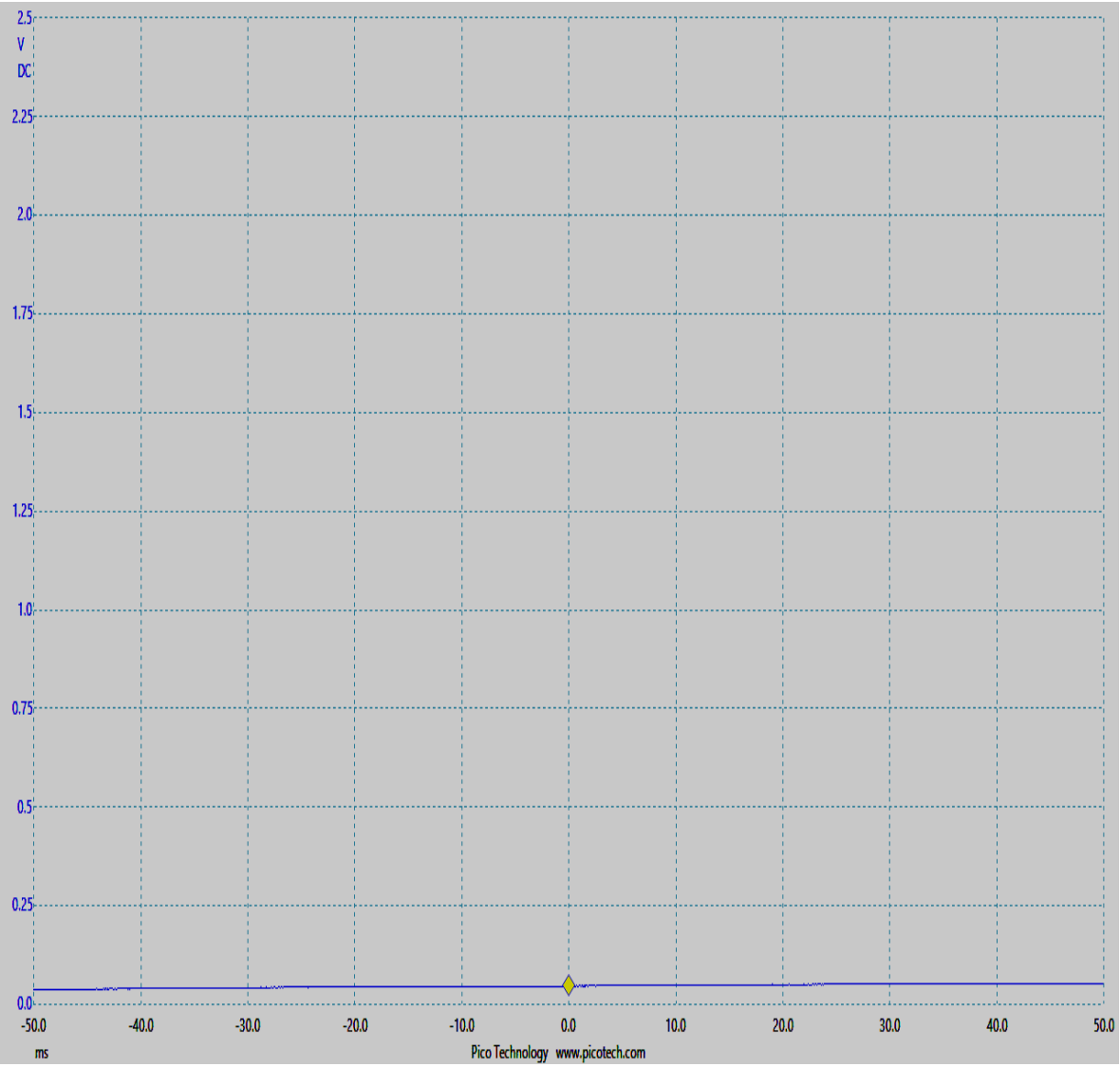

Figure 4.41: Typical pressure pulse measured at experimental rig to mimic damage due to acid (10 ml of acid)

Figure 4.40 and Figure 4.41 show the typical pressure pulse from an experiment to mimic damage due to corrosion. The results obtained when 10 ml of acid was used showed very little pulse while a more visible pulse was observed when 25 ml acid was used. The pattern in both cases of acid quantity was not consistent to develop a pattern, but the results were different from those observed in the other damage scenarios. This proved that the pulses from

a damage event caused by corrosion is different from those obtained from damage events caused by other impacts.

4.8 Wireless Processing and Transmission of Data

The free student license of the ThingSpeak platform was used for this work and as a result, there was a 15 second delay in the transfer of pulse data to the platform (ThingSpeak, 2019). To achieve a one second transfer of pulse data on the platform, a professional license was required. Using a pulse generator, sharp fronted pulses were generated into the pipe and these captured pressure pulses were transmitted wirelessly via the wireless communication device to the ThingSpeak platform. Figure 4.42, Figure 4.43, Figure 4.44, Figure 4.45, and Figure 4.46 show the measured pressure pulses at sensor 1 (s_1); sensor 2 (s_2); sensor 3 (s_3); sensor 4 (s_4); and sensor 5 (s_5) respectively for a pressure reading of 0.8 bar in the pulse generator. Figures 4.47, 4.48, 4.49, 4.50, and 4.51 show the measured pressure pulses at all five sensors for a pressure reading of 1 bar in the pulse generator.

The results from each sensor channel on the ThingSpeak platform were saved and imported from the platform as a Microsoft Excel file. This was then imported into the MATLAB® environment for proper analysis. The experimental setup was same as before with sensor 2 (s_2) closest to the tee connection and taken as the damage location. On the ThingSpeak platform, sensor 1 readings were displayed on field 1 while sensor 2 readings were displayed on field 5. Sensor 3, 4 and 5 readings were displayed on fields 3, 2 and 5 respectively on the ThingSpeak platform.

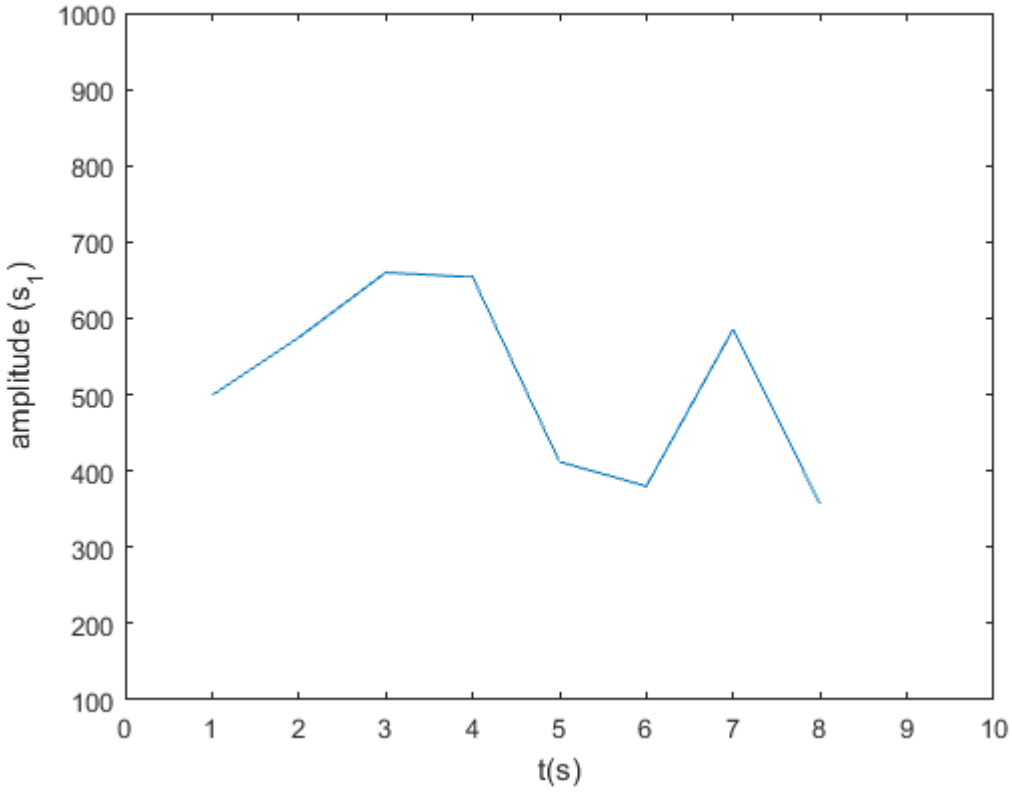

Figure 4.42: MATLAB® representation of measured pressure pulses at all five sensors for a 0.8 bar pressure reading in the pulse generator

