



Modelling CO₂ Corrosion of Pipeline Steels

Muhammad Hashim Abbas

A thesis submitted to the
School of Marine Science and Technology
Faculty of Science, Agriculture and Engineering,
Newcastle University
in fulfilment of the requirements
for the degree of
Doctor of Philosophy

November, 2016

Abstract

Over the years, several attempts have been made by various research institutions and petroleum companies to develop models for the prediction of CO₂ corrosion in pipelines, in order to better capture the underlying principles that cause it. Modelling CO₂ corrosion is important to the oil and gas and carbon capture and storage (CCS) industries, as it provides the means by which the prevention of the financial costs from lost production, the preservation of the environment as well as the health and safety of human lives can be achieved.

In this thesis, existing models have been investigated and compared against newly derived models in terms of their accuracy of prediction, by using an identical test dataset. A neural network (NN) model was developed, in which a detailed sensitivity analysis was carried out on Matlab training functions to determine their degree of suitability in CO₂ corrosion prediction. Results showed that the tansig transfer function was the most suitable and that a 2-layer network was sufficient to obtain desirable R^2 -values of ~ 0.9 for both low and high pressure CO₂ corrosion data.

Also, a linear regression model was developed based on predictor variables: temperature (T), CO₂ partial pressure (P_{CO_2}), fluid velocity (U) and pH, for both low and high pressure CO₂ data. The respective R^2 -values obtained are 0.65 and 0.7. An R^2 -value of 0.8 can be achieved for the low pressure CO₂ data; however the derived regression equation is inelegant and contains a combination of a large number of predictor terms.

From Monte Carlo analyses, the exponential and normal distributions were discovered to be the best fits for the low and high pressure CO₂ corrosion rate data, respectively. Further, parametric sensitivity analyses revealed the pH and fluid velocity to be the least and most significant variables for low pressure CO₂, respectively, while the velocity and temperature were the least and most significant variables for high pressure CO₂ corrosion, respectively.

Dedication

I dedicate this thesis to my wonderful family – my mother, father and sister. I am very grateful for all the words of advice, encouragement, support and prayers you have given me over the years. Writing this thesis would indeed not have been possible if not for the help you've all given me. Thank you so much for everything.

Table of contents

Abstract.....	i
Dedication.....	ii
List of Tables.....	viii
List of Figures.....	x
Nomenclature.....	xix
Greek letters.....	xxii
Chapter 1. Introduction.....	1
1.1 Corrosion Costs.....	2
1.2 Environmental Impact of CO ₂ Emissions.....	3
1.3 The Importance of CCS.....	8
1.4 CO ₂ Corrosion in Petroleum and CCS pipelines.....	9
1.5 Aims and Objectives.....	10
1.6 Thesis Structure and Content.....	13
Chapter 2. Literature Review on CO ₂ Corrosion.....	15
2.1 CO ₂ Corrosion - Introduction.....	16
2.2 An Outline of the CO ₂ Corrosion Process.....	16
2.3 Types of CO ₂ Corrosion Damage.....	19
2.3.1 Pitting.....	19
2.3.2 Mesa-type Attack.....	20
2.3.3 Flow-induced Localised Corrosion.....	21
2.4 Major Factors Influencing CO ₂ Corrosion.....	22
2.4.1 The Effect of Environmental Factors on CO ₂ Corrosion.....	22
Effect of Solution Chemistry.....	22
Effect of Temperature and CO ₂ Partial Pressure and Fluid Flowrate Velocities.....	23
CO ₂ Corrosion Prediction – Corrosion Rate-Determining Parameters.....	24
Presence of H ₂ S.....	27
Determination of pH for a Mixed CO ₂ /H ₂ S-System.....	28
Conditions for H ₂ S Corrosion Risks and H ₂ S Corrosion Mechanism.....	29
Mechanics and Kinetics of Iron Sulphide Film/Scale Formation.....	31
Species Transport through Boundary Layers.....	32
2.4.2 The Effect of Physical Factors on CO ₂ Corrosion.....	34
Effect of Crude oil.....	34
Effect of Flow and Erosion.....	35
Effect of Organic Acid.....	36

2.4.3 The Effect of Metallurgical Factors on CO ₂ Corrosion	36
Further Metallurgical Considerations - Alloying Elements.....	37
The Influence of Chromium	38
2.5 Established CO ₂ Corrosion Models	39
2.5.1 Hydrocor Model	40
2.5.2 Corplus Model.....	40
2.5.3 KSC Model.....	41
2.5.4 Multicorp Model.....	41
2.6 CO ₂ Corrosion Mechanism between Low and High Partial Pressures	42
2.6.1 Carbon Capture and Storage (CCS)	42
2.6.2 Research Directions.....	43
2.6.3 The Limitation of Henry's law for High Pressure Conditions	44
2.6.4 Methodology for CO ₂ Corrosion Prediction at High Pressure Conditions	45
2.7 Neural Networks	47
2.7.1 Case Study on the Implementation of Neural Networks in Model Prediction.....	49
2.8 Fuzzy Logic Systems	51
2.8.1 Case Study on the Execution of the Fuzzy Logic System.....	52
2.9 Conclusions.....	54
Chapter 3. CO ₂ Corrosion Models	57
3.1 Introduction – CO ₂ Corrosion Models	58
3.2 The 1975 de Waard-Milliams Correlation	59
3.3 The 1991 de Waard-Lotz-Milliams Correlation	61
3.3.1 Accounting for High-Temperature Protective Films.....	62
3.4 The 1995 de Waard-Lotz-Dugstad correlation	64
3.4.1 Parameter Study on the 1995 de Waard-Lotz-Dugstad Correlation – Investigating Changes in Flow Velocity and CO ₂ Fugacity on CO ₂ Corrosion Rate	66
3.5 The NORSOK M-506 Model.....	68
3.5.1 Matlab Model Results based on NORSOK Correlations	70
3.5.2 Parameter Study on the NORSOK M-506 Model – Investigating Changes in Water Cut Percentage and CO ₂ Mole Percentage on CO ₂ Corrosion Rate.....	71
3.6 The Nescic-Postlethwaite-Olsen Model (NPO Model)	72
3.6.1 Cathodic Reactions.....	73
3.6.2 Anodic Reactions	74
3.7 Introduction to Freecorp.....	76

3.7.1 Organic Acid Corrosion.....	76
3.7.2 Hydrogen Sulphide Corrosion	78
3.7.3 CO ₂ Corrosion Rate Trends and Parametric Study on Velocity of Fluid Flow	81
3.8 Conclusions	84
Chapter 4. Statistical Analysis.....	87
4.1 Introduction	88
4.2 Low Pressure CO ₂ Corrosion Data.....	88
4.2.1 Descriptive Statistics.....	88
4.2.2 Principal Component Analysis	91
4.2.3 Multiple Linear Regression	94
Variable Interactions	97
4.2.4 Nonlinear Regression.....	100
4.2.5 Response Surface Plots.....	102
4.3 High Pressure CO ₂ Corrosion Data	104
4.3.1 Descriptive Statistics.....	104
4.3.2 Principal Component Analysis	106
4.3.3 Multiple Linear Regression	108
Variable Interactions	110
4.3.4 Nonlinear Regression.....	114
4.3.5 Response Surface Plots.....	115
4.4 Conclusions	117
Chapter 5. Neural Network Models.....	119
5.1 Introduction – Historical Background	120
5.2 Definition of Neural Networks	122
5.3 The Simple Neuron and Neuron with Vector Input	123
5.4 The Transfer Function	124
5.5 Neural Network Architectures.....	125
5.5.1 Single-layer Feedforward Networks	125
5.5.2 Multilayered Feedforward Networks.....	125
5.5.3 Recurrent Networks	126
5.6 Design of the NN Model for Predicting CO ₂ Corrosion	127
5.6.1 NN Modelling of Low Pressure CO ₂ Corrosion Data	132
Summary of Training and Testing data for NN Modelling.....	132
Results	132

Discussion of the Final Neural Network Model Specifications	136
5.6.2 NN Modelling of High Pressure CO ₂ Corrosion Rate Data.....	138
Summary of Training and Testing data for NN Modelling	138
Results.....	141
Discussion of the Final NN Model Specifications	144
5.7 Conclusions.....	148
Chapter 6. Fuzzy Inference Systems	151
6.1 Introduction – Fuzzy Logic.....	152
6.2 Definition of Fuzzy Logic.....	153
6.3 Membership Functions.....	154
6.4 Mamdani and Sugeno Fuzzy Inference Systems	156
6.5 Low Pressure CO ₂ Corrosion Prediction using a Fuzzy Inference System	157
6.5.1 Methodology	157
6.6 High Pressure CO ₂ Corrosion Prediction using a Fuzzy Inference System.....	165
6.6.1 Methodology	165
6.7 Conclusions.....	172
Chapter 7. Monte Carlo Simulation	175
7.1 Introduction.....	176
7.2 Probability Distributions	177
7.3 Types of Probability Distributions.....	177
7.3.1 Discrete Probability Distributions	177
7.3.2 Continuous Probability Distributions.....	177
7.4 Monte Carlo Simulation of Low Pressure CO ₂ Corrosion Dataset.....	179
7.4.1 Experimental Dataset	179
7.4.2 Monte Carlo Dataset.....	182
Case Study I - Based on the Derived Regression Equation.....	185
Case Study II - Based on the Derived Regression Equation	187
Sensitivity Analysis	190
7.5 Monte Carlo Simulation of High Pressure CO ₂ Corrosion Dataset	191
7.5.1 Experimental Dataset	191
7.5.2 Monte Carlo Dataset.....	193
Case Study I - Based on the Derived Regression Equation.....	195
Case Study II - Based on the Derived Regression Equation	197
Sensitivity Analysis	200
7.6 Optimisation of Results.....	201

7.7 Conclusions	203
Chapter 8. Extended Discussion.....	207
8.1 Introduction	208
8.2 Model Performances of the 1991, 1995 De Waard, Norsok and Freecorp	208
8.2.1 Low Pressure CO ₂ Corrosion Prediction Results.....	208
The 1991 De Waard Model.....	208
The 1995 De Waard Model.....	209
Norsok Model.....	210
Freecorp Model	211
8.2.2 High Pressure CO ₂ Corrosion Prediction Results.....	212
The 1991 De Waard Model.....	212
The 1995 De Waard Model.....	213
Norsok Model.....	214
8.3 Applicability and Limitations of the Various Derived Models	215
8.4 Concluding Remarks on the Performances of each of the Models.....	218
Chapter 9. Conclusions and Recommendations for Further Work.....	221
9.1 Conclusions	222
9.2 Recommendations for Further Work.....	225
Appendices	226
A1.1 Project Gantt Chart	226
A1.2 Microsoft Excel Corrosion Database - Sample.....	227
A4.1 Statistical Student t-table	228
A4.2 Three-dimensional Plot of Corrosion Rate against CO ₂ Partial Pressure and Temperature for the Low Pressure CO ₂ Corrosion Dataset from Alternate Viewing Angles	229
A6.1a Three-dimensional Mamdani FIS Plot of Corrosion Rate against Temperature and CO ₂ partial pressure for the Low Pressure CO ₂ Dataset (Centroid Defuzzification)	230
A6.1b Three-dimensional Mamdani FIS Plot of Corrosion Rate against Temperature and CO ₂ partial pressure for the Low Pressure CO ₂ Dataset (Bisector Defuzzification).....	230
A6.2 Mamdani FIS Plot of Corrosion Rate against Temperature for the High Pressure CO ₂ Dataset	231
Developed Model Code	232
Regression Model Code	232
Neural Network Model Code	258
Monte Carlo Simulation Model Code	271
References	295

List of tables

Table 1-1. Share of corrosion costs for the U.S. industry categories (Koch et al., 2002).....	2
Table 1-2. RCP scenarios for the end of the 21 st century (IPCC, 2014).....	7
Table 1-3. Weight percentage chemical composition of St.52 carbon steel (Dugstad et al., 1994b)	10
Table 1-4. Weight percentage chemical composition of X65 carbon steel (Zhang et al., 2012a)	11
Table 2-1. Estimated corrosion rates of carbon-manganese steels for some fields in the Norwegian sector of the North Sea (Dugstad et al., 1994a)	25
Table 3-1. Summary of variables and their respective magnitudes required to obtain results in figure 3-7.....	70
Table 3-2. Scaling temperature maxima for the various models at $f_{CO_2}=0.3\text{MPa}$	84
Table 4-1. Descriptive statistics of the CO ₂ corrosion sample data.....	89
Table 4-2. Test data variables and their corresponding ranges.....	91
Table 4-3. Principal component analysis	91
Table 4-4. Regression model terms and statistics	94
Table 4-5. Descriptive statistics of the CO ₂ corrosion sample data.....	104
Table 4-6. Test data variables and their corresponding ranges.....	106
Table 4-7. Principal component analysis	106
Table 4-8. Regression model terms and statistics	109
Table 4-9 Low pressure CO ₂ corrosion model performance	118
Table 4-10 High pressure CO ₂ corrosion model performance.....	118
Table 5-1. Definitions of the transfer functions used (Beale et al., 2014).....	124
Table 5-2. Definitions of the training functions used (Beale et al., 2014; Sharma and Venugopalan, 2014).....	131
Table 5-4. Summary of the results obtained for the tested neuron configurations	136
Table 5-5. Summary of neural network properties	137
Table 5-6. Summary of the results obtained for the tested neuron configurations	145
Table 5-7. Summary of neural network properties	146
Table 5-8. Model performance for each dataset	149
Table 6-1. Summary of FIS processes and methods.....	157
Table 6-2. Membership function descriptor and corresponding corrosion rate ranges	159
Table 6-3. Some rules for the development of the Mamdani and Sugeno FIS.....	160
Table 6-4. Variables used in the study and their corresponding ranges	165

Table 6-5. Descriptors assigned to each of the input and output variable membership functions	166
Table 6-6. Rules for the development of Mamdani and Sugeno FIS	166
Table 6-7. Magnitudes of input variables and respective output variable results	167
Table 6-8. Model performance for each dataset	173
Table 7-1. Probability density functions for the normal and Weibull distributions (Harrison, 2010), (Johnson et al., 1994)	178
Table 7-2. Test statistic summary for normal and Weibull Probability density functions	180
Table 7-3. Probability density functions for the lognormal and exponential distributions (Harrison, 2010; Johnson et al., 1994)	181
Table 7-4. Statistical parameters for each distribution	182
Table 7-5. Summary statistics for the use of the Weibull distribution as a direct fit for the low pressure CO ₂ corrosion rate	184
Table 7-6 Summary statistics for the exponential and lognormal probability distributions ..	184
Table 7-7. Test statistic summary for normal and Weibull Probability density functions	185
Table 7-8. Summary statistics for Case Study I	187
Table 7-9. Best-fit distributions for each input parameter	187
Table 7-10. Test statistic summary for normal and Weibull probability density functions ...	188
Table 7-11. Summary statistics for Case Study II	190
Table 7-12. Test statistic summary for normal and Weibull probability density functions ...	192
Table 7-13. Statistical parameters for each distribution	193
Table 7-14 Summary statistics for the normal and logistic probability distributions	194
Table 7-15. Test statistic summary for normal and Weibull probability density functions ...	195
Table 7-16. Summary statistics for Case Study I	197
Table 7-17. Best-fit distributions for each input parameter	197
Table 7-18. Test statistic summary for normal and Weibull probability density functions ...	198
Table 7-19. Summary statistics for Case Study II	200
Table 7-20. Monte Carlo simulation results based on the derived regression equations	205
Table 7-21. The least and most influential predictor variables on CO ₂ corrosion	206
Table 8-1. R^2 -values for the derived models	218
Table 8-2. R^2 -values for the established models	219