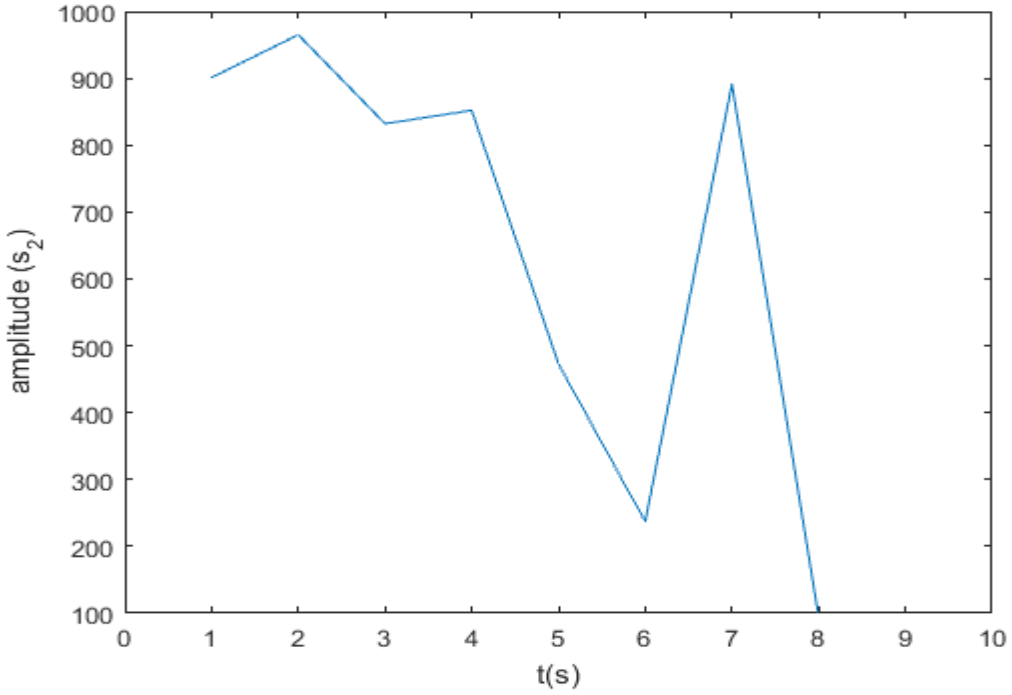

Figure 4.43: MATLAB® representation of measured pressure pulse at sensor 2 using the wireless communication device for a 0.8 bar pressure reading

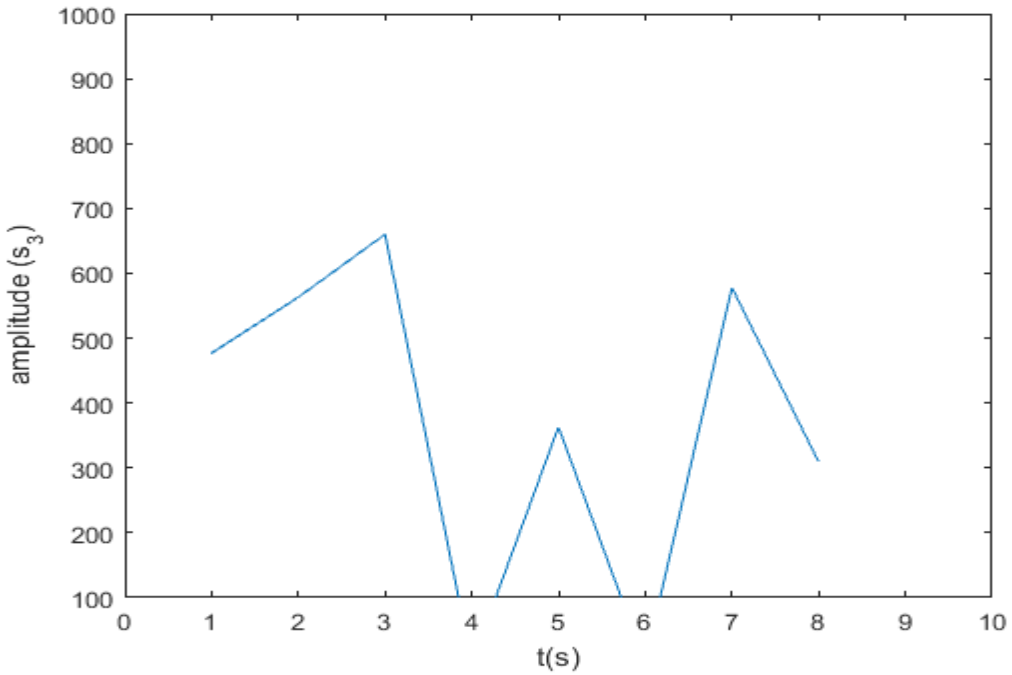

Figure 4.44: MATLAB® representation of measured pressure pulse at sensor 3 using the wireless communication device for a 0.8 bar pressure reading

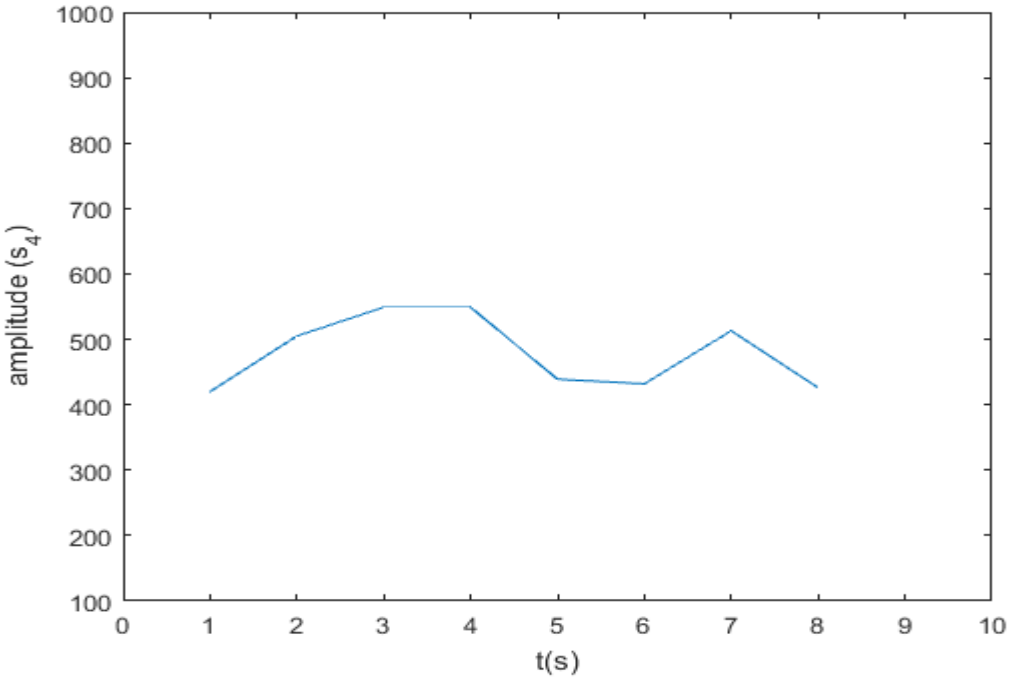

Figure 4.45: MATLAB® representation of measured pressure pulse at sensor 4 using the wireless communication device for a 0.8 bar pressure reading

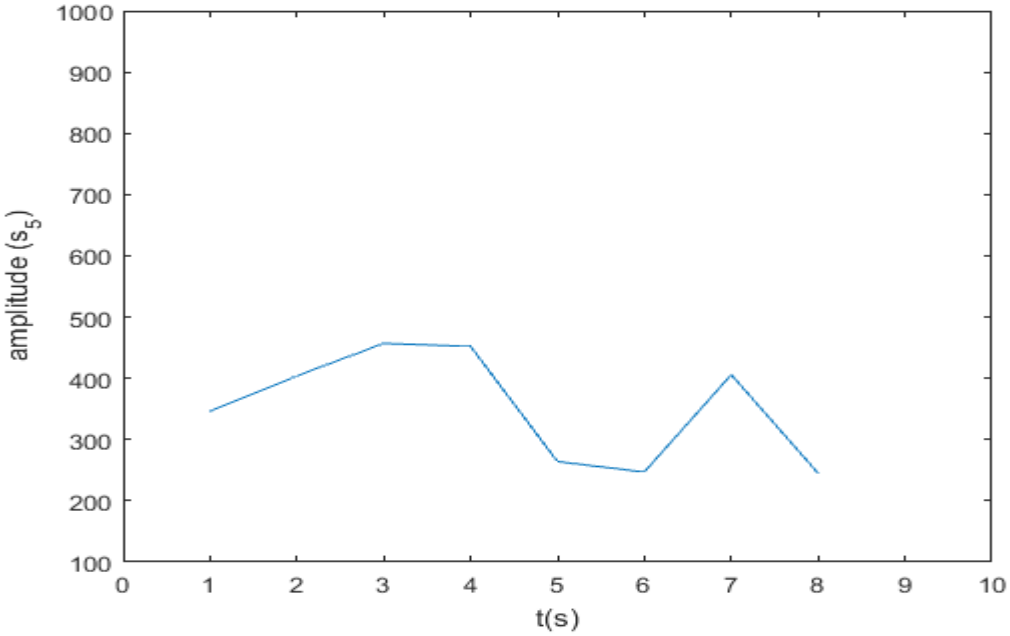

Figure 4.46: MATLAB® representation of measured pressure pulse at sensor 5 using the wireless communication device for a 0.8 bar pressure reading

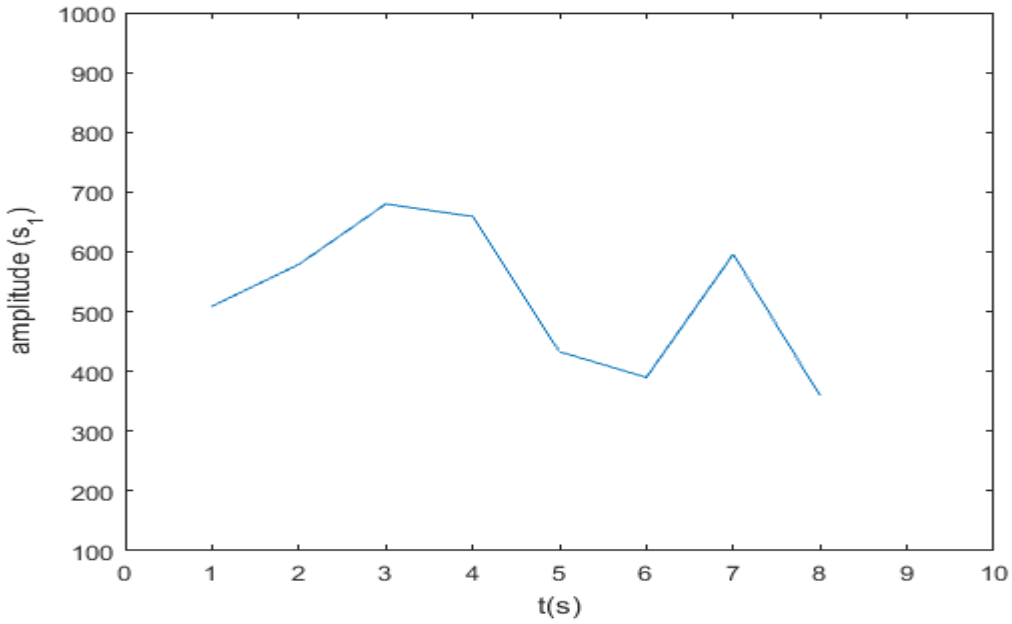

Figure 4.47: MATLAB® representation of measured pressure pulse at sensor 1 using the wireless communication device for a 1bar pressure reading

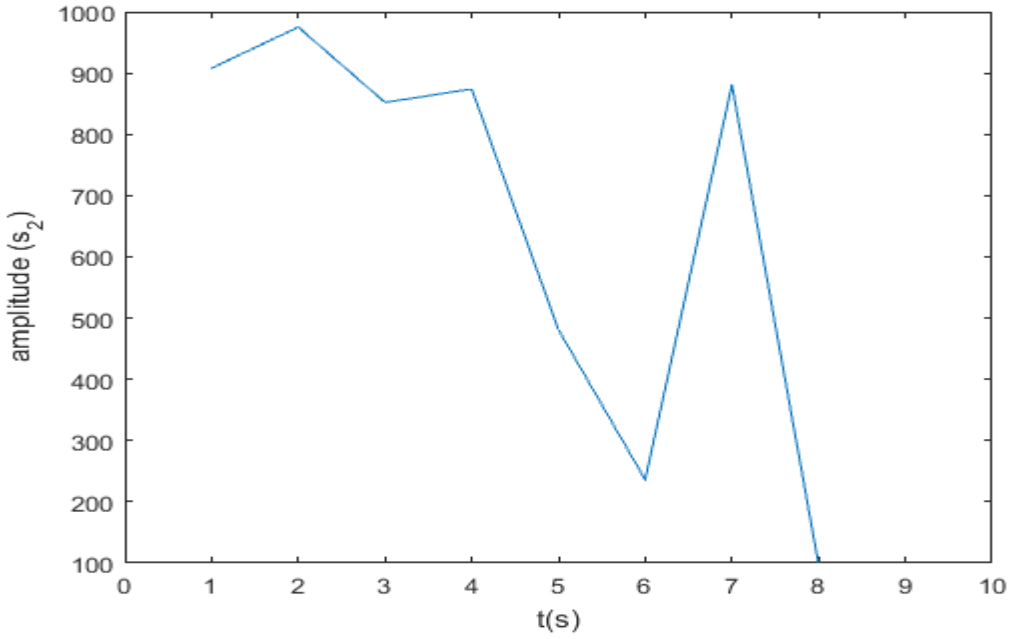

Figure 4.48: MATLAB® representation of measured pressure pulse at sensor 2 using the wireless communication device for a 1bar pressure reading

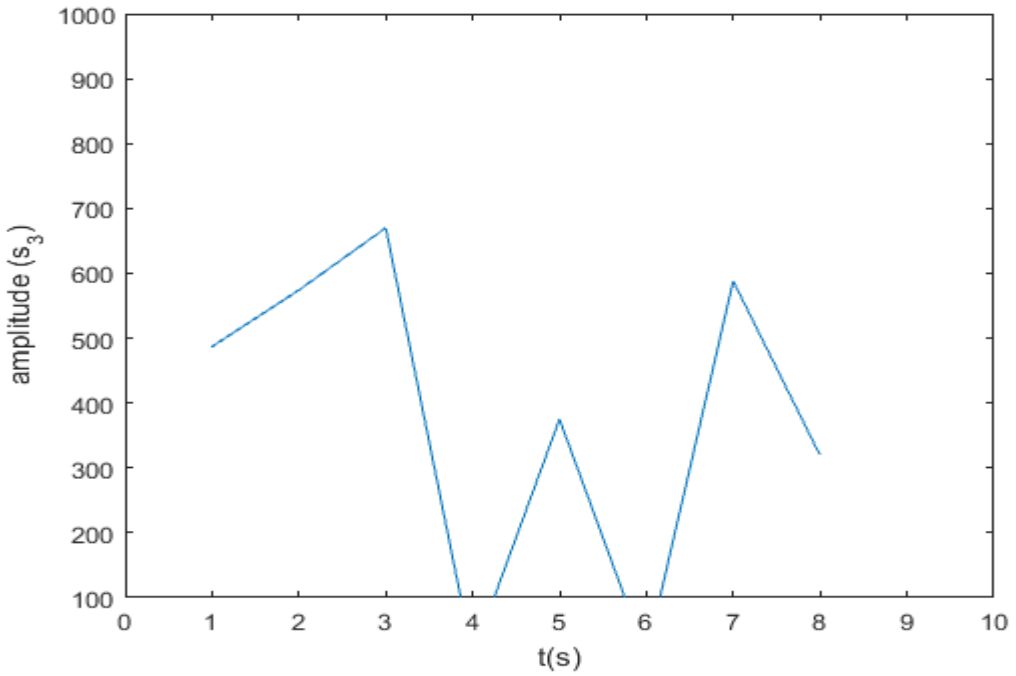

Figure 4.49: MATLAB® representation of measured pressure pulse at sensor 3 using the wireless communication device for a 1bar pressure reading