List of figures

Figure 1-1. The production and manufacturing corrosion costs share.....	3
(Koch et al., 2002).....	3
Figure 1-2. Share of each greenhouse gas in total global emissions (IPCC, 2014)	4
Figure 1-3. Keeling curve.	5
(NOAA, 2015)	5
Figure 1-4. World CO ₂ emissions by sector (IEA, 2010)	6
Figure 1-5. Projected Keeling curve based on climate modelling	8
(Meinshausen et al., 2011)	8
Figure 1-6. World total primary energy supply (IEA, 2012)	8
Figure 2-1. Diagram showing CO ₂ corrosion including the effect of O ₂	17
(Song et al., 2005).....	17
Figure 2-2 Pitting corrosion of a steel pipe (Newman, 2010).....	19
Figure 2-3 Pitting corrosion of a steel pipe (Newman, 2010).....	20
Figure 2-4. Mesa-type corrosion in a CO ₂ -rich system (Kermani and Morshed, 2003).....	20
Figure 2-5. Flow-induced localised pipeline corrosion	21
Figure 2-6. Major factors affecting CO ₂ corrosion	22
Figure 2-7. Plot illustrations indicating the effect of the primary parameters on CO ₂ corrosion of carbon steel (Dugstad et al., 1994a).....	24
Figure 2-8. Morphologies for protective and nonprotective scale formation (Crolet et al., 1996, Kermani and Morshed, 2003)	27
Figure 2-9. Schematic of H ₂ S corrosion – corrosion rate of steel in H ₂ S solutions is said to be under mass-transfer control. (Sun and Nesic, 2007).....	32
Figure 2-10. Plot showing comparative trend of elemental contributions to corrosion- resistance (Kermani et al., 2001), (Kermani et al., 2003)	38
Figure 2-11. Plot showing the category of chromium content of steel and corresponding corrosion rate (Kermani et al., 2001; Kermani et al., 2003)	39
Figure 2-12. Phase diagram for pure CO ₂ – the captured gas can be transported strictly on either side of the dashed line without cross over. Dense-phase region represents the most efficient means of pipeline transport. (IEA GHG, 2010).....	42
Figure 2-13. Phase envelopes of CO ₂ with a combination of impurities	43
(Downie et al., 2007).....	43
Figure 2-14. Solubility of CO ₂ in H ₂ O (Spycher et al., 2003)	44
Figure 2-15. Flow loop of input to output via continuous adjustment of weights	47

(Demuth et al., 2009).....	47
Figure 2-16. Basic layered structure of a three-layer artificial neural network.....	48
(Al-Fattah and Startzman, 2003)	48
Figure 2-17. The purelin and log-sigmoid transfer functions (Beale et al., 2014)	48
Figure 2-18. Plot showing the effect of sulphate ion on pitting potential of steel in 3% NaCl solution. (Cottis et al., 1999)	50
Figure 2-19. Steps in fuzzy inference process	52
Figure 2-20. 3-D Surface contour plots showing the relationships between input variables (CO ₂ mole % with temperature and gas production rate with temperature) and the output variable, CO ₂ corrosion rate (Hajizadeh, 2007)	53
Figure 3-1. Plot of Corrosion Rate against Temperature for varying carbon dioxide partial pressures, $P_{CO_2} = 0.01, 0.03, 0.1$ and 0.3MPa (without inhibition from carbonate scale).....	60
Figure 3-2. Typical plot of corrosion rate against temperature from experimental CO ₂ corrosion data (Dugstad et al., 1994b; Halvorsen et al., 1999; Sun and Nesic, 2004)	60
Figure 3-3. Plot of corrosion rate against temperature for varying CO ₂ fugacities, $f_{CO_2} = 0.01, 0.03, 0.1$ and 0.3MPa	63
Figure 3-4. Plot of corrosion rate against temperature for varying CO ₂ fugacities $f_{CO_2} = 0.01, 0.03, 0.1$ and 0.3MPa	65
Figure 3-5. Plot of Corrosion rate against temperature (Conditions are: $T=10-100^\circ\text{C}$, $f_{CO_2}=1\text{MPa}$, $\text{pH}=5.0$ and for varying flowrate velocity)	67
Figure 3-6. Plot of corrosion rate against temperature (Conditions are: $T=10-100^\circ\text{C}$, $U=6.0\text{m/s}$, $\text{pH}=5.0$ and for varying carbon dioxide fugacities).....	68
Figure 3-7. NORSOK M-506-generated and Matlab 2012a-generated plots of corrosion rate against temperature for varying CO ₂ fugacities: 0.03MPa (0.3bar), 0.1MPa (1bar) and 0.3MPa (3bar). Conditions are: $\text{pH}=4$, CO_2 mole = 100% , Shear stress= 2Pa	69
Figure 3-8. Plot of carbon dioxide corrosion against temperature	71
(For varying water cut percentages)	71
Figure 3-9. Plot of carbon dioxide corrosion against temperature	72
(For varying CO ₂ mole percentages)	72
Figure 3-10. Nesic-Postlethwaite-Olsen (NPO) model output for $T=100^\circ\text{C}$ – Tafel plot of potential against \log_{10} (current density).....	75
Figure 3-11. Plot of corrosion rate against temperature for the NPO model for.....	75
CO ₂ fugacities: 0.01MPa (0.1bar), 0.03MPa (0.3bar), 0.1MPa (1bar) and 0.3MPa (3bar). Other conditions where applicable are: $\text{pH}=4$, CO_2 mole = 100% , Shear stress= 2Pa	75

Figure 3-12. Scatterplot of Freecorp model readings against experimental readings for two datasets: Crolet et al, (1999) and Guo et al, (2005)	78
Figure 3-13. Species concentration profile - H ₂ S concentration as a function of the distance from steel surface	79
Figure 3-14. Scatterplot of Freecorp model readings against experimental readings for two datasets: Lyle and Schutt (1998) and Nesic et al. (2008).....	81
Figure 3-15. Plot of corrosion rate against temperature for varying CO ₂ partial pressures for a fluid velocity, $U=0.1\text{m/s}$	83
Figure 3-16. Plot of corrosion rate against temperature for varying fluid velocities.....	83
Figure 3-17. Plot of CO ₂ corrosion rate against temperature for the various models for $f_{CO_2}=0.3\text{MPa}$ (3bar)	85
Figure 4-1. Boxplot showing the overall range (box and whiskers), inter-quartile range (white space in blue edged box), median, skewness and outliers (red plus symbols) in each of the variables in the dataset	90
Figure 4-2. Boxplot showing the range of values, inter-quartile range, median and skewness of the P_{CO_2} and pH variables (Outliers are red plus symbols).....	90
Figure 4-3. Pareto chart showing the spread of the data (variance) among each of the principal components	92
Figure 4-4. Two-dimensional biplot showing the orientation (magnitude and direction) of the variables in the principal component axes	92
Figure 4-5. Scree plot showing decreasing eigenvalues (variances) for all the principal components	93
Figure 4-6. Three-dimensional plot of corrosion rate against temperature and CO ₂ partial pressure (P_{CO_2}) showing regression model through the points.....	95
Figure 4-7. Multiple linear regression model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate	96
Figure 4-8. Variable interactions between:	97
A. P_{CO_2} and Temperature B. Velocity and Temperature	97
Figure 4-9. Variable interactions between:	98
A. pH and Temperature B. Velocity and P_{CO_2}	98
Figure 4-10. Variable interactions between:	99
A. pH and P_{CO_2}	99
B. pH and Velocity.....	99

Figure 4-11. Hougen-Watson nonlinear model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – Error is the difference between experimental and model corrosion rate	101
Figure 4-12. Leverage-CO ₂ corrosion data plot and residuals-CO ₂ corrosion data plot	102
Figure 4-13. Response surface modelling plots – CO ₂ corrosion rate can be investigated by changing the magnitude of each predictor variable.....	103
Figure 4-14. Response surface model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate	103
Figure 4-15. Boxplot showing the overall range (box and whiskers), inter-quartile range (white space in blue-edged box), median, skewness and outliers (red plus symbols) in each of the variables in the dataset.....	105
Figure 4-16. Boxplot showing the range of values, inter-quartile range, median, skewness and outliers of the pH variable	105
Figure 4-17. Pareto chart showing the spread of the data (variance) among each of the Principal Components	107
Figure 4-18. Two-dimensional showing the orientation of the variables in the principal component axes	107
Figure 4-19. Scree plot showing the decreasing magnitudes of eigenvalues for all the principal components.....	108
Figure 4-20. Three-dimensional plot of corrosion rate against temperature and CO ₂ partial pressure (P_{CO_2}) showing regression model through the points.....	109
Figure 4-21. Multiple linear regression model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data.....	110
Figure 4-22. Variable interactions between:.....	111
A. P_{CO_2} and Temperature B. Velocity and Temperature.....	111
Figure 4-23. Variable interactions between:.....	112
A. pH and temperature.....	112
B. Velocity and P_{CO_2}	112
Figure 4-24. Variable interactions between:.....	113
A. pH and P_{CO_2}	113
B. pH and velocity	113
Figure 4-25. Hougen-Watson nonlinear model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data.....	114

Figure 4-26. Leverage-CO ₂ corrosion data plot and residuals-CO ₂ corrosion data plot.....	115
Figure 4-27. Response surface modelling plots – CO ₂ corrosion rate can be investigated by changing the magnitude of each predictor variable	116
Figure 4-28. Response surface model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data	116
Figure 5-1. An artificial model of a neuron showing the transformation of input signals, x_i , to an output signal, y_i (Veelenturf, 1995).....	120
Figure 5-2. Stages in machine learning (Haykin, 1999)	122
Figure 5-3. Flow loop of input to output via continuous adjustment of weights..... (Demuth et al., 2009)	123
Figure 5-4. A simple neuron (Beale et al., 2014).....	123
Figure 5-5. A Single-layer feedforward network.....	125
Figure 5-6. A multilayered feedforward network (Al-Fattah and Startzman, 2003).....	125
Figure 5-7. A recurrent network with no self-feedback loop (Haykin, 1999)	126
Figure 5-8. Flowchart showing the sequence of steps in the development of the neural network model.....	127
Figure 5-9. Block diagram of the neural network model.....	128
Figure 5-10. Arbitrary plot of predicted corrosion rate against experimental corrosion rate	129
Figure 5-11. Combined performances of each training function defines the degree of suitability of the transfer function in determining the final NN model parameters	131
Figure 5-12. Line plots of the sum total of absolute error against number of neurons for the logsig transfer Function	132
Figure 5-13. Line plots of R^2 -value against number of neurons for the logsig transfer function	133
Figure 5-14. Line plots of R^2 -value against number of neurons for the tansig transfer function	134
Figure 5-15. Multiple bar charts showing the Mean R^2 -coefficients for each training function for both logsig and tansig transfer functions.....	134
Figure 5-16. Multiple bar charts showing the absolute error means for each training function for both logsig and tansig transfer functions.....	135
Figure 5-17. Line plots of sum total of absolute error against number of neurons for the tansig transfer function	135
Figure 5-18. NN model corrosion rate against experimental corrosion rate.....	137
Figure 5-19. Error bar chart	138

Figure 5-20. Bar chart showing the distribution of data-points from each source	139
Figure 5-21. Bar chart showing the distribution of data-points for the grouped experimental corrosion rates	140
Figure 5-22. Bar chart showing the distribution of data-points for the recorded temperatures	140
Figure 5-23. Corrosion rate against temperature plot for the training dataset.....	141
Figure 5-24. Line plots of R^2 -values of training functions against number of neurons for the logsig transfer function.....	142
Figure 5-25. Line plots of R^2 -values of training functions against number of neurons for the tansig transfer function	142
Figure 5-26. Bar charts showing the means of the R^2 -values of each training function for both logsig and Tansig transfer functions.....	143
Figure 5-27. Bar charts showing the means of absolute corrosion rate error of each training function for Both logsig and tansig transfer functions	144
Figure 5-28. Line plots of sum of absolute corrosion rate error against neuron configuration for the traincgf and trainlm functions	145
Figure 5 -29. NN Model corrosion rate against experimental corrosion rate plot.....	147
Figure 5-30. Error bar chart – Error is the difference between experimental and model corrosion rate	147
Figure 5-31. Corrosion rate against temperature plot for the testing dataset with NN model predictions	148
Figure 6-1. Close connection between classical logic and classical set and their relationships with predicates of logic.....	152
Figure 6-2. Membership function of a fuzzy set	153
Figure 6-3. Triangular and trapezoidal membership functions	154
(Fuzzy Logic Toolbox User’s Guide, 2015).....	154
Figure 6-4. Linear piece-wise membership function (Ukil, 2010).....	155
Figure 6-5.Generalised form of a continuous membership function.....	155
(Fuzzy Logic Toolbox User’s Guide, 2015).....	155
Figure 6-6. Fuzzy inference diagram showing all parts of the fuzzy inference process – from fuzzification through to defuzzification (Fuzzy Logic Toolbox User’s Guide, 2015).....	158
Figure 6-7. pH membership function plots.....	159
Figure 6-8. Three-dimensional plot of corrosion rate against P_{CO_2} and temperature – result of Mamdani FIS	160

Figure 6-9. Mamdani FIS plot of corrosion rate against temperature.....	161
Figure 6-10. Mamdani FIS plot of corrosion rate against CO ₂ partial pressure	162
Figure 6-11. Three-dimensional plot of corrosion rate against flowrate velocity and temperature – result of Mamdani FIS	163
Figure 6-12. Mamdani FIS plot of corrosion rate against flowrate velocity.....	163
Figure 6-13. Mamdani FIS model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate.....	164
Figure 6-14. A generic fuzzy inference system showing input variables to the left (T , P_{CO_2} , pH, fluid flowrate) and the output variable to the right (Corrosion Rate)	165
Figure 6-15. Rule Viewer for all seven rules showing mini graphical representations of each of the input variables and their effect on the output variable.....	167
Figure 6-16. Three-dimensional plot of corrosion rate against P_{CO_2} and temperature – result of Mamdani FIS.....	168
Figure 6-17. Three-dimensional plot of corrosion rate against P_{CO_2} and temperature – result of Sugeno FIS	168
Figure 6-18. Mamdani FIS plot of Corrosion Rate against CO ₂ partial pressure	169
Figure 6-19. Sugeno FIS plot of Corrosion Rate against CO ₂ partial pressure	170
Figure 6-20. Mamdani FIS Model Corrosion Rate against Experimental Corrosion Rate plot and Error Bar chart for selected test data – Error is the difference between Experimental and Model Corrosion Rate.....	171
Figure 7-1. Normal probability plot and Weibull probability plot	179
Figure 7-2. Normal probability density and Weibull probability density plots for experimental CO ₂ corrosion rate dataset.....	181
Figure 7-3. Number of runs used in Monte Carlo simulations	183
Figure 7-4. Monte Carlo simulation using statistical parameters of the experimental dataset	183
Figure 7-5. Probability density function plot for Monte Carlo simulation of low pressure CO ₂ corrosion rate.....	186
Figure 7-6. Cumulative density function plot for Monte Carlo simulation of low pressure CO ₂ corrosion rate.....	186
Figure 7-7. Probability density function plot for Monte Carlo simulation of low pressure CO ₂ corrosion rate.....	188