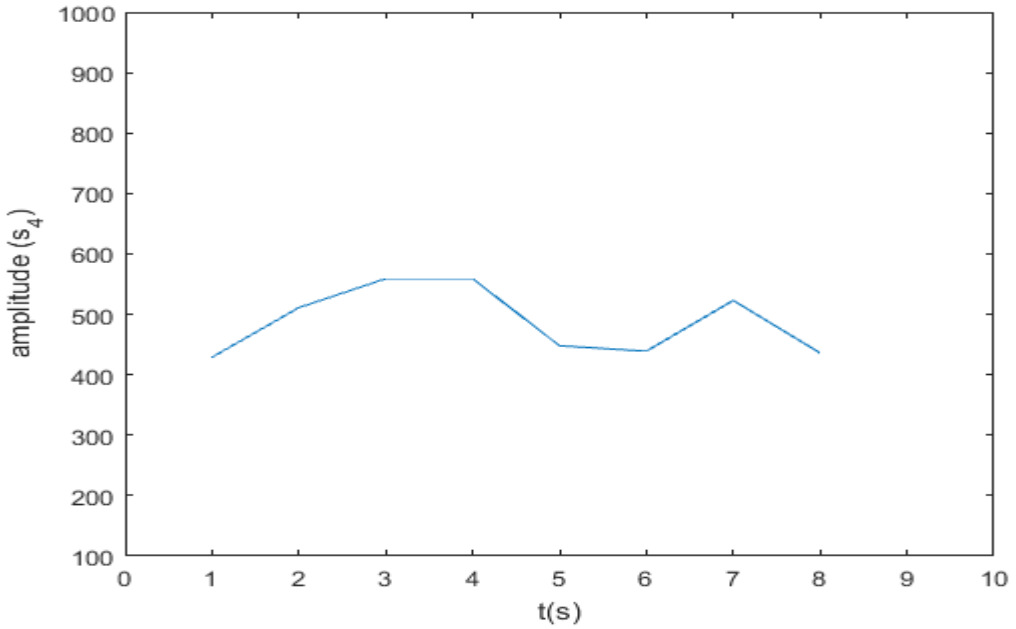

Figure 4.50: MATLAB® representation of measured pressure pulse at sensor 4 using the wireless communication device for a 1bar pressure reading

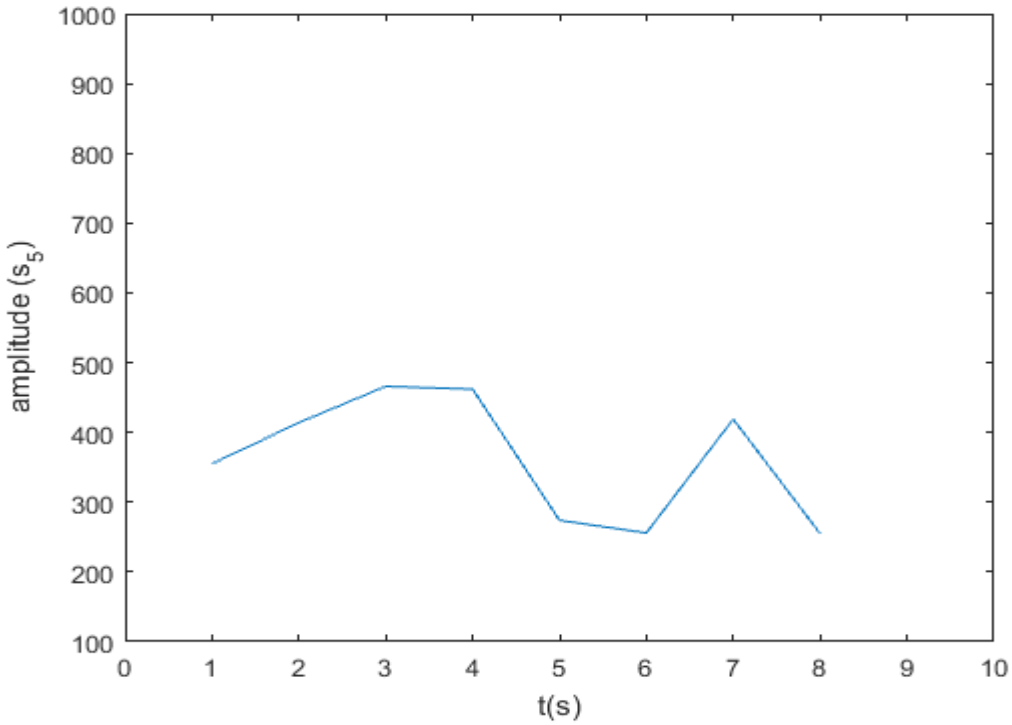

Figure 4.51: MATLAB® representation of measured pressure pulse at sensor 5 using the wireless communication device for a 1bar pressure reading

Based on the sensor locations on the experimental set, a pressure pulse from the pulse generator would get to sensor 2 first, then to sensors 1, 3, 4, and 5 respectively. This was as a result of their respective distances from the pulse generator. When a pressure of 0.8 bar was used, the pulse in Figure 4.43 has an amplitude value of 901, which is the highest value for all the five sensors confirming that the pulse got to sensor 2 first. Figure 4.42 shows a pulse with an amplitude value of 499, the second highest value for all five sensors confirming that the pulse got to sensor 1 after sensor 2. Figure 4.44 shows a pulse with an amplitude value of 477; Figure 4.45 shows a pulse with an amplitude value of 420; Figure 4.46 shows a pulse with an amplitude value of 346. This confirms that the pressure pulse eventually got to sensors 3, 4, 5in that order respectively.

For the case of pressure of 1.0 bar, the pulse in Figure 4.48 has an amplitude value of 908, which is the highest value for all the five sensors confirming that the pulse also got to sensor 2 first. Figure 4.47 shows a pulse with an amplitude value of 509, the second highest value for all five sensors confirming that the pulse also got to sensor 1 after sensor 2 in this case too. Figure 4.49 shows a pulse with an amplitude value of 487; Figure 4.50 shows a pulse with an amplitude value of 429; Figure 4.51 shows a pulse with an amplitude value of 355. This also goes to confirm that the pressure pulse eventually got to sensors 3, 4, 5 in that order respectively. In general, the results of these experiments confirm the aforementioned and also the effectiveness of the wireless communication device

A total of 15 tests were repeated using five different pressure readings on the pulse generator and the wireless communication device. Figure 4.52 is a single representation of all the 5 pressure pulses. Plate VI and Plate VII show the screen display of the ThinsSpeak platform.