Figure 7-8. Cumulative density function plot for Monte Carlo simulation of low pressure CO ₂ corrosion rate	189
Figure 7-9. Sensitivity analysis – effect of each parameter on corrosion rate.....	190
Figure 7-10. Normal probability plot and Weibull probability plot	192
Figure 7-11. Normal probability density and Weibull probability density plots for experimental CO ₂ corrosion rate dataset	193
Figure 7-12. Monte Carlo simulated probability density function plot showing normal and Logistic curve fits	194
Figure 7-13. Probability density function plot for Monte Carlo simulation of high pressure CO ₂ corrosion rate	196
Figure 7-14. Cumulative density function plot for Monte Carlo simulation of high pressure CO ₂ corrosion rate	196
Figure 7-15. Probability density function plot for Monte Carlo simulation of high pressure CO ₂ corrosion rate	198
Figure 7-16. Cumulative density function plot for Monte Carlo simulation of high pressure CO ₂ corrosion rate	199
Figure 7-17. Sensitivity analysis – effect of each parameter on corrosion rate.....	201
Figure 7-18. Probability plot showing the Weibull distribution as the best-fit for CO ₂ partial pressure variable	203
Figure 8-1. The 1991 De Waard model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate	209
Figure 8-2. The 1995 De Waard model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data.....	210
Figure 8-3. Norsok model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data	211
Figure 8-4. Freecorp model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate	212
Figure 8-5. The 1991 De Waard model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model Corrosion rate	213
Figure 8-6. The 1995 De Waard model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data.....	214

Figure 8-7. Norsok model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data..... 215

Nomenclature

a	neural network output signal (scalar); algorithm; equation of state (EOS) parameter →measure of the attractive forces between molecules
a_1, a_2, a_3	algorithms for the purelin, logsig and tansig transfer functions respectively
a_{H^+}	activity of H^+ species
$[a, b]$	a given range in which the random variable X may fall within
A	Arrhenius pre-exponential factor
A_{H_2S}, B_{H_2S}	Arrhenius constants, $A_{H_2S} = 1.30 \times 10^{-4} \text{ mol/m}^2\text{s}$ and $B_{H_2S} = 15500\text{J/mol}$
A_{OS}	area of the mackinawite scale (m^2)
b	bias to the neural network input signal; equation of state (EOS) parameter →related to size of the molecule
b_a	anodic Tafel slope on the Tafel plot
b_c	cathodic Tafel slope on the Tafel plot
c_{b,H_2S}	bulk concentration of H_2S in the liquid phase (mol/m^3)
c_{s,H_2S}	concentration of H_2S on the steel surface, $c_{s,H_2S} = 1.00 \times 10^{-7} \text{ in mol/m}^3$
C	mixed fluid flow constant (specific to a given material)
C_{MT}	the 1995 De Waard-Lotz-Dugstad correlation term - highest possible mass transfer rate of corrosive species
C_R	the 1995 De Waard-Lotz-Dugstad correlation term - highest possible reaction rate (when mass transfer is infinitely fast)
C_{CORR}	the 1995 De Waard-Lotz-Dugstad correlation CO_2 corrosion rate (mm/year)
Corr Rate	CO_2 corrosion rate (mm/year)
CR_{H_2S}	corrosion rate of steel due to H_2S (mm/year)
$[C_{CO_2}]$	concentration of CO_2 in aqueous solution
d	pipe diameter (m)

D_{H_2S}	diffusion coefficient for dissolved H ₂ S in water, $D_{H_2S} = 2.00 \times 10^{-9} \text{m}^2/\text{s}$
E_A	activation energy (KJ/mol)
E_{corr}	corrosion potential at the point where the sum of the individual current densities passes through zero
f	function that transforms predictor variables (input) to response variables (output); neural network transfer function that produces output a
$f(x)$	probability density function (pdf) of X
f_{CO_2}	CO ₂ fugacity (MPa)
$f(pH)_t$	complex Norsok function of pH and temperature
F	Faraday constant
$F(x)$	integral of $f(x)$ and is the cumulative density function (cdf) of X over the range $[a, b]$
$[H^+]_b$	concentration of hydrogen ions in bulk solution
i_{corr}	corrosion current at E_{corr}
$i_{(H^+)} \text{ or } i_{\alpha(H^+)}, i_{o(H^+)}$	instantaneous activation controlled cathodic current and initial exchange current density for hydrogen ion respectively (A/m ²)
$i_{Fe}, i_{o(Fe)}$	instantaneous anodic current term for the dissolution of the metal (Fe) and initial exchange current density of the metal respectively (A/m ²)
$i_{H_2CO_3}$	carbonic acid cathodic exchange current density (A/m ²)
$i_{lim(H^+)}^d$	limiting current
H_0	null hypothesis
k	Arrhenius reaction rate coefficient
k_m, k_r	mass transfer rate constant and activation reaction kinetics constant of CO ₂ corrosion reaction – applicable to De Waard and NPO correlations
k_{m,H_2S}	mass transfer coefficient for H ₂ S in the hydrodynamic boundary layer, $k_{m,H_2S} = 1.00 \times 10^{-4} \text{m/s}$ in nearly stagnant condition
k_W	shape parameter for the Weibull probability distribution

K_1	dissociation constant for the hydration of aqueous H ₂ S
K_2	dissociation constant for bisulphide ion: HS ⁻
K_{CO_2}, K_{H_2O}	equilibrium constants for the solubility of H ₂ O in CO ₂ and CO ₂ in H ₂ O respectively
K_H	Henry's law constant (mol/litre.MPa)
K_{sp}	solubility limit (mol/l) ²
K_t	Norsok constant dependent on temperature
m_{H^+}	molar concentration of H ⁺ species
m_{os}	mass of the mackinawite scale (kg)
n	all possible values of the random variable X; the sum of the product of the input scalar (p) and weight (w) added to bias (b); any given number
N	randomly generated number
p_1, p_2, \dots, p_j	neural network input signal (scalar)
P, P^0 and P_{tot}	pressure (applied pressure), atmospheric pressure (0.101325MPa) and the total system pressure respectively (MPa)
P_{CO_2}	CO ₂ partial pressure (MPa)
P_{H_2S}	H ₂ S partial pressure (MPa)
Pr	probability of an event occurring
R	molar gas constant (J/mol.K)
R^2	statistical coefficient of determination (dimensionless)
S	wall shear stress (Pa)
Sat	supersaturation (dimensionless)
T	temperature (°C)
T_0	lowest Monte Carlo simulated temperature (°C)
T_1	highest Monte Carlo simulated temperature (°C)
T_{scale}	scaling temperature (K)
T_{sim}	Monte Carlo simulated temperature (°C)
T_{std}	temperature of gas at standard conditions (25°C) and atmospheric pressure

T_k	Absolute temperature (K)
U	fluid flowrate velocity (m/s)
V	volume (m ³)
V_{corr}	the 1975 and 1991 De Waard-Milliams correlation CO ₂ corrosion rate (mm/year)
V_e	mixed fluid velocity (m/s)
\bar{V}_{CO_2}	average partial molar volume of CO ₂ (cm ³ /mol)
\bar{V}_{H_2O}	average partial molar volume of water (cm ³ /mol)
w_1, w_2, \dots, w_j	weights to the neural network input signal
x	an unknown quantity to be fuzzified (transformed) into a fuzzy set; predictor or input variable ($T, P_{CO_2}, U, \text{pH}$)
x_1, x_2, \dots, x_n	input variables T, P_{CO_2}, U and pH for the regression equation
$x_i, y_i, (x_i, y_i)$	experimental corrosion rate, model predicted corrosion rate and abscissa-ordinate coordinate pair respectively
x_{CO_2}	mole fraction of CO ₂ in the aqueous phase
\bar{x}, \bar{y}	mean values of x_i and y_i respectively
X	discrete or continuous random variable
y	response or output variable; corrosion rate (mm/year) for generalised regression equations; theoretical steel surface ($y=0$)
y_{H_2O}	mole fraction of H ₂ O in the CO ₂ phase

Greek letters

$\beta_0, \beta_1, \beta_2, \dots, \beta_n$	constants or predictor variable coefficients for regression equation
η	overpotential (volts, V)
γ_{H^+}	activity coefficient of H ⁺ species
δ_{os}	thickness of the mackinawite scale (m)
δ_s	theoretical distance of aqueous boundary layer to steel surface ($y=\delta_s$)
ε	outer mackinawite scale porosity

ε_r	set of independent random disturbances for generalised regression equations
λ	scale parameter for the Weibull probability distribution; adjusts the exponential decay for the exponential function
ψ	outer mackinawite scale tortuosity factor
μ	mean for the normal probability distribution; scale parameter for the log-normal distribution
$\mu(x)$	fuzzy membership function
Φ_{H_2O}	fugacity coefficient for H ₂ O (dimensionless)
Φ_{CO_2}	fugacity coefficient for CO ₂ (dimensionless)
ρ_{FeS}	density of the mackinawite (FeS) scale (kg/m ³)
ρ_m	mixed fluid density of gas and liquid (kg/m ³ or lb/ft ³)
σ	standard deviation for the normal probability distribution; scale parameter for the lognormal distribution
θ	location parameter for the Weibull probability distribution

Chapter 1. Introduction

1.1 Corrosion Costs

The effect of carbon dioxide (CO₂) corrosion on pipelines is of great relevance to the petroleum as well as the Carbon Capture and Storage (CCS) industries. CO₂ corrosion is responsible for lost production as it brings about the gradual degradation of pipe internals with time.

The U.S. refinery capacity was approximately one-fifths of the total world refinery capacity as of 1991 (Campbell, 1991), and is ~27% of the global refinery capacity as of 2014. In 2002, the production and manufacturing industry accounted for approximately 13% (or \$18 billion) of the corrosion costs in industry categories (Koch et al., 2002). The share for each of the industry categories is shown in Table 1-1.

Table 1-1. Share of corrosion costs for the U.S. industry categories (Koch et al., 2002)

Industry Category	Percentage Share (%)
Infrastructure	16
Utilities*	35
Transportation	21
Production and Manufacturing	13
Government	15

* Utilities includes corrosion costs from the natural gas distribution system (~10% of Utilities)

The industry category of interest is production and manufacturing, as it is accountable for most of the corrosion costs. The breakdown of costs in this category is shown in Figure 1-1. From the pie chart of Figure 1-1, the oil and gas industry costs cut across the oil and gas exploration, petroleum refining as well as the petrochemical and pharmaceutical sectors. These costs amount to 39%.

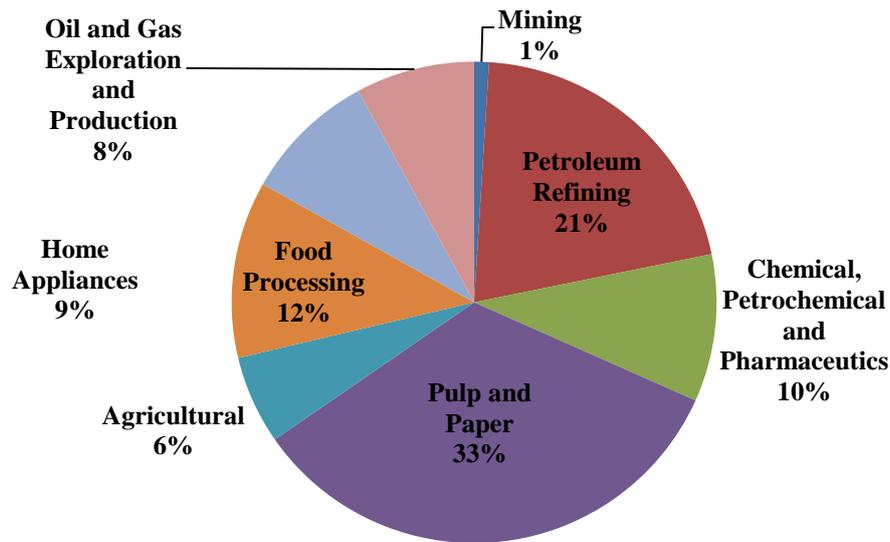


Figure 1-1. The production and manufacturing corrosion costs share

(Koch et al., 2002)

In addition, the natural gas distribution corrosion costs account for 10% of the Utilities industry category, as seen from Table 1-1. Hence, the cumulative share for the petroleum industry was in excess of 40% of the U.S. total corrosion costs as of 2002. As of 2014, the cost of corrosion in the production and manufacturing sector was \$34.4 billion, almost double the reported 2002 amount, with the oil and gas industry accounting for more than half.

As of 2011, the annual cost of corrosion rose to \$2.2 trillion, amounting to over 3% of the world's gross domestic product (GDP) (Hays, 2011). The cost of general corrosion is said to be between 3 to 5% of an industrialised nation's GDP (Schmitt et al., 2009). Also, approximately 60% of all oil and gas field failures are related to CO₂ corrosion (Nyborg, 2005).

1.2 Environmental Impact of CO₂ emissions

According to the Inter-governmental Panel for Climate Change (IPCC), observational evidence suggests that the composition of the atmosphere is changing as a result of increased concentrations of greenhouse gases (GHG) such as carbon dioxide (CO₂) and methane (CH₄).

A recent report by the IPCC also suggests that most of the observed increase in global average temperatures since the mid-20th century is most likely due to the observed increase in anthropogenic greenhouse gas concentrations (IPCC, 2014).

CO₂ is evidently the most significant greenhouse gas given that annual emissions of the gas have risen by almost 80% between 1970 and 2010, from 21 to 49 gigatonnes, and as of 2010, represented approximately 77% of the total global greenhouse gas emissions (IPCC, 2014). The pie chart in Figure 1-2 shows the share of each greenhouse gas in global emissions.

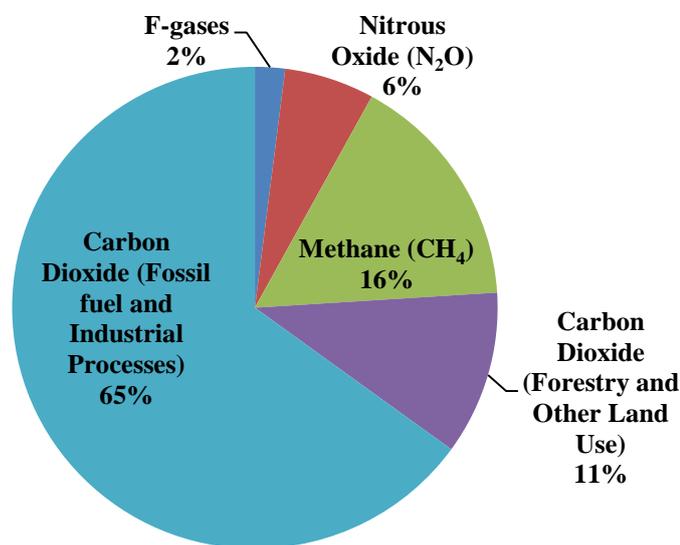


Figure 1-2. Share of each greenhouse gas in total global emissions (IPCC, 2014)

On this note, atmospheric CO₂ concentration has elevated beyond its pre-industrial revolution magnitude, 280ppm, to the value of 400ppm, as of 2013 (IPCC, 2014; Dlugokencky, 2014). This observational evidence is depicted by the rise in atmospheric CO₂ concentration as shown in the Keeling curve of Figure 1-3. From Figure 1-3, it is seen that atmospheric CO₂ concentration increased steadily on an annual basis from 1960 to date. This rise in greenhouse gas levels ultimately results in projected changes in climate such as increasing temperatures, changes in precipitation, sea-level rise and increased frequency and intensity of some extreme climatic events leading to increased climate variability.

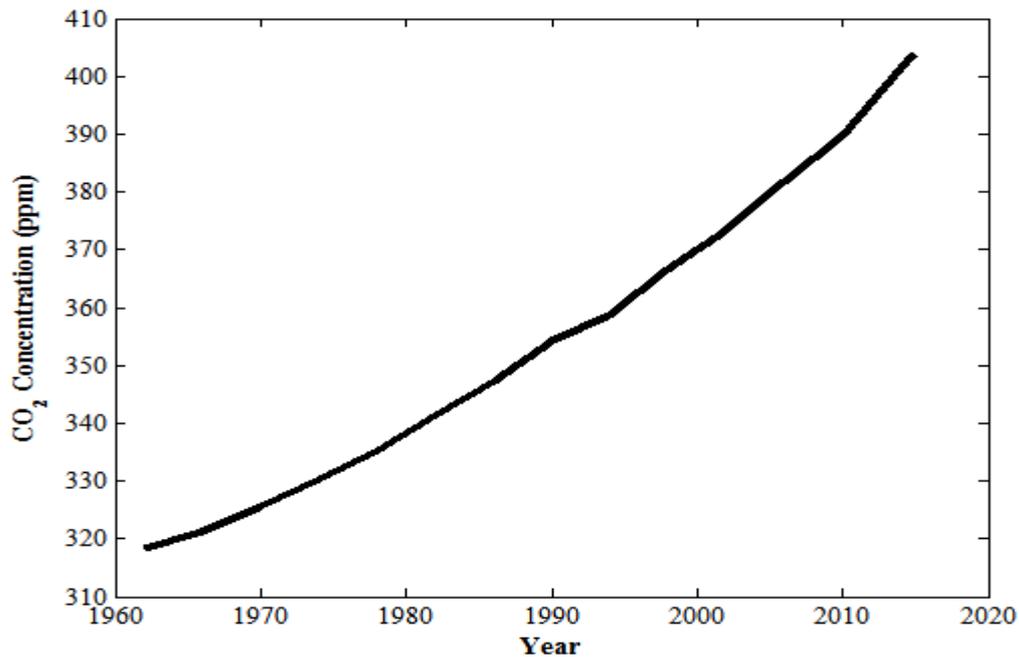
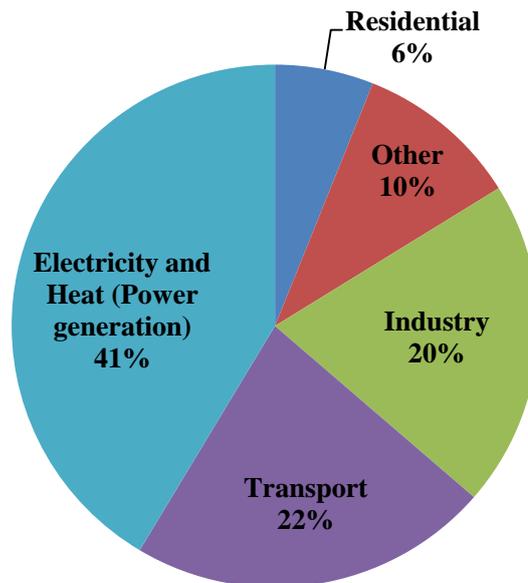


Figure 1-3. Keeling curve (NOAA, 2015)

Globally, these environmental changes have a profound impact on biodiversity such that the ecosystems are unable to adopt or induce self-regulatory mechanisms to nullify the changes, thus mitigation processes are required. CO₂ is being emitted anthropogenically at a faster rate than the natural processes of the carbon cycle are able to remove it. One major reason for the earth's ecosystems inability to counteract the effect of these changes is land use – deforestation. The impact of deforestation is two-fold; firstly, it reduces the number of trees, which are the earth's natural carbon sinks and secondly, when trees are felled, they are often burned, releasing their carbon contents into the atmosphere.

Mitigation is the anthropogenic intervention to reduce net greenhouse gas emissions that would lessen the pressure on natural ecosystems as well as human systems from the effects of climate change. Furthermore, mitigation involves the reduction of greenhouse gases by the reduction of fossil fuel use or by the increase in the natural carbon uptake rate by ecosystems. An examination of CO₂ emissions by sector indicates that electricity and heat generation is the largest sector (41%) responsible for CO₂ emissions as shown in Figure 1-4.



*Other refers to emissions from commercial services, agriculture, fishing and energy industries other than electricity and heat generation

Figure 1-4. World CO₂ emissions by sector (IEA, 2010)

Further evidence of the impact of CO₂ emissions can be inferred from climate modelling. In 2014, the IPCC proposed four Representative Concentration Pathway (RCP) scenarios. These are: RCP 2.6, RCP4.5, RCP6.0 and RCP8.5. The RCPs are consistent with a wide range of possible changes in predicted anthropogenic GHG emissions (IPCC, 2014). The names of the RCPs come from their respective radiative forcing values from the end of the 21st century with reference to pre-industrial revolution values: 2.6, 4.5, 6.0 and 8.5W/m² (IPCC, 2014). Radiative forcing of a gas is defined as the difference between the incident solar radiation on the earth's atmosphere and the outgoing infrared radiation caused by the increased concentration of the gas (IPCC, 2014). CO₂ has the highest contribution to radiative forcing and accounts for more than 80% of the radiative forcing from GHGs for the climate modelling pathway scenarios due to its high atmospheric concentration, warming effect, long residence time in the atmosphere and its global spatial distribution (Clarke et al., 2007). Since 1750, its radiative forcing has increased by 1.88W/m² or ~65% of the increased forcing by all long-lived greenhouse gases (LLGHG) (Dlugokencky et al, 2014). Table 1-2 is a summary of the RCP scenarios.

Table 1-2. RCP scenarios for the end of the 21st century (IPCC, 2014)

Future Scenario	Definitions	Global Mean Surface Warming (°C)	Global Mean Sea level rise (m)
RCP2.6	Assumes global annual emissions peak between 2010 and 2020	1.00	0.40
RCP4.5	Assumes global annual emissions peak around the year 2040	1.80	0.47
RCP6.0	Assumes global annual emissions peak around the year 2080	2.20	0.48
RCP8.5	Assumes global annual emissions increase indefinitely throughout the 21 st century	3.70	0.63

Of all these scenarios, RCP2.6 represents the best possible future pathway for emissions however; it represents a near-ideal situation. In terms of the current global emission trends, based on socio-economic factors, RCP 4.5 and RCP6.0 represent the more likely future pathways while RCP8.5 is an extreme. The effect each of these scenarios has on the CO₂ atmospheric concentrations is shown in Figure 1-5.

It can be observed from Figure 1-5 that the atmospheric CO₂ concentration peaks for RCP2.6, 4.5 and 6.0 do not coincide with the stated emissions peaks listed in Table 1-2. For instance, the emissions peak for RCP2.6 is between 2010 and 2020, yet the atmospheric concentration of the gas does not peak until approximately 2040. The atmospheric CO₂ concentration peaks occur much later due to the long-life or long residence time of CO₂.

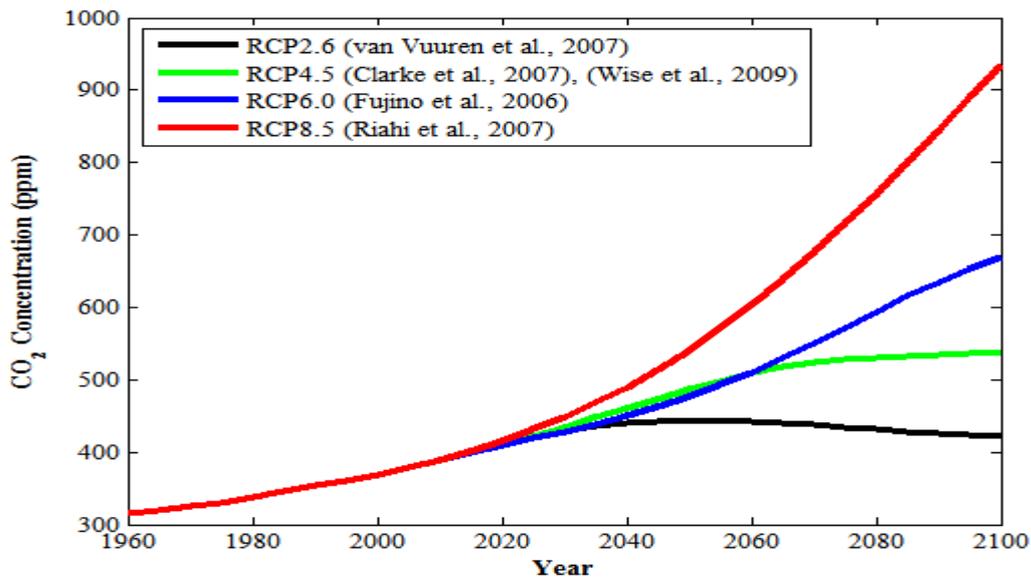


Figure 1-5. Projected Keeling curve based on climate modelling
 (Meinshausen et al., 2011)

1.3 The Importance of CCS

Worldwide, the power generation sector is heavily reliant on coal which is the most carbon intensive of fuels thus amplifying its share in global emissions as shown in the pie chart of Figure 1-4.

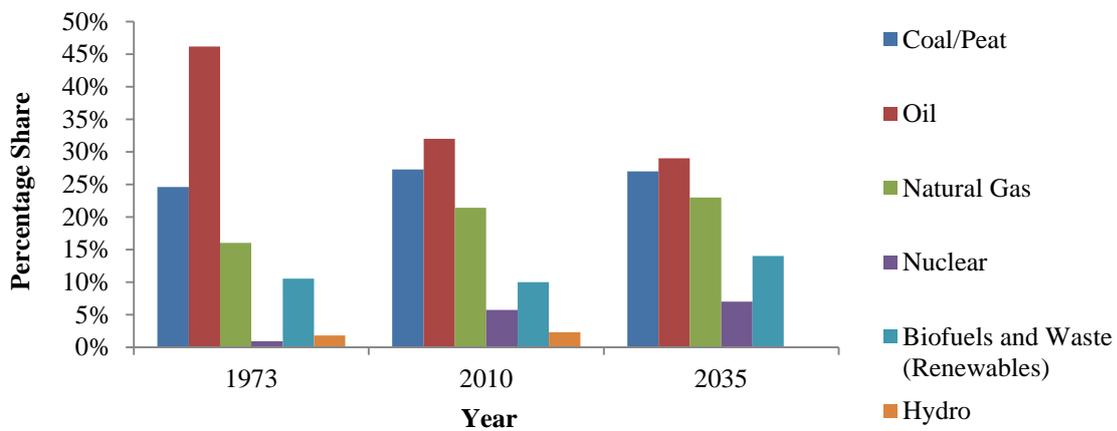


Figure 1-6. World total primary energy supply (IEA, 2012)

It is worth noting that from Figure 1-6, fossil fuel (oil, gas and coal) combustion accounts for approximately 80% of the total world CO₂ emissions. Also shown in Figure 1-6, is the projected world energy consumption for the year 2035, and this shows very little change in the usage of coal. This implies that while some countries will inevitably reduce their reliance on coal, other countries may increase their consumption of it – in general, some countries will depend on it more than others.

In order to reduce greenhouse gas emissions however, coal-fired plants will have to be deployed in conjunction with CCS technologies. This is a low-cost option (World Energy Council, 2007; IPCC, 2014). In a context where a gas-fired system is deployed instead of the coal-fired alternative, CCS technologies should be used alongside even though, generally speaking, the former is more efficient and environmentally cleaner than the latter (IEA, 2012).

1.4 CO₂ Corrosion in Petroleum and CCS pipelines

Corrosion of pipelines under low pressure CO₂ conditions has been covered extensively with readily accessible corrosion data made available (Dugstad et al., 1994b; Halvorsen et al., 1999; Sun and Netic, 2004). CO₂ corrosion under low pressure conditions is characteristic of the oil and gas industry and comes about when CO₂ in produced gases dissolve in water, thus forming carbonic acid (Garverick, 1994). For gas condensate wells producing formation water, produced gas coupled with acid gases, (H₂S, CO₂), if liquid water is allowed to condense on the tubing, CO₂ corrosion may occur (Garverick, 1994). CO₂ corrosion has also become an increasingly common phenomenon in enhanced oil recovery applications, where CO₂ and water mixtures are injected directly into the depleted well to facilitate extraction (Garverick, 1994; Satter et al., 2007).

Corrosion of pipelines under high pressures comes about by the transportation of the gas in the dense liquid-phase from its point of capture (Power plant) to a storage site (abandoned coal mine or oilfield) (Downie et al., 2007). Elevated pressures in this context exceed 7.4MPa (74bar), hence corrosion risks are possible. Corrosion risks are potentially more severe when impurities such as traces of water droplets, H₂S, SO_x and NO_x gases are present (Downie et al., 2007). These gases are present in CO₂ streams depending on the fuel source of the power-generating emitter: fuel or flue gas (Granite and O'Brien, 2005).

Fuel gases are combustible gases which are direct products of the partial combustion of a fuel during gasification. They usually contain higher amounts of CO than flue gases (Zevenhoven and Kilpinen, 2004). Flue gases are the final gaseous products of complete combustion and typically contain higher amounts of CO₂ than fuel gases (Zevenhoven and Kilpinen, 2004). The costs associated with pipeline CO₂ corrosion coupled with the unique challenges of modelling corrosion risks at elevated CO₂ pressures, in order to reduce global emissions of the gas signifies the importance of a research undertaking in this field.