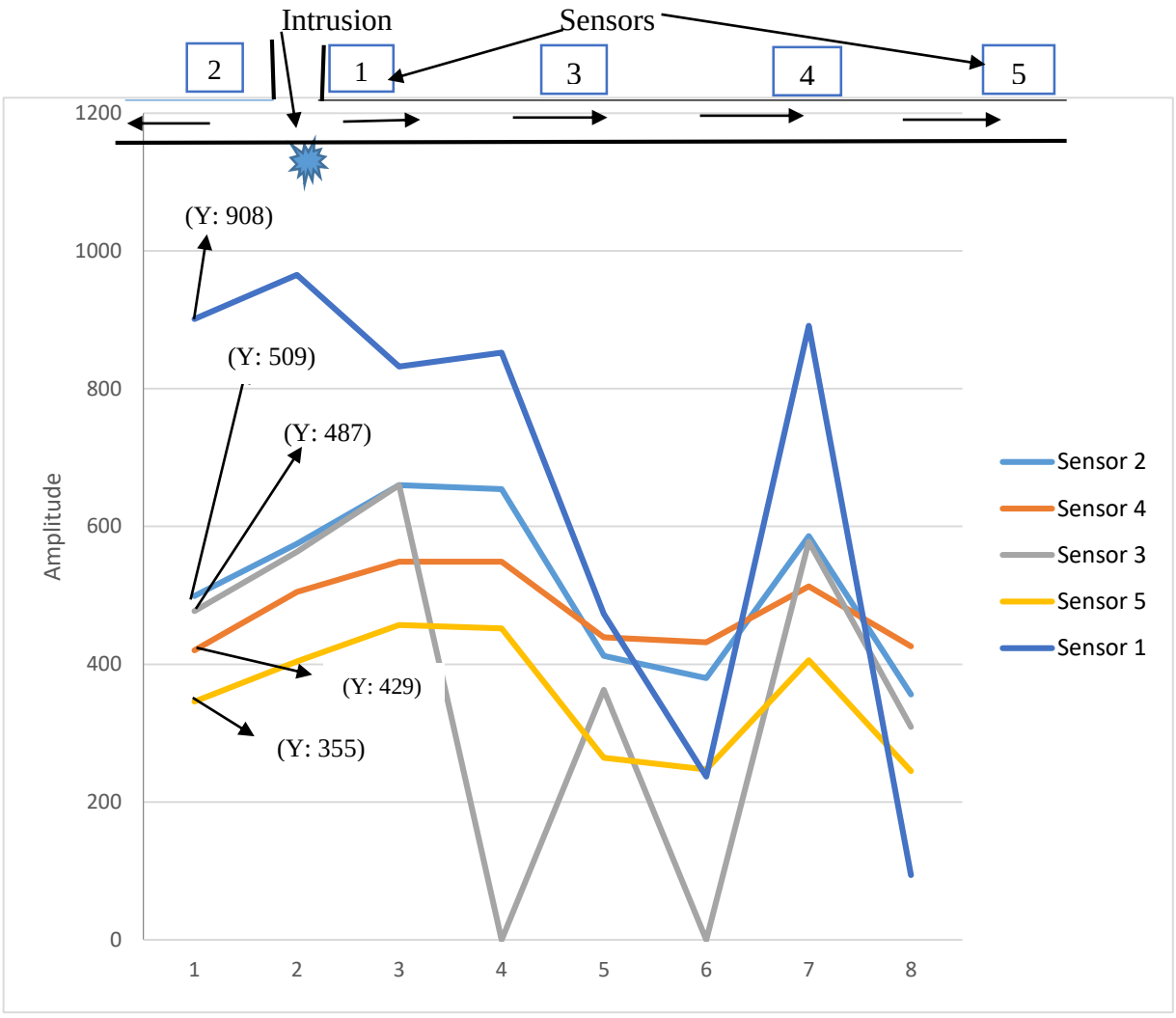

Figure 4.52: ThingSpeak analytics platform page showing measured pressure pulses from all sensors for a 1 bar pressure reading

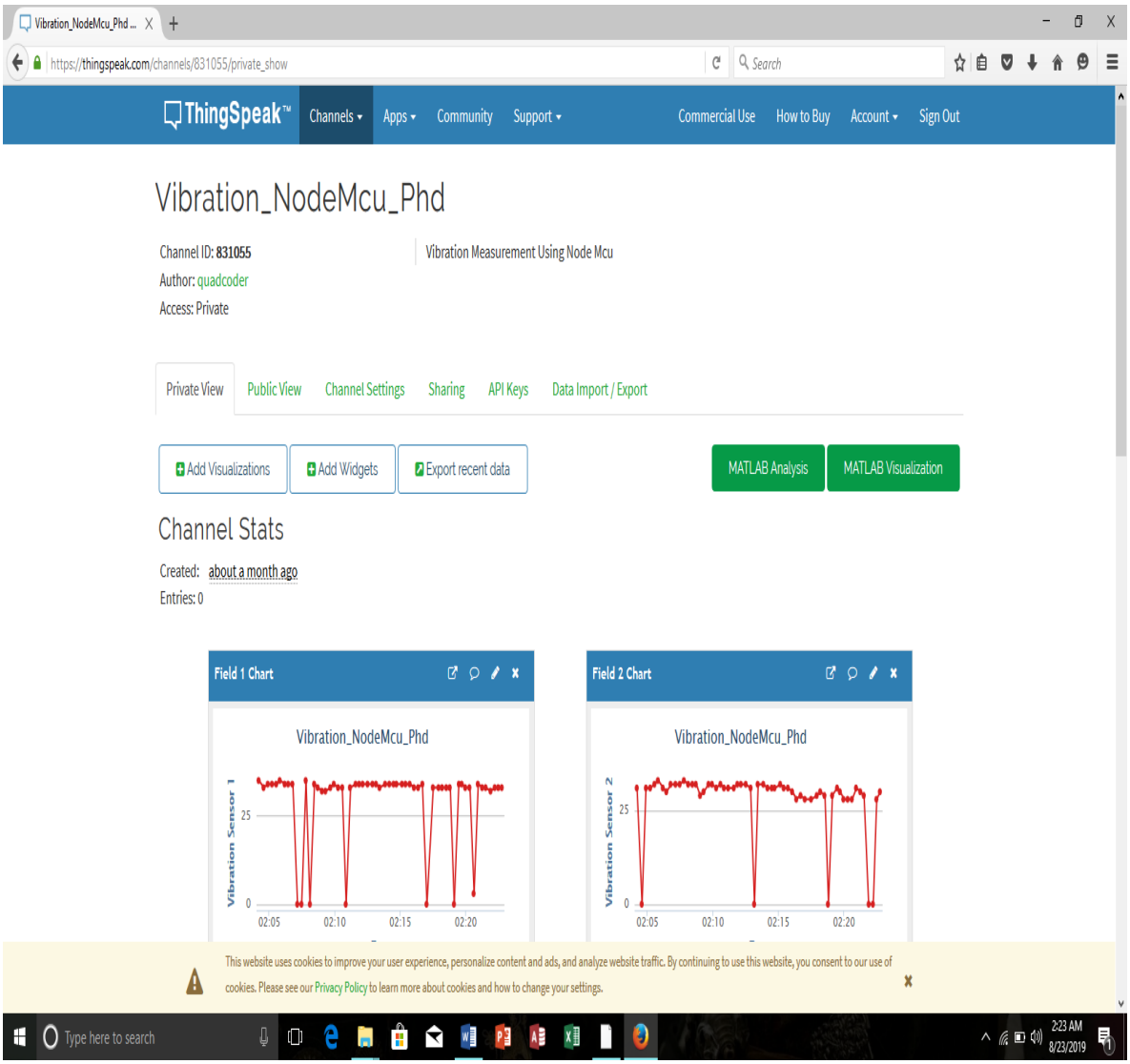

Plate VI: ThingSpeak analytics platform page showing measured pressure pulses from sensor channels 1 and 2 (ThinsSpeak, 2019)