1.5 Aims and Objectives

The aims and objectives for this research are:

1. The writing-up of a literature review in order to be fully aware of the latest research activities being carried out in the field of CO₂ corrosion. Preparation of the literature review would aid the preparation of a project Gantt chart (see Appendix A1.1), planning of project activities and to direct the focus of the research.
2. The gathering of CO₂ corrosion data from sources in research literature. Data is to be collected from multiple experimental sources. In these sources, corrosion tests were carried out on samples of ferritic-pearlitic carbon steel for low pressure data. One such chemical composition for the low pressure CO₂ dataset is presented in Table 1-3 (Dugstad et al., 1994b):

Table 1-3. Weight percentage chemical composition of St.52 carbon steel (Dugstad et al., 1994b)

C	Si	Mn	S	P	Cr	Ni	V	Mo	Cu
0.180	0.340	1.500	0.017	0.023	0.080	0.030	0.003	0.030	0.100

Other elements include: Al=0.009, Sn=0.000, Nb=0.002 and Fe= 97.686

St.52 is a carbon-manganese steel. For the high pressure dataset, corrosion experiments were carried out on a variety of steels. However one common steel type

was the X65 pipeline carbon steel with a ferritic-pearlitic microstructure (Zhang et al., 2012a). Its chemical composition is shown in Table 1-4:

Table 1-4. Weight percentage chemical composition of X65 carbon steel (Zhang et al., 2012a)

C	Si	Mn	P	S	N	Cr	Mo	Ni	Cu
0.100	0.310	1.480	0.350	0.005	0.007	-	-	-	-

Iron is the only other element; Fe= 97.748

3. The preparation of a database for storage of the gathered experimental data. A database was created using Microsoft Excel (see Appendix A1.2).
4. The analysis of the stored data. This involves the plotting of profiles for corrosion rates against temperature or with other variables such as the CO₂ partial pressure, flow velocity, and pH, depending on what is being modelled at that instance.
5. The writing, debugging and execution of Matlab model codes for the respective established CO₂ corrosion models obtained from research literature. All model codes are written in Matlab 2012a. These models include the 1975, 1991 and 1995 De Waard as well as the Norsok and Nescic-Postlethwaite-Olsen (NPO) models. The limitations of each of the models are then fully discussed.
6. The use of the R^2 -statistic as a means for assessing model performance.
7. The division of the low and high pressure CO₂ datasets into model-developing and testing datasets.
8. The development of statistical models. Statistical analysis would include descriptive statistics, principal component analysis (PCA) and regression of the respective corrosion datasets.

9. The development of neural network (NN) models for both low and high pressure CO₂ corrosion data based on the designated model-developing datasets. Given that the use of NNs as prediction tools for low pressure CO₂ data has been established (Nesic and Vrhovac, 1999), but not for high pressure; it would be useful to develop a NN model for predicting high pressure CO₂ corrosion as this would constitute a novel approach.
10. The development of a fuzzy inference system (FIS) model capable of predicting corrosivity for the low and high pressure CO₂ corrosion datasets.
11. The application of Monte Carlo simulations to develop a parametric sensitivity study on the influence of the predictor variables (temperature, CO₂ partial pressure, flow velocity and pH) on CO₂ corrosion rates; for both low and high pressure CO₂ corrosion datasets. CO₂ corrosion research literature has very limited content when it comes to the use of statistical tools and Monte Carlo modelling. Some aspects of statistical modelling such as the use of PCA as well as the Monte Carlo-simulated sensitivity analysis constitute novel approaches in this field.
12. The established 1991 and 1995 De Waard, Norsok and Freecorp models are tested with the same test datasets used with the derived models. The model results based on the application of the R^2 -coefficient for both low and high pressure CO₂ datasets are compiled together and discussed.

1.6 Thesis Structure and Content

This Chapter has so far, dealt with the monetary costs of CO₂ corrosion and its impact on world climate. There was also a discussion of the recently adopted climate modelling scenarios applicable from current timelines until the end of the 21st century. The thesis aims and objectives have been fully discussed, showing a progression from a review of the literature to data collection, storage and analysis. Specific modelling techniques considered to be novel in approach with respect to CO₂ corrosion research literature have also been highlighted.

The following chapters have generally been organised the same way as the ordered list of the thesis aims and objectives. Chapter 2 is the literature review for CO₂ corrosion and this is dealt with in great detail. Topics covered include: CO₂ corrosion mechanism, the factors affecting CO₂ corrosion, the different types, an introductory account of CO₂ corrosion models as well as the methodology for modelling CO₂ corrosion from low to high CO₂ partial pressures. Also discussed are introductory reviews of the use of NNs and fuzzy logic as tools suitable for CO₂ corrosion prediction.

Chapter 3 deals with the established CO₂ corrosion correlations and models. These include the aforementioned De Waard, Norsok and Freecorp models. The NPO model is also discussed as it is the basis for the development of the Freecorp model. These models are compared against each other by applying identical parametric conditions for the temperature, CO₂ partial pressure, flow velocity and pH and assessing their predictions. Other topics covered include organic acid and H₂S corrosion.

Chapter 4 covers the statistical analysis of the low and high pressure CO₂ datasets. Explanations on the full descriptive statistics, principal component analysis and regression relationships are outlined. Also, response surface model plots for the CO₂ corrosion datasets are discussed. Again, the R^2 -coefficient is used to report model performances for the derived regression equations, which are subsequently used in Chapter 7.

In Chapter 5, the methodology by which a NN model is developed for the purpose of predicting corrosion rates for both low and high pressure CO₂ datasets is discussed. The basic definition of the R^2 -statistic from first principles is also outlined. The methodology for NN model development involves the use of performance tests based on the R^2 -value for given

Matlab 2012a training and transfer functions. A process of elimination is carried out to determine which combination of transfer and training functions is best-suited to CO₂ corrosion modelling. Also, the determination of the final NN model size is discussed and this is proven to be a direct consequence of the size of the datasets used. The low and high pressure CO₂ datasets were each split into model-developing and testing sets. The R^2 -value is then used to report model performance following testing with test set.

Chapter 6 covers fuzzy modelling. An account is given of how fuzzy logic originated. The various membership functions are discussed. The methodology for the development of a fuzzy logic model based on the creation and applicability of rules ('if-then') is explained. The R^2 -statistic is used as an assessment of model performance when tested with a test set.

The seventh chapter covers Monte Carlo simulation. Definitions of relevant concepts such as the random variable and the different types of probability distributions are stated. A brief basis is outlined on how to determine best-fit distributions for a given dataset. The low and high pressure CO₂ datasets are analysed in order to determine their best-fit theoretical distributions. The pre-determined regression equations are then used as case study scenarios for establishing a parametric sensitivity analysis for CO₂ corrosion rates in terms of the given predictor variables.

Chapter 8 is an extended discussion of the performance results obtained for the established models outlined in Chapter 3. An identical test dataset to the ones used for the assessment of performances of the derived models in Chapters 5 to 8, is used as the basis for evaluating the accuracy of model predictions. Performance results of the derived models in Chapters 4 to 8 are discussed in an expansive manner. The advantages, limitations and their applicability are explained in detail.

The Conclusions and Recommendations for Further work are outlined in Chapter 9. Generally, the conclusion is discussed in the same logical manner as the thesis aims and objectives. Furthermore, those aspects of modelling that have are beyond the scope of this research are stated as recommendations for further work.

Chapter 2. Literature Review on CO₂ Corrosion

2.1 CO₂ Corrosion - Introduction

In Section 1.4, the formation of carbonic acid in pipelines was briefly discussed. The scenarios in which low and high pressure CO₂ corrosion are prevalent in the petroleum and CCS industries respectively, were also explained. In this chapter, the mechanism and types of CO₂ corrosion will be discussed. The major factors influencing CO₂ corrosion are also outlined.

Some established models developed for the prediction of CO₂ corrosivity are also outlined. Modelling techniques for the accurate prediction of high pressure CO₂ corrosion in pipelines such as neural networks and fuzzy logic are also discussed as viable research directives in the latter sections of this chapter.

Carbon dioxide corrosion was first documented to occur as early as the 1940's in the U.S. oil and gas industry (Perez, 2013) and has since been followed by several studies on corrosion rate prediction (Crolet and Bonis, 1991; Song et al., 2005; Hernandez et al., 2006). CO₂ corrosion, also termed sweet corrosion involves the dissolution of gaseous carbon dioxide into an aqueous phase where, via a hydration reaction with water forms carbonic acid, H₂CO₃, the specie which induces corrosive attack on the surface of the metal (Kermani and Morshed, 2003).

Dry carbon dioxide is itself non-corrosive at oilfield temperature and pressure conditions (T=0-100°C, P<7.4MPa) however when dissolved in an aqueous solution, it initiates a series of electrochemical reactions culminating in the aqueous phase coming in contact with the metal surface thus inducing the dissolution of the latter.

2.2 An Outline of the CO₂ Corrosion Process

In general, gaseous carbon dioxide exerts a partial pressure on water resulting in its dissolution. The aqueous carbon dioxide then undergoes hydration thus forming carbonic acid which does not fully dissociate, unlike strong mineral acids, which completely dissociate in aqueous media (Kermani and Morshed, 2003).

The following set of equilibria depicts its incomplete dissociation:



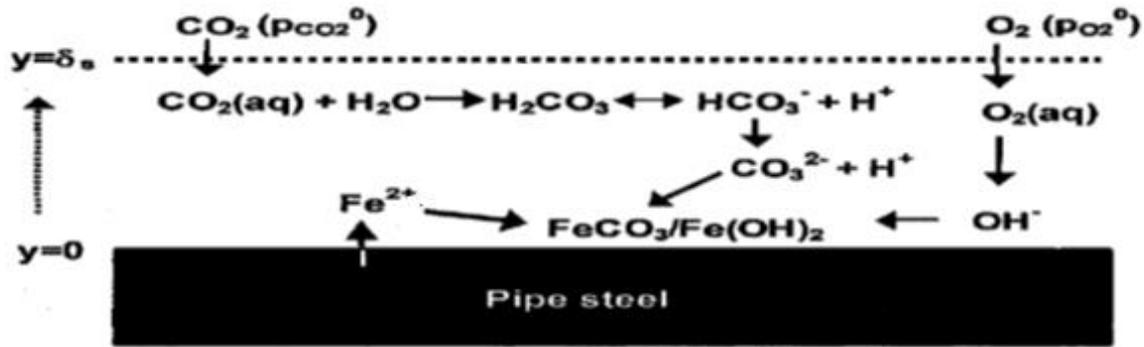


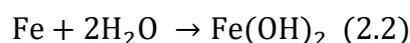
Figure 2-1. Diagram showing CO₂ corrosion including the effect of O₂

(Song et al., 2005)

Figure 2-1 is a summary of reactions occurring on the surface of the metal. The horizontal short-dashed line represents the aqueous boundary situated in close proximity, at a distance, $y=\delta_s$, to the steel surface. The steel surface is represented by a distance, $y=0$. CO₂ present in the oil and gas stream exerts a partial pressure, CO₂ (P_{CO_2}), leading to its dissolution in the aqueous phase, CO₂ (aq). Its reaction with water leads to the formation of carbonic acid (H₂CO₃). Carbonic acid eventually undergoes dissociation into the bicarbonate ion (HCO₃⁻) and hydrogen ion. The bicarbonate ion undergoes further dissociation into the carbonate ion and hydrogen ion. Meanwhile, steel undergoes oxidation into ferrous ions (Fe²⁺). The eventual reaction between ferrous ions and the carbonate ions results in the formation of ferrous carbonate (FeCO₃).

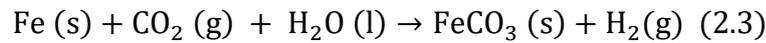
Also, a possibility in the corroding system is the presence of O₂ (Song et al., 2005). This exerts a partial pressure, O₂ (P_{O_2}), leading to its dissolution in the aqueous phase O₂ (aq). Aqueous O₂ then gets oxidised in the presence of water, forming hydroxide ions (OH⁻). The consequent reaction between the aforementioned ferrous ions and hydroxide ions results in the formation of ferrous hydroxide, Fe(OH)₂.

It is often the case that the corrosion of iron is generally summarised as follows (Crolet et al., 1999):



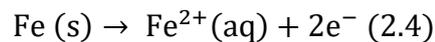
The CO₂ corrosion mechanism is a series of electrochemical reactions that involve the anodic dissolution of iron and the cathodic evolution of hydrogen. (Nesic and Vrhovac, 1999)

The overall reaction is:



The anodic dissolution of iron leads to the formation of various films such as FeCO₃ (siderite), Fe₃O₄ (magnetite) and Fe₃C (cementite) which can be protective or non-protective depending on film-formation conditions and the type of film formed.

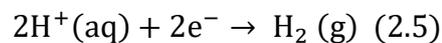
The anodic dissolution of iron is outlined as follows:



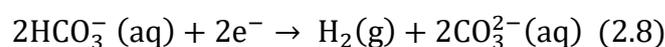
For the cathodic reactions, it has been assumed that the presence of CO₂ increases the rate of hydrogen evolution reaction. For strong acids that are known to dissociate fully, hydrogen ions will be evolved readily in solution however since the CO₂ corrosion mechanism is mass transfer limited therefore hydrogen ion evolution cannot exceed the rate at which it is transported to the metal surface from bulk solution. (Nesic and Vrhovac, 1999)

In solutions of pH>4, the mass transfer controlled current is small but the presence of the weak acid, H₂CO₃, which is the hydrated form of CO₂, enables hydrogen evolution at a much higher rate. Hence at any given pH, the presence of CO₂ leads to a much higher corrosion rate than would be found in a solution of a strong acid. Carbonic acid may be reduced at the cathode leading to the evolution of hydrogen gas. In any case, both reactions are said to proceed independently of each other (Gray et al., 1989). Another cathodic reaction is the reduction of water (Tanupabrungsun et al., 2013).

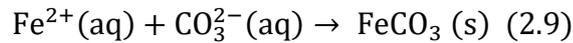
The cathodic evolution of hydrogen is outlined as follows:



Also it has been suggested that at high pH conditions in CO₂-rich solutions, that the reduction of the bicarbonate ion becomes important:



The following reaction summarises film formation:



2.3 Types of CO₂ Corrosion Damage

Carbon dioxide corrosion often takes up the form of general (uniform) corrosion, which is characterised by an electrochemical reaction proceeding uniformly over the entire surface area of the metal. It may also take three forms of localised attack – pitting, mesa-type attack and flow-induced corrosion. It is also worth noting that combined CO₂-erosion corrosion is characterised by the following physical descriptions: horse shoe, ripple effect marks, comet tails and dinosaur footprints, (Crolet, 1994; Kermani and Smith, 1994) while the previously mentioned forms apply strictly to pure CO₂ corrosion, and are the subject of this discussion.

2.3.1 Pitting

This is a form of localised corrosion that usually occurs in low-velocity fluid flow environments around dew point temperatures in a gas-producing well (Kermani and Morshed, 2003). It results in the appearance of holes in the metal surface, as seen in Figures 2-2 A and 2-2B.



Figure 2-2 Pitting corrosion of a steel pipe (Newman, 2010)

Figure 2-2A is a close-up view of pipe steel that has undergone pitting corrosion and Figure 2-2B shows the pit size on the steel surface in relation to the rest of the pipe. Pitting corrosion is difficult to predict and may give the appearance of small holes on the steel surface and is known to develop under the surface of the metal triggering mechanical failures by fatigue, as seen in Figure 2-3 (Newman, 2010).

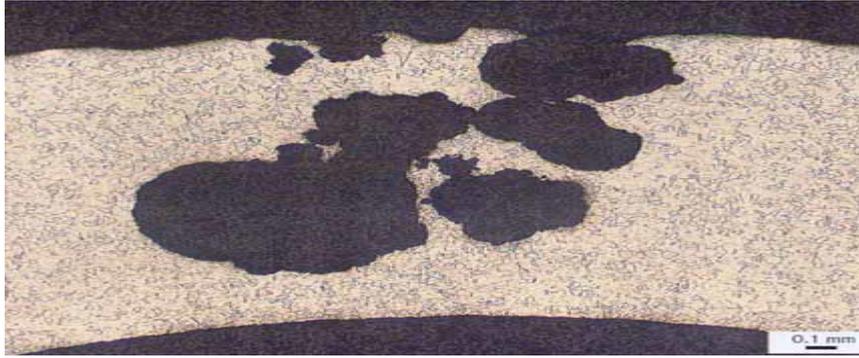


Figure 2-3 Pitting corrosion of a steel pipe (Newman, 2010)

Pitting susceptibility increases with increasing temperature and CO_2 partial pressure. It is also reported that nearly all metals and alloys regardless of type are likely to undergo pitting corrosion provided the conditions are right. While some researchers have pointed out that chloride ions are not a necessary component for propagating pitting corrosion (Videm and Dugstad, 1989) and conversely, the addition of lead (Pb) inhibited pitting corrosion through deposition at local anodes (Schmitt and Feinen, 1983; Schmitt and Engels, 1988); there is generally no applicable rule for the prediction of this type of corrosion since its propagation has been put down to its dependency on various factors.

2.3.2 Mesa-type Attack

This is a type of localised corrosion that takes place in low to medium fluid-flow environments as shown in Figure 2-4, where the protective film forms, however it is unstable and unable to withstand the intensity of the operating regime. Low to medium velocity fluid flow is still capable of washing away any protective films formed on the metal surface.

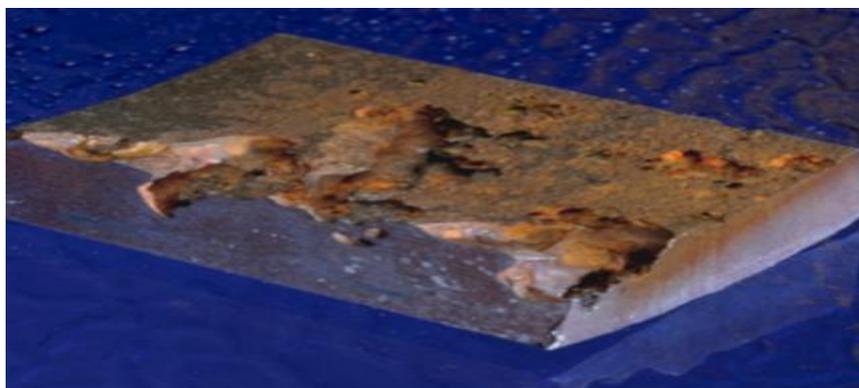


Figure 2-4. Mesa-type corrosion in a CO_2 -rich system (Kermani and Morshed, 2003)

In terms of physical appearance, typically, mesa-type attacks manifest in large flat-bottom basin-like depressions with sharp edges – corrosion damage showing in areas of localised attack is well in excess of surrounding areas. It was proposed by (Crolet et al., 1996) that the microstructurally formed galvanic coupling between the ferrite phase of steel and the cementite layer (Fe_3C) is a possible cause to promote mesa-attack in sweet environments. It is likely to occur in mature gas wells and young wells provided that the acid gas pressures are high in the latter. It was also mentioned by the same author, that while this form of corrosion is a little sensitive to fluid flow velocities, it is more dependent on the fluid composition.

Although a study performed by Ikeda and others (Ikeda et al., 1984) attributed the initiation of mesa-attack to the competitive film formation reactions between Fe_3O_4 and ferrous carbonate (FeCO_3), the presence of magnetite scale has not been detected in actual field conditions (Crolet, 1994; Crolet, 2002). It was concluded that the initiation and propagation of mesa-attack corrosion is due to inadequate protection offered by ferrous carbonate film formation on the metal surface (Crolet, 1994; Crolet, 2002). However there is still no definitive understanding of the way in which mesa-attack occurs and the precise nature of prevailing conditions required for its propagation is uncertain hence further systematic studies are necessary to prevent future occurrence in the field.

2.3.3 Flow-induced Localised Corrosion

This form of corrosion usually starts from pits/troughs that had previously been sites of localised mesa attack (Kermani and Morshed, 2003). This form of corrosion is solely dependent on high fluid flowrates.



Figure 2-5. Flow-induced localised pipeline corrosion

<https://co2corrosionchem409.wikispaces.com/Background+of+CO2+Corrosion>

High velocity fluid flow proceeds to generate turbulent eddies around mesa-attack sites, sweeping away any films formed as well as scale growths (Schmitt and Feinen, 1983),

(Schmitt and Engels, 1988). Once the scales are damaged or destroyed, the flow velocity then prevents the re-formation of protective film thereby leaving fluids to corrosively attack the exposed metal surface.

2.4 Major Factors Influencing CO₂ Corrosion

A number of key factors play a role in influencing the extent of CO₂ corrosion. These parameters are broadly classified into the following categories as shown in Figure 2-6 below:

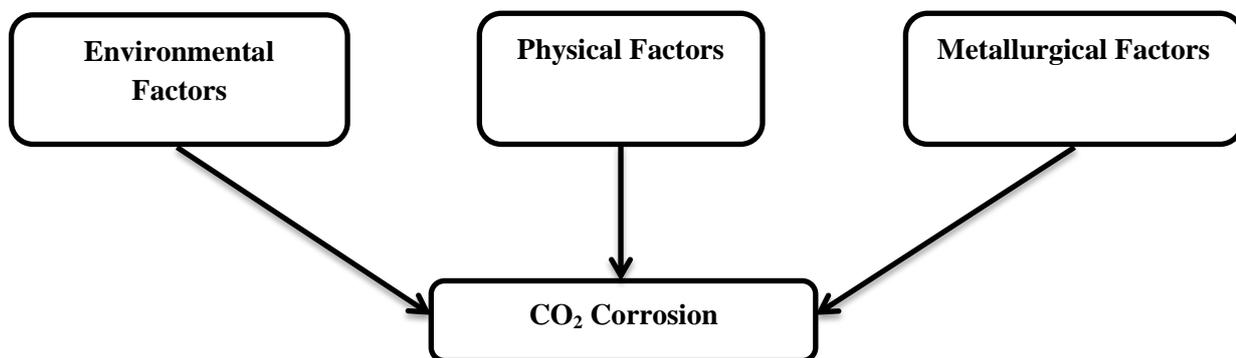


Figure 2-6. Major factors affecting CO₂ corrosion

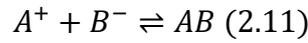
2.4.1 The Effect of Environmental Factors on CO₂ Corrosion

These are factors that affect the inherent corrosivity of the aqueous phase and as such influence carbon dioxide corrosion. These factors include solution chemistry, CO₂ partial pressure (mol% CO₂), temperature, in-situ pH, and the presence of hydrogen sulphide (H₂S) as well as the effect of organic acids such as acetic acid (CH₃COOH, HAc) (Kermani and Morshed, 2003). Some of which will be discussed in other sections of this review. In-situ pH will be discussed in this Section, in presence of H₂S.

Effect of Solution Chemistry

Solution chemistry deals with the relative concentration of dissolved ions with respect to each other, in the aqueous solution and how this affects the pH. While it will be discussed in the next Section that the pH of the solution is mostly influenced by the CO₂ partial pressure, solution chemistry is also very important given that it controls the formation and stability of protective film layers. Supersaturation, *Sat*, is defined as the ratio of the product of the ionic (cation and anion) concentrations to the solubility limit, K_{sp} :

$$Sat = \frac{[A^+][B^-]}{K_{sp}} \quad (2.10)$$



Where:

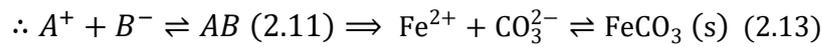
Sat is supersaturation and is dimensionless

A^+ and B^- are ionic species in ' $\frac{\text{mol}}{\text{l}}$ ', that form an insoluble salt (AB)

K_{sp} is the solubility limit in ' $\left(\frac{\text{mol}}{\text{l}}\right)^2$ ',

For CO_2 corrosion, A^+ and B^- are Fe^{2+} and CO_3^{2-} , respectively. Thus:

$$\therefore Sat = \frac{[A^+][B^-]}{K_{sp}} \quad (2.10) \Rightarrow Sat_{\text{FeCO}_3} = \frac{[\text{Fe}^{2+}][\text{CO}_3^{2-}]}{K_{sp_{\text{FeCO}_3}}} \quad (2.12)$$



High supersaturation of Fe^{2+} and CO_3^{2-} ions leads to the precipitation of FeCO_3 on the metal surface as a film layer, once the ferrous carbonate solubility limit (K_{sp}) is exceeded, which leads to a consequent reduction in corrosion rate by providing an extended diffusion length between the metal surface and corrosive medium, as seen in the Corrosion rate vs. pH plot of Figure 2-7B (Kermani and Morshed 2003). Low Fe^{2+} concentrations in solution has the effect of reducing corrosion rates but at a much slower rate because the rate of formation of FeCO_3 scale is much slower, as seen in the Corrosion Rate vs. pH plot of Figure 2-7B.

Effect of Temperature, CO_2 Partial Pressure and Fluid Flowrate Velocities

Temperature has the effect of increasing corrosion rates and this is due to the reasoning behind the Arrhenius and Kinetic theories, as seen in Figure 2-7C (Tan and Chan, 2011). Higher temperatures provide molecules with greater amounts of energy, thus enabling faster collisions between reacting species and speeding up reaction rates. The higher the CO_2 partial pressure, in general, the greater the corrosion risk, as this directly translates to higher concentrations of carbonic acid and lower pH-values of the solution surrounding the metal surface, as seen in Figure 2-7A. Also, higher fluid flowrate velocities imply greater corrosion risks due to the mechanical washing away of the protective film on the metal surface. A more detailed explanation of how these factors affect corrosion rate is described in the next Section.

The following illustrations depict the general trend for some environmental factors on CO₂ corrosion.

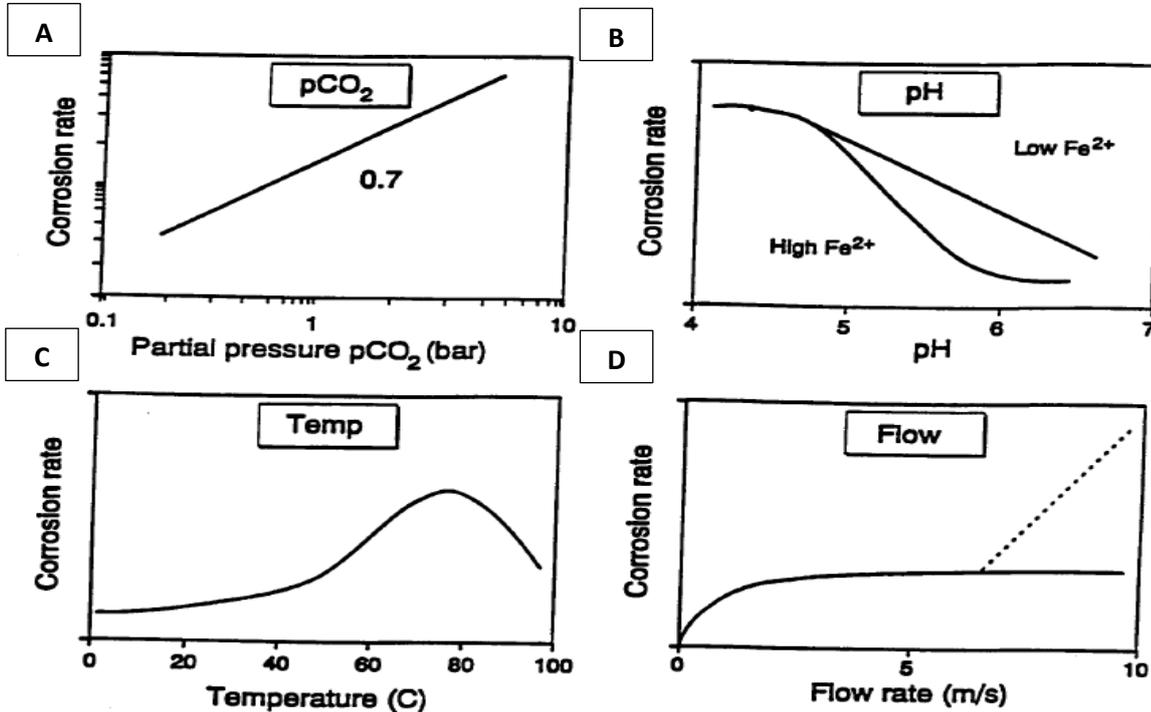


Figure 2-7. Plot illustrations indicating the effect of the primary parameters on CO₂ corrosion of carbon steel (Dugstad et al., 1994a)

Figure 2-7A shows a linear relationship between CO₂ partial pressure to an exponent of 0.7 and corrosion rate (Dugstad et al., 1994a). An increase in CO₂ partial pressures results in incremental rates of corrosion. Figure 2-7B shows a near-linear decrease in corrosion rate with increasing pH. Corrosion rate however decreases more slowly for low dissolved ferrous ion concentrations than for higher ferrous ion concentrations, due to supersaturation occurring more readily for the latter than the former thus forming a protective scale sooner on the steel surface. Figure 2-7C shows an increase in temperature resulting in a corresponding increase in corrosion rates albeit a peak occurs at the 60-80°C range due to protective magnetite scale formation, which subsequently leads to a fall in corrosivity. Figure 2-7D shows an initial increase in fluid flow velocity leads to an increase in corrosion rates, after which increasing fluid flow velocity leads to a levelling-off of corrosivity. For very high fluid velocities exceeding 7m/s, there is the possibility of a further increase in corrosion rate, particularly for elevated temperatures (Eriksrud and Søntvedt, 1983), due to the mechanical removal of surface film (Dugstad et al., 1994a).

CO₂ Corrosion Prediction – Corrosion Rate-Determining Parameters

The impact of CO₂ corrosion on the oil and gas industry as well as the CCS industries is well documented. (Kermani and Morshed, 2003; IEAGHG, 2010). In the oil and gas industry, carbon dioxide is present as a sub-component of petroleum fluids or from enhanced CO₂ recovery processes while for CCS applications, CO₂ is the primary component containing small quantities of contaminants such as SO_x, NO_x and water.