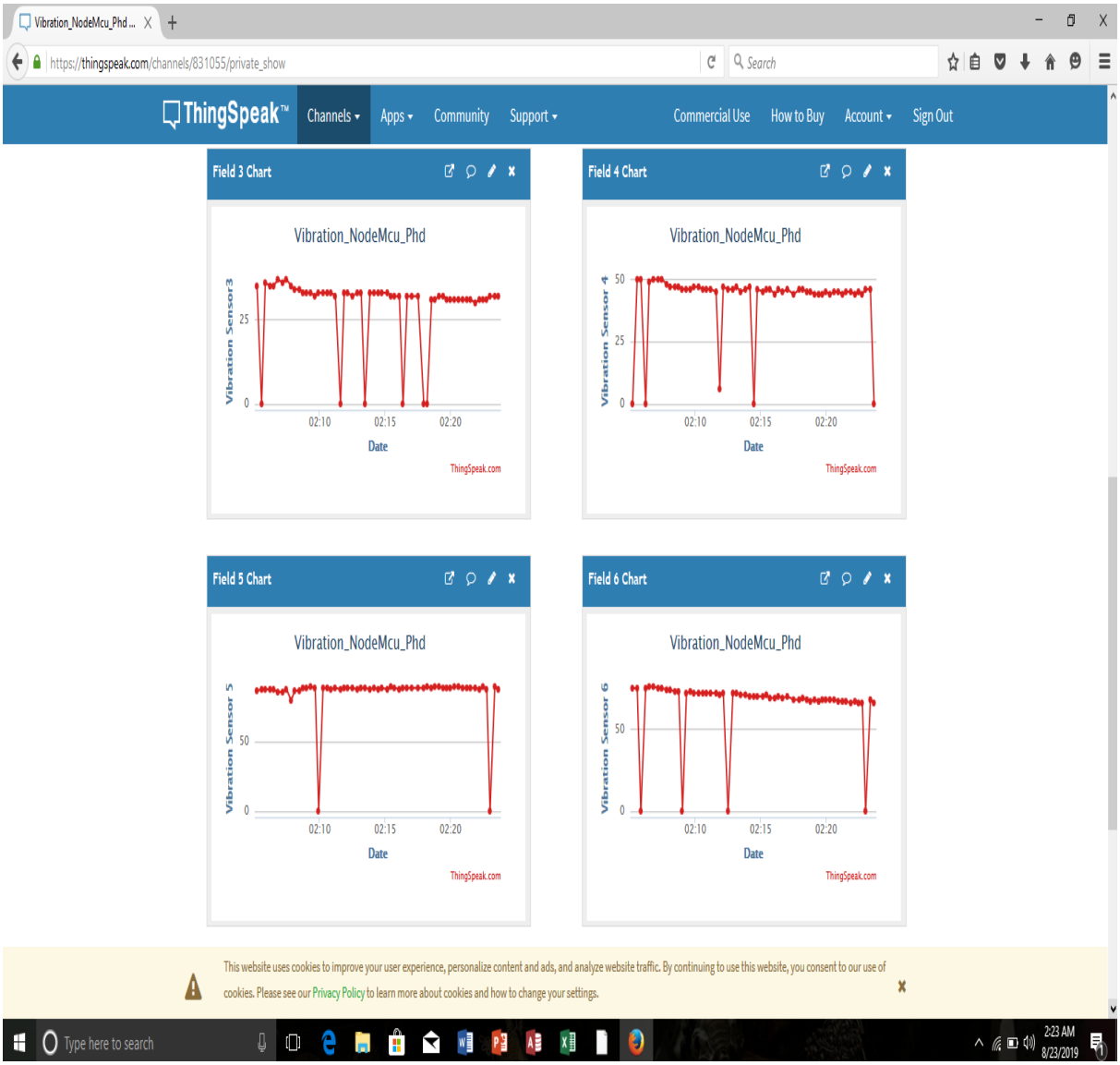

Plate VII: ThingSpeak analytics platform page showing measured pressure pulses from sensor channels 3, 4, and 5 (ThinsSpeak, 2019)

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Impacts and intrusions are among the major causes of pipeline damage. These events cause vibrations in the source point that can be used to detect, locate and characterise the defect. In this work, various mathematical techniques; algorithms, numerical experiments and a wireless communication device for real time monitoring of pipelines have been presented, developed and discussed. This system was tested and the following have so far been established in the course of this research work:

1. Mathematical methods to carry out damage detection, location and characterisation in a pipe were developed
2. The developed algorithms were first assessed by simulating them in Matlab® using data obtained from other works. The shapes and magnitude of the measured original pulses and reconstructed pulses agreed quite well confirming that the workability of the developed model. A difference of 0.01 m only was obtained between the pulse magnitudes of both the original and reconstructed pulses.
3. The simulation was verified using an experimental test rig and results obtained showed were similar to those obtained from the simulation. These experimental results are consistent with the model results which gave a similar level of estimation of the pulse amplitude.
4. A wireless communication device was developed for transmission and processing of measured pressure pulses wirelessly to an analytics platform (ThingSpeak) for real time monitoring.

5.2 Recommendations

For future research, a lot of problems abound that should be focused upon. For the continuation of this thesis work, the directions that should be prioritised are listed here.

i. The determination of the size of the burst or damage was not included in this work. In order to determine the size of a burst on a pipe, the method of transient damping should be used. This is because the pressure pulse information is used more by this method.

i. Transmission of measured pulse data via the wireless communication device was done at the rate of once every 15 seconds. This made carrying out a lot of tests very difficult due to time constraint. Future work should be carried out with the professional license of ThingSpeak to facilitate data transmission every second.

ii. Wireless sensors should be used in future experimental setups to mimic real life scenarios as much as possible.

5.3 Contributions to Knowledge

The research work has been able to contribute to knowledge in the following ways:

i. A low-cost monitoring system with the ability to perform real-time damage detection, location and characterisation, and allows you to view the results of the measured pressure pulses in real time on a computer system or even on a smart phone from any location in the world was developed.

ii. Reconstruction of the original pulse at an event location on a pipe from the combination of pressure pulses measured from two other locations along the same pipe was achieved.

iii. Damage location in a water-filled pipe using the technique of delay in arrival times of pressure pulses was achieved.

iv. Characterisation and classification of different damage events on a pipe was also achieved.

REFERENCES

Abraham, D. A. (2017). Signal Processing. Applied Underwater Acoustics. *Elsevier*. 743-807. doi:10.1016/B978-0-12-811240-3.00011-4

A.D.A.M. Consumer Health Inc. (2017). Retrieved from www.adam.com

Adnan, N. F., Ghazali, M. F., Amin, M. M., Hamat, A. M. A. (2015). Leak Detection in Gas Pipelines by Acoustic and Signal Processing-a Review. *3rd International Conference of Mechanical Engineering Research. IOP Conference Series: Material Science and Engineering.* doi:10.1088/1757-899X/100/1/012013.

Agbaeze, K.N. (2000). Petroleum Pipeline Leakages. *PPMC Report for Chief Officers Mandatory Course 026, Lagos*, 5, 25-28.

Agbakwuru, J. (2011). Pipeline Potential Leak Detection Technologies: Assessment and Perspective in the Nigeria Niger Delta Region. *Journal of Environmental Protection*, 2, 1055-1061.

Agrawal, A. (2008). A Theoretical, Numerical and Experimental Investigation of Guided Wave Propagation in Hollow Cylinders. Master Thesis, Carnegie Mellon University.

Amnesty International, (2015). Nigeria: Clean it up: Shell's false claims about oil spill response in the Niger Delta. *Amnesty*, 12, 15-18.

Ayn, J. (2019). Internet of Things in the oil and gas industry – current applications. *Oil and Gas*, 3, 5-7.

Blackledge J.M. (2006). Digital Signal Processing. Dublin Institute of Technology, School of Electrical and Electronic Engineering. ARROW@DIT.

Brunone, B. and Ferrante, M. (2001). "Detecting Leaks in Pressurised Pipes by Means of Transients." *Journal of Hydraulic Research*, IAHR 39(5), 539–547.

Burgess, R. C. (2014). Filters, Analog/Digital. Encyclopedia of the Neurological Sciences (Second Edition). *Elsevier*, 229-307. doi:10.1016/B978-0-12-385157-4.00530-3

Changhang, X., Jing, X., Wuyang, Z., Qingzhao K., Guoming, C., Gangbing, S. (2017). Experimental Investigation on the Detection of Multiple Surface Cracks Using

Vibrothermography with a Low-Power Piezoceramic Actuator. *Sensors*, 17, 27-35. doi:10.3390/s17122705

Cheddadi, Y., Cheddadi, H., Cheddadi, F., Errahimi, F., Es‐sbai1, N. (2020). Design and implementation of an intelligent low‐cost IoT solution for energy monitoring of photovoltaic stations. *SN Applied Sciences*, 2, 11-16. doi:10.1007/s42452-020-2997-4

Chung, H., Bergman, J., Li, F., Zhang, D., Janapati,V., and Cheung, C. (2015). Real-Time Active Pipeline Integrity Detection System for Gas Pipeline Safety Monitoring. *Energy Research and Development*, 4, 45-53.

CIA World Factbook, (2019). The World Factbook, Field Listing: Pipelines. 9, 987-991. Skyhorse Publishers. ISBN-10:1510758259.

DOSITS, (2019). Discovery of Sound in the Sea. Retrieved from https://dosits.org

Enrique, S., Rui, S., Ricardo, P. (2016). Damage Detection Based on Power Dissipation Measured with PZT Sensors through the Combination of Electro-Mechanical Impedances and Guided Waves. *Sensors*, 16, 639. doi:10.3390/s16050639.

Epoxy Oilserv, (2019). Pipeline leak detection industry for the oil and gas will grow say industry experts. *Epoxyoil*, 10, 15-16.

Ferrante, M. and Brunone, B. (2013). Pipe System Diagnosis and Leak Detection by Unsteady-State Tests: Harmonic Analysis. *Advances in Water Resources*, 26, 95–105.

Finnemore, E. J., Franzini, J. B. (2002). Fluid Mechanics with Engineering Applications. 10 editions, Boston, McGraw-Hill. ISBN 0-07-243202-0; 0-07- 112196-X.

Gamma, (2019). Gamma Ukraine. Retrieved from www.microchip.ua

Giurgiutiu, V. (2014). Structural Health Monitoring with Piezoelectric Wafer Active Sensors. Second Edition. Academic Press, 807-862.

Gold, J. (2019). How the oil and gas industry exploits IoT. Network World. Retrieved from https://www.networkworld.com/article/3445204/how-the-oil-and-gas-industry-exploits-iot.html

Golmohamadi, M. (2015). Pipeline Leak Detection. Masters Theses, Missouri University of Science and Technology.

Goswami, J. C., Chan, A. K. (1999). Fundamentals of Wavelets: Theory, Algorithms, and Applications. New York, Wiley. ISBN 0-471-19748-3.

Guan, L., Du, Y., Li, L. (2004). Wavelets in Petroleum Industry: Past, Present and Future. SPE 89952. *SPE Annual Technical Conference and Exhibition, Houston, Texas, U. S. A.*, 5, 12-15.

Guofeng, D., Qingzhao, K., Hua, Z., Haichang, G. (2017). Multiple Cracks Detection in Pipeline Using Damage Index Matrix Based on Piezoceramic Transducer-Enabled Stress Wave Propagation. *Sensors*, 17, 1812. doi:10.3390/s17081812.

Hale, J.M. and Olugboji, O. A. (2011). Damage Detection in Gas Pipelines by Remote Impact Measurement. *Journal AU*, 7, 11-15.

Haseloh, P. and LaFleur, D. (2013). Pipeline Leak Detection System. *Patents*, 7, 13-17.

Henrie, M., Carpenter, P., Nicholas, R. E. (2016). Pipeline Leak Detection Handbook. First Edition. Gulf Professional Publishing, 57-89.

Hill, S. (2019). Operations & integrity management and compliance in an age of IIOT. *Pipeline Technology Journal*, 3, 7-11.

Hobby Electronics, (2019). UK electronic components supplier. Retrieved from www.hobbytroics.co.uk

Ibrahim, H., Mostafa, N., Halawa, H., Elsalamouny, M., Daoud, R., Amer, H., Adel, Y., Shaarawi, A., Khattab, A., ElSayed, H. (2019). A layered IoT architecture for greenhouse monitoring and remote control. *SN Applied Sciences*, 1, 223-226. doi:10.1007/s42452-019-0227-8

Jihoon, C., Joonho, S., Choonggeun, S., Suyong, H., Doo. I. P. (2017). Leak Detection and Location of Water Pipes using Vibration Sensors and Modified ML Pre-filter. *Sensors*, 17, 2104. doi:10.3390/s17092104

Joshi, N. (2019). *Refining the oil and gas industry with IoT*. Forbes. Google books, New Jersey.

Junxiao, Z., Liang, R., Siu-Chun, H., Ziguang, J., Gangbing, S. (2017). Gas Pipeline Leakage Detection Based on PZT Sensors. *Smart Materials and Structures,* 26, 1-7. doi:10.1088/1361-665X/26/2/025022

Kia, X., Qingshan, D., Lujun, C., Siuchun, H., Gangbing, S. (2018). Damage Detection of a Concrete Column Subject to Blast Loads Using Embedded Piezoceramic Transducers. *Sensors,* 18, 1377. doi:10.3390/s18051377

Lee, P., Vítkovský, J., Lambert, M., Simpson, A. and Liggett, J. (2003). Frequency Response Coding for the Location of Leaks in Single Pipeline Systems. *ASCE,* 23, 25-30.

Mclintyre, J. T. (2017). Fluid Pipeline Leak Detection and Location with Miniature RF Tags. *Patents,* 15, 42-47.

Misiunas, D. (2004). Burst Detection and Location in Pipelines and Pipe Networks with Application in Water Distribution Systems. Ph.D. Thesis, Lund University, Lund, Sweden.

Motaz, D., and Yousef-Awwad, D. (2017). Smart Water Leakage Detection Using Wireless Sensor Networks (SWLD). *International Journal of Networks and Communications*, 7(1), 1-16. doi:10.5923/j.ijnc.20170701.01

Nandi, A. K., and Wong, D. M. L. (2014). Intelligent Vibration Signal Processing for Condition Monitoring. *Surveillance,* 5, 21-26.

Namuq, M. A. (2013). Simulation and Modelling of Pressure Pulse Propagation in Fluids inside Drill Strings. PhD Thesis, Faculty of Geosciences, Geo-engineering and Mining, Institute of Drilling Engineering and Fluid Mining.

NASA (2018). Speed of Sound. Retrieved from https://www.grc.nasa.gov/www/k-12/airplane/sound.html

Ninduwezuor-Ehiobu, N. (2017). *Opportunity Identification Report*. ZEST, University of Waterloo, Canada.

NNPC, (2007). Pipeline Oil Spill and Remediation in NDS. *NNPC Annual Reports*, 12, 51-54.

Oil and Gas IQ. (2015). Oil and Gas 2015 Predictions. What Lies in the Year Ahead. *Oil and Gas,* 5, 14-19.

Olugboji, O. A., Yisa, J. J. and Ajani, C. K. (2015). Problem of Calculating Time Delay between Pulse Arrivals. *American Journal of Engineering Research (AJER),* 4(4), 132-142.

Olugboji, O. A. (2011). Development of an Impact Monitoring System for Petroleum Pipelines. *Journal AU*, 15(2), 115-120.

Olugboji, O. A and Yisa, J. J. (2012). Event Reconstruction by Inverse Methods. *International Journal of Electronics; Mechanical and Mechatronics Engineering*, 4(3), 735 - 746.

Olugboji, O. A., Yisa, J. J., Hale, J.M. (2013). Event Reconstruction by Digital Filtering. *Advances in Signal Processing,* 1(3), 48-56. doi:10.13189/asp.2013.010303

Onuoha, F. (2009). Why the poor pay with their lives: Oil pipeline Vandalisation, Fires and Human Security in Nigeria. *Disaster,* 33(3), 369-389.

Ortiz, C. E. P., Aguiar, R. B., Pires, A. P. (2009). Wavelet Filtering of Permanent Downhole Gauge Data. SPE 123028. *Latin American and Caribbean Petroleum Engineering Conference held Cartagena, Colombia*, 1(2), 99-104.

Park, H. W., Sohn, H., Law, K .H. and Farrar, C. R., (2007). Time Reversal Active Sensing for Health Monitoring of a Composite Plate. *Journal of Sound and Vibration*. Vol. 302, No. 1-2, pp. 50-66.

Raj, P., Rajasree, Jayasari, Mittal, Y., Mittal, V. K. (2015). A Web-based Intelligent Spybot. *Sensors*, 5, 14.19. doi:10.1007/978-3-319-26832-3_44

Rajtar, J.M.; Muthiah, R. (1997). Pipeline leak detection system for oil and gas flow lines. *Journal of Manufacturing Science and Engineering*, 119(1), 105-109. doi:10.1115/1.2836545

Rashid, S., Qaisar, S., Saeed, H. and Felemban, H. (2014). A Method for Distributed Pipeline Burst and Leakage Detection in Wireless Sensor Networks Using Transform Analysis. *International Journal of Distributed Sensor Networks*, 5(3), 26-31. doi:10.1155/2014/939657

Ravi, S.; Karthikraj, S.; Saareesan, D.; Kishore, R. (2016). Pipeline Monitoring using Vibroacoustic Sensing – A Review. *International Research Journal of Engineering and Technology*, 3(1), 1-5.

Razi, P.; Esmaeel, R.A.; Taheri, F. (2013). Improvement of a Vibration-based Damage Detection Approach for Health Monitoring of Bolted Flange Joints in Pipelines. *Structural Health Monitoring*, 12, 207–224.

Seppanen R.K. (2016). Translational Research on Oral and Maxillofacial Sciences. *Hammas*, 3, 9-14.