For both processes, CO₂ corrosion is induced by the dissolution of the gas in an aqueous medium coupled with the initiation of carbonic acid equilibria; and is dependent on the CO₂ mole fraction or percentage. In general, CO₂ corrosion will increase in magnitude, the greater its mole fraction within the oil and gas stream, likewise this rule is applicable to CCS processes albeit the higher CO₂ mole fractions and presence of acid gases potentially implies greater corrosion rates are to be expected for the latter. Other factors intrinsic to petroleum fluids such as operating pressures, temperatures and pH also have an influence on CO₂ corrosion rates as shown in Table 2-1.

Table 2-1. Estimated corrosion rates of carbon-manganese steels for some fields in the Norwegian sector of the North Sea (Dugstad et al., 1994a)

Field Operator	Type	CO ₂ content (mol %)	Pressure (bar)	Max. temperature (°C)	Calculated pH	Estimated corr. rate (mm/year)
Tommeliten Statoil	Oil/Gas	3	235	90	6.1	11 (90°C) 15 (67°C)
Lille-Frigg Elf Aquitaine	Gas	2.4	440	80	Sat.*	13 (80°C) 15 (68°C)
TOGI Norsk Hydro	Gas	0.2	100	55	Sat.*	5.1
Sleipner West Statoil	Gas/oil	9	150	90	6.8	12 (90°C) 19 (56°C)

Corrosion rates are calculated by using the 1991 De Waard correlation (De Waard et al., 1991)

* The pH at FeCO₃ saturation has been used in the corrosion rate calculation

From Table 2-1, it is observed that though CO₂ mole content is the primary determinant of the magnitude of predicted corrosion rates, pressures, temperatures and pH also contribute hence these are utilised in CO₂ corrosion prediction modelling. The 1991 De Waard model is described in greater detail in Section 3.3. The high CO₂ mole content (9%) of the Sleipner West field implies that it has the greatest predicted CO₂ corrosion rates (12 and 19mm/year) of all the fields.

It must be noted that while these parameters are the main causes of corrosion, small variations in film formation layers and corrosion product layers (of the aqueous phase) also play a part. For instance, in a scenario when all primary parameters are kept constant: temperature, partial pressure, fluid velocity and pH, while the amount of corrosion product in the water phase is varied, relatively high corrosion rates can be expected. (Dugstad et al., 1994a). The reason for these high corrosion rates is attributed to changes in the properties of the thin layer of corrosion products and scale which accumulate on the surface of the metal. The morphology and composition of the film layer determine the type of corrosive attack – worst-case, low corrosion under protective film or localised corrosion (mesa/pitting) (Dugstad et al., 1994a). It is also said that the film layers can interact with the transport of corrosion inhibitors to the surface of the metal thereby controlling inhibitor performance (Dugstad et al., 1994a).

In ‘worst-case’ (nonprotective film) corrosion, the presence of ferrous ions (Fe²⁺) ions in solution leads to the formation of iron carbide film, Fe₃C. This film deposits on the metal surface, un-corroded and due to its naturally porous nature, offers no protection to the metal surface underneath as seen in Figure 2-8, for the nonprotective Fe₃C or Fe₃C-FeCO₃ films (Dugstad et al., 1994a).

In most cases, it enhances CO₂ corrosion as a result of it having a greater overpotential than iron. It sets up a galvanic contact between itself and iron and this accelerates cathodic reactions that lead directly to the anodic dissolution of the metal in the presence of << 1ppm Fe²⁺ (Dugstad, 1998).

It may also lead to increased anodic dissolution of the metal by enhancing local acidification around the metal surface – since cathodic reactions occur preferentially at iron carbide sites, the aqueous phases in these regions become more alkaline as they are separated from the metal. The net effect is a change in the water composition where cathodic regions become more alkaline and anodic ones become more acidic leading to increased dissolution rates of the metal by internal localised acidification. (Crolet, 1994; Dugstad, 1998).

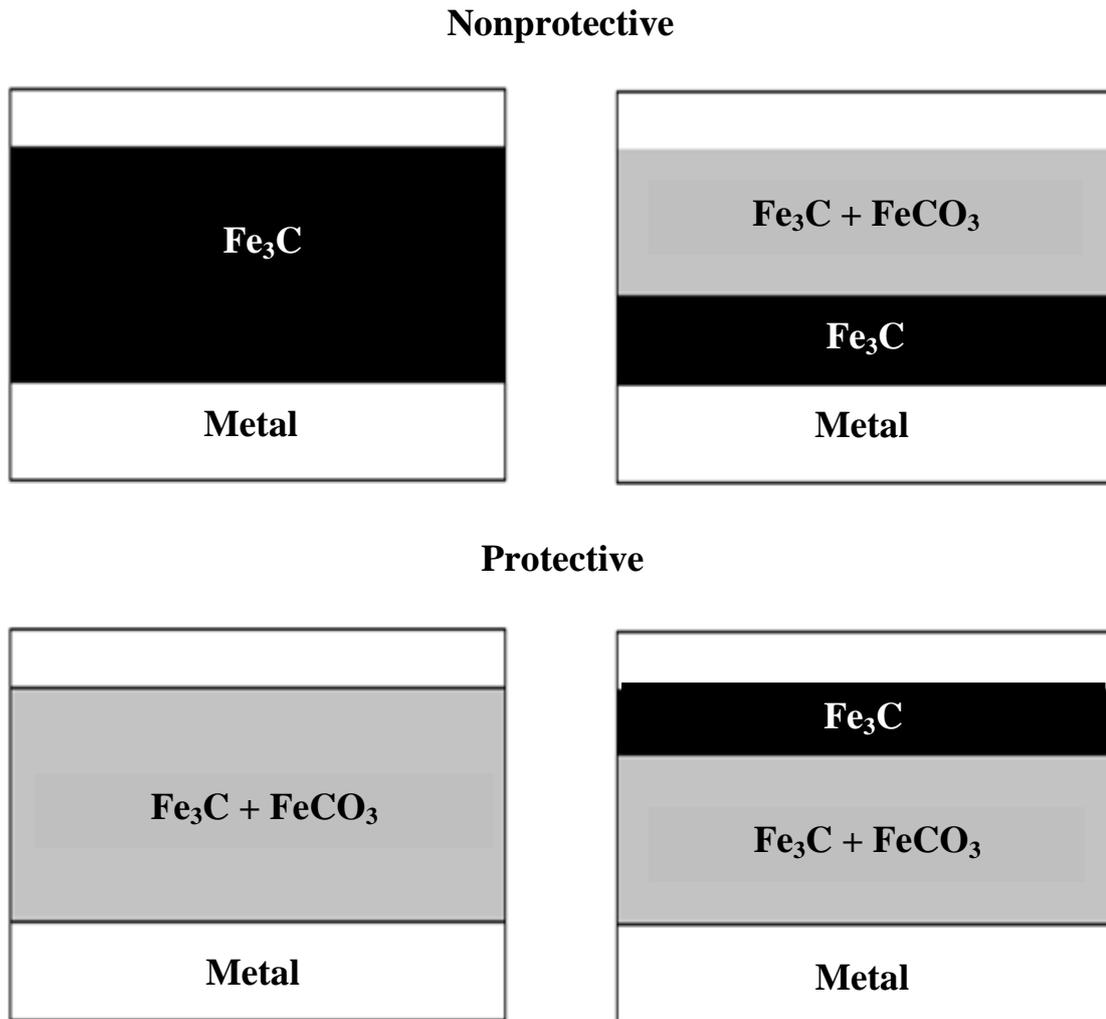


Figure 2-8. Morphologies for protective and nonprotective scale formation (Crolet et al., 1996, Kermani and Morshed, 2003)

It is worth noting that the FeCO_3 and Fe_3C films are the most commonly observed surface films on carbon and low-alloy steels in CO_2 -containing environments (Kermani and Morshed, 2003). This film is protective when the FeCO_3 phase covers the metal surface, effectively sealing it completely as seen for the protective film in Figure 2-8, or if it is integrated within the Fe_3C phase.

Presence of H_2S

$\text{CO}_2/\text{H}_2\text{S}$ corrosion is widely encountered in the oil and gas industry. Hydrogen sulphide corrosion, often termed as sour corrosion, occurs in fields whose oil and gas streams have some amount of hydrogen sulphide satisfying the following condition: $P_{\text{CO}_2}/P_{\text{H}_2\text{S}} < 200$ (Yap

and Srinivasian, 2010). This is the condition for corrosive effect as stated in Table 2-2. It is suggested that the simultaneous presence of both acid gases leads to a very aggressive environment that can cause severe corrosion of steels but ultimately, the extent of corrosivity is dependent on the aforementioned relative partial pressures of each gas with respect to each other (Yap and Srinivasian, 2010).

In general, the higher the partial pressure of H₂S, the lower the pH of the system since it is an acid gas. When the system temperature, H₂S and CO₂ partial pressures are constant, the pH increases with time, as the hydrogen ion concentration falls in the aqueous phase.

The presence of CO₂ and H₂S in bulk solution greatly influences the solution pH however it should be noted that while pH is an important parameter for the estimation of corrosion rates, the in-situ pH is of particular importance. The pH reported from results of water analyses is not in-situ as the water analyses are carried out after samples have been exposed to the atmosphere. This pH does not accurately describe the system (Yap and Srinivasian, 2010). In-situ pH is said to be determined from ionic modelling of the solution or can be calculated using the Brönsted concept as stated below (Yap and Srinivasian, 2010):

$$\text{Thus } \text{pH} = -\log a_{\text{H}^+} \Rightarrow \text{pH} = -\log \gamma_{\text{H}^+} m_{\text{H}^+} \quad (2.14)$$

Where:

a_{H^+} = activity of H⁺ species

γ_{H^+} = activity coefficient of H⁺ species

m_{H^+} = molar concentration of H⁺ species

By the Brönsted concept, a Brönsted acid is a proton donor while a Brönsted base is a proton acceptor (Naiman, 1948).

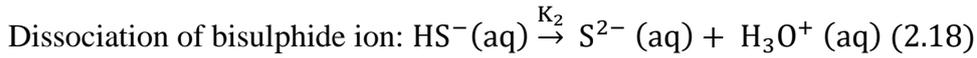
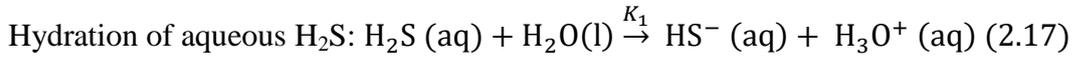
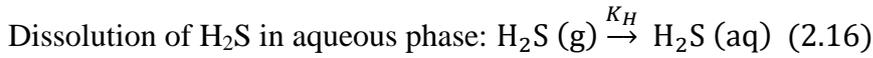
Determination of pH for a Mixed CO₂/H₂S-System

For systems that contain both CO₂ and H₂S, the following ionic balance has been developed by using the Brönsted concept:

$$\text{H}^+ = [\text{HCO}_3^-] + 2[\text{CO}_3^{2-}] + [\text{HS}^-] + 2[\text{S}^{2-}] + [\text{OH}^-] - \text{C}_{\text{HCO}_3}^- - 2\text{C}_{\text{HS}}^{2-} - \text{C}_{\text{CO}_3}^{2-} - 2\text{C}_{\text{S}^{2-}} \quad (2.15)$$

The species in square brackets represent equilibrium concentrations which can be determined by dissociation reactions for both CO₂ and H₂S. In this thesis, only the dissociation of H₂S is

shown. The following set of reactions summarise the dissolution of H₂S into the aqueous medium and its subsequent dissociation:



The equilibrium constant terms, K_H, K₁ and K₂ are then written and simplified as follows:

$$K_H = \frac{[\text{H}_2\text{S}]}{p\text{H}_2\text{S}} \quad (2.19)$$

Making [H₂S] the subject of the equation, thus [H₂S] = K_HpH₂S (2.20)

$$K_1 = \frac{[\text{HS}^-][\text{H}^+]}{[\text{H}_2\text{S}]} \quad (2.21)$$

Making [HS⁻] the subject of the equation, thus [HS⁻] = $\frac{K_1[\text{H}_2\text{S}]}{[\text{H}^+]}$ (2.22)

$$K_2 = \frac{[\text{S}^{2-}][\text{H}^+]}{[\text{HS}^-]} \quad (2.23)$$

Making [S²⁻] the subject of the equation, thus [S²⁻] = $\frac{K_2[\text{HS}^-]}{[\text{H}^+]}$ (2.24)

Substituting eqn. (2.16) in eqn. (2.18), we have $\Rightarrow [\text{HS}^-] = \frac{K_1 K_H p[\text{H}_2\text{S}]}{[\text{H}^+]}$ (2.25)

Substituting eqn. (2.25) in eqn. (2.24), we have $\Rightarrow [\text{S}^{2-}] = \frac{K_1 K_2 K_H p[\text{H}_2\text{S}]}{[\text{H}^+]^2}$ (2.26)

Once all the equilibrium constants are determined in this way, several measured parameters such as temperature, pressure, CO₂ and H₂S mole fraction, ionic strength, Henry's law constants and equilibrium constants in conjunction with an accurate water analysis are required to solve Equation. (2.11) (Yap and Srinivasian, 2010).

Conditions for H₂S Corrosion Risks and H₂S Corrosion Mechanism

The relative proportions of carbon dioxide to hydrogen sulphide determine the levels of corrosion risks associated with systems that contain a mixture of the two acid gases. In terms of corrosion assessment, H₂S has been discovered to have a three-fold role as summarised in

Table 2-2. In a similar way to the CO₂ corrosion mechanism, H₂S dissolves in the aqueous bulk solution to exhibit corrosive effects. Gaseous hydrogen sulphide exerts a partial pressure thus enabling it to dissolve in solution to form bisulphide ion (HS⁻). The bisulphide ion then undergoes further dissociation to form the sulphide ion (S²⁻), as shown as Equations. (2.16), (2.17) and (2.18). These equations constitute the cathodic reactions.

The anodic reactions are as follows:

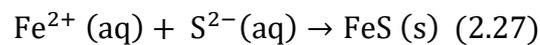
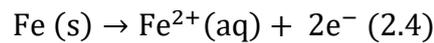


Table 2-2. Corrosion assessment in CO₂/H₂S systems is dictated by three conditions.