Shell in Nigeria, (2019). Security, Theft, Sabotage and Spills. *Shell Annual Reports and Publications*, 10, 18-21.

Soliman, M. Y., Ansah, J., Stephenson, S., Mandal, B. (2001). Application of Wavelet Transform to Analysis of Pressure Transient Data. *SPE Annual Technical Conference and Exhibition, New Orleans, LA*, 3, 56-60.

ThingSpeak, (2019). ThingSpeak Analytics Platform. Retrieved from www.thingspeak.com

Tomiwa, P., Ayeni, B., Aina, A. (2020). Intelligent pipeline monitoring system based on internet of things. *Scientific Research Journal*, 4(3), 20-26. doi:10.31364/SCIRJ/v8.i8.2020.P0820793

Wang, X. J., Lambert, M. F., Simpson, A. R. and Vitkovsky, J. P. (2001). Leak Detection in Pipeline Systems and Networks: A Review. *The Institution of Engineers, Australia Conference on Hydraulics in Civil Engineering, Hobart*, 10, 11-16.

Wang, X., Lambert, M., Simpson, A., Liggett, J. and Vítkovský, J. (2002). Leak Detection in Pipeline Systems Using the Damping of Fluid Transients. *Journal of Hydraulic Engineering*, 128(7), 697–711.

Wolf, M. (2019). Embedded System Interfacing: Design for the Internet-of-Things (IoT) and Cyber-Physical System (CPS). First Edition. *Morgan Kaufmann*, 93-128. doi:10.1016/B978-0-12-817402-9.00005-4

Záruba, J. (1993). Water Hammer in Pipe-line Systems. *Elsevier*, 43, 57-59.

Zisselman, E., Adler, A., Elad, M. (2018). *Compressed Learning for Image Classification: A Deep Neural Network Approach*. Handbook of Numerical Analysis: Processing, Analysing and Learning of Images, Shapes and Forms, 1(19), 3-17. doi:10.1016/bs.hna.2018.08.002

APPENDICES

Appendix A

MATLAB® Code for Sampling and Reconstruction

```
%plot of original pressure pulses s1, s2, s3, s4, s5, s6

t  = Time;

s1 = ChannelCh1;

s2 = ChannelCh2;

s3 = ChannelCh3;

s4 = ChannelCh4;

s5 = ChannelCh5;

s6 = ChannelCh6;


plot(t*0.001,s1)

xlim([0 0.05])

xlabel('time(s)')

ylabel('s_1')


plot(t*0.001,s2)

xlim([-0.01 0.05])

xlabel('time(s)')

ylabel('s_2')


plot(t*0.001,s3)

xlim([0 0.05])
```

```
xlabel('time(s)')

ylabel('s_3')


plot(t*0.001,s4)

xlim([0 0.05])

xlabel('time(s)')

ylabel('s_4')


plot(t*0.001,s5)

xlim([0 0.05])

xlabel('time(s)')

ylabel('s_5')


plot(t*0.001,s6)

xlim([0 0.05])

xlabel('time(s)')

ylabel('s_6')


%To satisfy Nyquist criterion

Ts = 0.000076;              %sampling period

Fs = 26.32;                 %sampling frequency

n = 1:1:1320;               %number of samples

N = length(n);              %sample length
```

```
%sampled signal
%sampling of pressure pulse s2
s11_samples = s1(1:1:1320);    %1320 samples of s1.
stem(n,s11_samples)
xlim([0 300])
xlabel('n')
ylabel('Discrete time signal s11(n)')

%sampled signal
%sampling of pressure pulse s3
s33_samples = s3(1:1:1320);    %1320 samples of s3.
stem(n,s33_samples)
xlim([0 300])
xlabel('n')
ylabel('Discrete time signal s33(n)')

%reconstruction of original signal s2
%recostruction factor k = 2.235
s2=zeros(N,length(t));
for i=1:N
   s2(i,:)=(s11_samples(i)*s33_samples(i))*rectpuls(Fs*t-i+1);
end
```

```
plot(t,sum(s2),'b-',t,s1,'r:',t,s3,'r:')

grid;

xlim([0 40])

legend('Reconstructed Signal','Original Signal')

ylabel('Reconstructed Signal s2(t1)')

xlabel('time(ms)')
```

Appendix B

Code for Programming of Arduino

```
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
SoftwareSerial s(5,6);

void setup() {
  s.begin(115200);
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT);
  pinMode(A3,INPUT);
  pinMode(A4,INPUT);
  pinMode(A5,INPUT);

}
StaticJsonBuffer<1000> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();
void loop() {

  // If the DHT-11 is not connected to correct pin or if it doesnot
//work no data will be sent


  root["Sensor1"]= analogRead(A0);
```

```
  root["Sensor2"]= analogRead(A1);
   root["Sensor3"]= analogRead(A2);
   root["Sensor4"]= analogRead(A3);
   root["Sensor5"]= analogRead(A4);
   root["Sensor6"]= analogRead(A5);


if(s.available()>0)
{
 root.printTo(s);
 delay(1000);
}
}
```

Appendix C

Code for Programming of ESP8266 Module

```
/*
  WriteMultipleFields

  Description: Writes values to fields 1,2,3,4 and status in a single ThingSpeak update every 20 seconds.

  Hardware: ESP8266 based boards

  !!! IMPORTANT - Modify the secrets.h file for this project with your network connection and ThingSpeak channel details. !!!

  Note:
  - Requires ESP8266WiFi library and ESP8622 board add-on. See https://github.com/esp8266/Arduino for details.
  - Select the target hardware from the Tools->Board menu
  - This example is written for a network using WPA encryption. For WEP or WPA, change the WiFi.begin() call accordingly.

ThingSpeak (https://www.thingspeak.com ) is an analytic IoT platform service that allows you to aggregate, visualise, and
analyse live data streams in the cloud. Visit https://www.thingspeak.com to sign up for a free account and create a channel.

Documentation for the ThingSpeak Communication Library for Arduino is in the README.md folder where the library was installed.
```

See https://www.mathworks.com/help/thingspeak/index.html for the full ThingSpeak documentation.

For licensing information, see the accompanying license file.

*/

```
#include <SoftwareSerial.h>
SoftwareSerial s(6,5);
#include <ArduinoJson.h>
#include "ThingSpeak.h"
#include "secrets.h"
#include <ESP8266WiFi.h>

char ssid[] = SECRET_SSID;   // your network SSID (name)
char pass[] = SECRET_PASS;   // your network password
int keyIndex = 0;            // your network key Index data (needed only for WEP)
WiFiClient  client;

unsigned long myChannelData = SECRET_CH_ID;
const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

// Initialise our values

void setup() {
```

```
  Serial.begin(115200);  // Initialise serial
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);  // Initialise ThingSpeak
  s.begin(115200);
  while (!Serial) continue;
}
void loop() {
StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& root = jsonBuffer.parseObject(s);
  if (root == JsonObject::invalid())
    return;

  Serial.println("JSON received and parsed");
  root.prettyPrintTo(Serial);
  Serial.print("Data 1 ");
  Serial.println("");
  int data1=root["Sensor1"];
  Serial.print(data1);
    Serial.println("");
 Serial.print("   Data 2 ");
  int data2=root["Sensor2"];
  Serial.print(data2);
  Serial.println("");
  int data3=root["Sensor3"];
  Serial.print(data3);
```

```
  Serial.println("");
  int data4=root["Sensor4"];
  Serial.print(data4);
  Serial.println("");
  int data5=root["Sensor5"];
  Serial.print(data5);
  Serial.println("");
  int data6=root["Sensor6"];
  Serial.print(data6);
  Serial.println("");
  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }
  // set the fields with the values
  ThingSpeak.setField(1, data1);
  ThingSpeak.setField(2, data2);
```

```
 ThingSpeak.setField(3, data3);
 ThingSpeak.setField(4, data4);
  ThingSpeak.setField(5, data5);
    ThingSpeak.setField(6, data6);
 // figure out the status message
 // write to the ThingSpeak channel
 int x = ThingSpeak.writeFields(myChanneldata, myWriteAPIKey);
 if(x == 200){
  Serial.println("Channel update successful.");
 }
 else{
  Serial.println("Problem updating channel. HTTP error code " + String(x));
 }
 // change the values
 delay(6000); // Wait 20 seconds to update the channel again
}
```