(Yap and Srinivasian, 2010)

	No Corrosive Effect	Mitigating Effect	Corrosive Effect
Conditions	Occurs when pH ₂ S < 0.01psia (6.895x10 ⁻⁵ MPa)	Occurs when <i>P</i> _{CO₂} / <i>P</i> _{H₂S} > 200psia (1.379MPa)	Occurs when <i>P</i> _{CO₂} / <i>P</i> _{H₂S} < 200psia (1.379MPa)
Outcome	No significant impact of H ₂ S on corrosion rate is observed For systems containing CO ₂ , the CO ₂ corrosion mechanism will proceed	Formation of FeS (mackinawite) scale at temperatures of 60°C Scales formed are meta-stable and preferentially form over FeCO ₃ scale Net effect is decrease in corrosion rates due to a reduced surface area exposed to attack	FeS scales dominate over FeCO ₃ since the mole content of H ₂ S exceeds that of CO ₂ Scales are meta-stable; when T<60°C and when T>120°C, scales become unstable, porous and offer little protection

Following the oxidation of iron to ferrous ions, the ferrous ions then react with sulphide ions to form iron sulphide scale (FeS). The reaction between ferrous ions and sulphide ions to form a thin layer of ferrous sulphide on the metal surface is highly pH and temperature-dependent and often mitigates corrosion. (Yap and Srinivasian, 2010)

Mechanics and Kinetics of Iron Sulphide Film/Scale Formation

In an attempt to comprehend the mechanics and kinetics of the formation of iron sulphide scale in mixed CO₂/H₂S systems for the accurate prediction of mixed CO₂/H₂S corrosion, experiments have been performed on glass cells to describe H₂S corrosion (Sun and Netic, 2007). Experimentation involved weight loss procedures in which hydrogen sulphide corrosion was measured using rectangular and cylindrical specimens of X65 pipeline steel. The retention rate of iron sulphide scales was also measured in these experiments.

A mechanistic model was developed based on results from the experiments. From the research findings, it was discovered that iron sulphide scale in the form of mackinawite is predominant over other forms of iron sulphide scale particularly at the initial stages of scale formation. Also, H₂S is observed to react with the metal surface directly by a 'solid-state' reaction which differs from the supersaturation-precipitation theory mode by which iron carbonate scale is formed in CO₂-rich environments. By 'solid-state' reaction, the implication is that there is still evidence that iron sulphide precipitates out of solution following supersaturation but in addition, a direct reaction between H₂S and the metal surface is the prevalent mode of combination. Furthermore, there are several scales formed in the H₂S environment in contrast to a CO₂-only environment. Some of these scales are: amorphous ferrous sulphide, mackinawite ((Fe, Ni)_{1+x}S [where x=0 to 0.11]), cubic ferrous sulphide, smythite (Fe_{3+x}S₄ [where x=0 to 0.3] or Fe₃S₄), greigite (Fe²⁺Fe³⁺₂S₄ or Fe₃S₄), pyrrhotite (Fe_{1-x}S), troilite (FeS) and pyrite (FeS₂) (Sun and Netic, 2007; James and Fleischer, 1966). These are all various forms of iron sulphide scale but it is worth noting that the actual number of these sulphides in existence is still a source of debate between mineralogists and thermodynamicists (Smith and Joosten, 2006). These scales are covered in more detail in Section 3.7.2 In addition, evidence from a preceding journal article: (Sun et al., 2006), supports the fact that the supersaturation-precipitation theory does not always hold for H₂S corrosion, in that, corrosion rates always exceed precipitation rates and the explanation is that iron sulphide scale is formed mainly by ferrous ions released from the metal surface by corrosion and not by ferrous ions present in bulk solution.

In terms of the kinetics of iron sulphide film formation, it is concluded that H₂S corrosion rates generally increase with increasing concentrations of H₂S however corrosion rates decrease with increasing reaction time. The reason for the decrease of corrosion rate with time is due to the formation of stable mackinawite scales whose thickness increases over time on the metal surface (Sun et al., 2006).

Species Transport through Boundary Layers

In the assumption that H_2S corrosion is mass-transfer controlled and similar to the mass-transfer mechanism for CO_2 corrosion, the rate of evolution of hydrogen ions cannot exceed the rate at which they are transported to the metal surface. Hence, the H_2S corrosion rate is dependent on the rate at which hydrogen ions are transported to the metal surface from the bulk solution as seen in Figure 2-9.

Both H_2S and H^+ species will travel to the metal surface from the bulk solution via convective diffusion (turbulent eddies) and then by molecular diffusion.

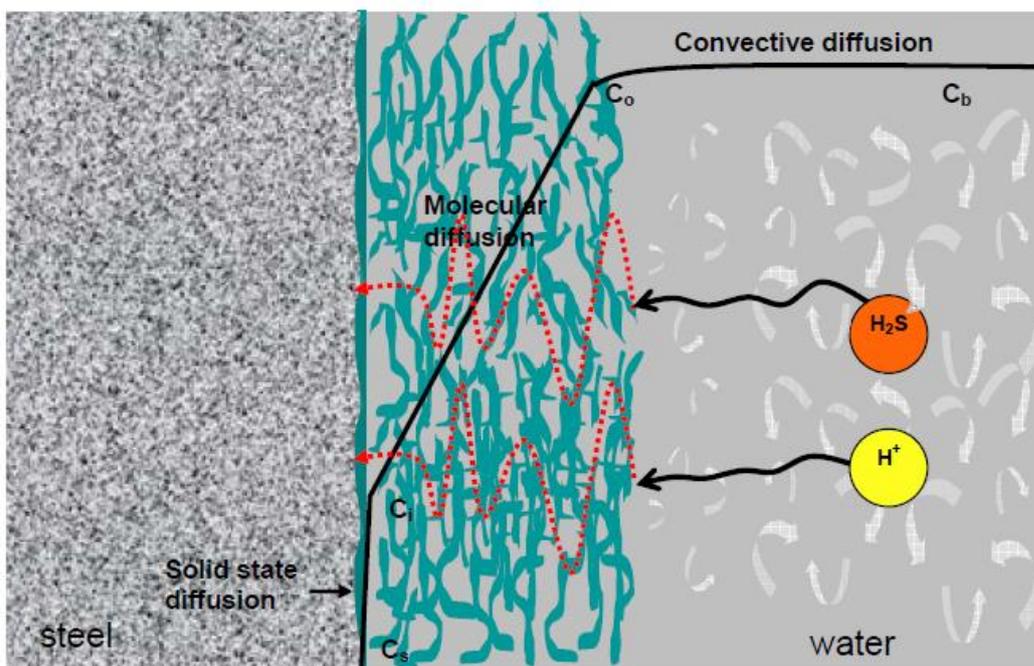


Figure 2-9. Schematic of H_2S corrosion – corrosion rate of steel in H_2S solutions is said to be under mass-transfer control. (Sun and Netic, 2007)

A Mathematical Model for the Prediction of H₂S Corrosion Rate

A number of assumptions were made in the development of the mathematical model. These assumptions are discussed here. Firstly, there is always a very thin but dense mackinawite film of about $\ll 1\mu\text{m}$ on the metal surface and it acts as solid state diffusion barrier for the sulphide species involved in corrosion. Secondly, film growth is cyclical going through growth, cracking and delamination stages. Also, the film (outer scale) continuously grows in thickness over time and lastly, the outer scale is layered, very porous and is loosely attached hence is prone to peeling and spalling (Sun and Netic, 2007).

These assumptions coupled with the findings from experimentation have led to the conclusion that using the concept of fluxes and the fact that the corrosion rate of steel in H₂S solutions is mass-transfer limited; at steady-state (equilibrium), all fluxes are equal.

$$CR_{H_2S} = A_{H_2S} e^{\frac{B_{H_2S}}{RT_k}} \ln \frac{c_{b,H_2S} - CR_{H_2S} \left(\frac{\delta_{0.5}}{D_{H_2S} \varepsilon \psi} + \frac{1}{k_{m,H_2S}} \right)}{c_{s,H_2S}} \quad (2.28)$$

Where:

CR_{H_2S} is the corrosion rate of steel due to H₂S in mm/year

A_{H_2S}, B_{H_2S} are the Arrhenius constants, $A_{H_2S} = 1.30 \times 10^{-4} \frac{\text{mol}}{\text{m}^2\text{s}}$ and $B_{H_2S} = 15500 \text{ J/mol}$

c_{s,H_2S} is the concentration of H₂S on the steel surface and is set to 1.00×10^{-7} in mol/m³

c_{b,H_2S} is the bulk concentration of H₂S in the liquid phase in mol/m³

δ_{os} is the thickness of the mackinawite scale $\Rightarrow \delta_{os} = m_{os}/(\rho_{FeS}A_{os})$ in m

D_{H_2S} is the diffusion coefficient for dissolved H₂S in water, $D_{H_2S} = 2.00 \times 10^{-9}$, in m²/s

k_{m,H_2S} is the mass transfer coefficient for H₂S in the hydrodynamic boundary layer, $k_{m,H_2S} = 1.00 \times 10^{-4}$ in nearly stagnant condition, in m/s

ε is the outer mackinawite scale porosity

ψ is the outer mackinawite scale tortuosity factor

T_k is the temperature in Kelvin

R is the molar gas constant

This equation is non-linear with respect to CR_{H_2S} but can be solved by the application of appropriate numerical techniques.

2.4.2 The Effect of Physical Factors on CO₂ Corrosion

These are also very important factors that include water-wetting, the presence of wax and wax deposition, corrosion film characteristics, the effect of crude oil and erosion.

Effect of Crude Oil

While experiments have been carried out to determine the magnitudes of corrosion of associated steel pipelines in brine environments, it is not the same as carrying out these experiments in the presence of the particular crude oil and brine. Differences are likely to be present in measurements recorded for the two instances and these have often led to gross errors when using corrosion test results to predict corrosion scenarios in the field.

Crude oil composition is therefore said to have an effect on CO₂ corrosion. Experimental research provided supporting evidence of this, which also showed that more sophisticated analyses of the data presented in the study were required such as the use of artificial neural networks (ANNs) (Hernandez et al., 2006). The article suggests the fact that though interfacial tension and fluid dynamics play an important role in CO₂ corrosion, the role of the given crude oil is also significant, in that the composition of the latter affects interfacial tension. (Hernandez et al., 2006). It is also suggested that wettability is strongly affected by the presence of surface active compounds and these are believed to be polar molecules containing oxygen (O), nitrogen (N) and sulphur (S) molecules.

The heavier the fraction of the crude, the greater the number of oxygen, nitrogen and sulphur molecules because such crudes contain asphaltene and resins. In other words, the polar molecules, O, N and S, in these crudes are as a result of the presence of polarisable compounds, asphaltenes and resins (Hernandez et al., 2006). Polarizable compounds and polar molecules thus exert a synergistic effect which changes the wettability of crude oil by reducing the interfacial tension between oil and water therefore causing the system to be more oil-wet. The net effect is a decreased dissolution rate of the metal at its surface because

corrosion of the metal proceeds at high rates when the metal surface is water-wet as opposed to when it is oil-wet.

Effect of Flow and Erosion

It is generally accepted that the higher the fluid flowrates flowing through a segment of pipe, the greater the associated corrosion risks as seen in the corrosion rate vs flow rate plot of Figure 2-7D. Therefore corrosion test results are greater for high fluid flow than for very low fluid flow. However, the effect of flow is still a contentious area in CO₂ corrosion prediction (Hernandez et al., 2006). Flow regimes influence corrosion risk but there is very little experimentation to accurately quantify their contribution to flow-induced CO₂ corrosion.

The prevailing flow regimes in a segment of pipe will depend on the fluid velocity through the pipe, pipe orientation and geometry, pipe inclination and length as well as temperature and pressure conditions. Once any of the aforementioned variables undergo a slight change along the length of the pipe segment, then this change induces a change in flow regime.

Typical flow regimes found in oil and gas production facilities include stratified, wavy stratified, rolling wave, plug flow and annular (Bondos et al., 2007). The significance of knowledge of the flow regime, particularly of mixed phase fluids, is the fact that the type of wetting occurring in any given scenario is easily determined. The key factors here are oil/water ratio, emulsion tendency/stability and water cut percentages (Hernandez et al., 2006). When water cut percentages are greater than 30%, water then becomes the continuous phase and there are higher corrosion risks (Kermani and Morshed, 2003).

Erosion occurs when solid particles that may range in size such as sand accompany fluid flow thus creating an abrasive effect on the metal surface (Giourntas et al., 2015). Erosion-corrosion is a form of tribo-corrosion material loss mechanism caused by flowing fluid (in the presence of solid particles) damaging both the surface layers such as the passive film/corrosion products and the base metal (Hu et al., 2011). It involves electrochemical corrosion processes and mechanical wear (Hodgkiess et al., 1999). This form of corrosion is gaining greater attention not only because of its destructive nature but also this regime of degradation has a greater likelihood of occurring particularly for old oil wells known to produce high levels of sand in produced fluids.

Experimentation is often very useful when erosion occurs as a result of particulate fluid flow. Currently, there are no industry guidelines that assess and control erosional corrosion caused

by particulate fluid flow however the commonly cited equation formulated by the American Petroleum Institute (API): API RP-14E is used as a resource tool for evaluating pure gas-in-liquid-erosion and is stated as follows (API, 1981):

$$V_e = C / \sqrt{\rho_m} \quad (2.29)$$

Where:

V_e = Mixed fluid velocity (of gas and liquid) in m/s or ft/s

C = Constant (specific to given material)

ρ_m = Mixed fluid density (of gas and liquid) in kg/m³ or lb/ft³

This equation is empirical and it is more often the case that any judgements made using the equation in predicting corrosion risks are by the operator's personal experience or discretion.

Effect of Organic Acid

The presence of organic acids in CO₂-containing environments is known to influence and complement CO₂ corrosion of pipelines. In test simulations, the addition of acetic acid (CH₃COOH), reduces the protectiveness of films on the metal surface and makes it more susceptible to mesa-type corrosive attacks (Crolet, 1994; Crolet, 2002). Generally, the presence of CH₃COOH causes a significant increase in corrosion rates in CO₂ environments and is known to take over as the main source of corrosivity even in conditions where CO₂ partial pressures are considerably low. In some cases where traces of CH₃COOH are freely-occurring, iron acetate scales are more prevalent than iron carbonate scales (Crolet et al., 1999). Organic acid corrosion is further discussed in Section 3.8.

2.4.3 The Effect of Metallurgical Factors on CO₂ Corrosion

The key factor here is the chemical composition of the alloy in use. Heat treatment and microstructural features also play important roles on corrosion of carbon steels in CO₂ environments. While it is generally accepted that the presence of chromium (Cr) in steel alloys is beneficial in terms of improving corrosion resistance, the optimum amounts of chromium needed varies with the environment. It must also be borne in mind that the greater the fractional amounts of chromium in these steel alloys, the greater the costs are in its manufacture and its eventual installation.

Molybdenum (Mo) has also been found to improve the corrosion-resistant characteristics of steel alloys and while the addition of copper (Cu) also gives the alloys similar beneficiary traits; it tends to reduce the effectiveness of corrosion inhibitors (Gulbrandsen and Nyborg, 2000). In general, the addition of small quantities of copper, nickel, chromium and possibly molybdenum together have been found to improve corrosion-resistance of carbon steels (Videm and Dugstad, 1989).

Further Metallurgical Considerations - Alloying Elements

Substantial laboratory work has been carried out to systematically test the corrosion-resistance of low-alloy steels. Research publications by (Kermani et al., 2001) reveal that corrosion-resistance is enhanced by the application of the following principles:

- I. Lowering carbon (C) and adding carbide-forming alloying elements to maximise the effect of any subsequent addition of chromium (Cr) and Molybdenum (Mo) by ensuring that they remain in solid solution.
- II. Achieving the desired properties by micro-alloying additions and mechanical and heat treatments.

In microalloying, steel compositions are designed with low carbon content containing microalloying elements such as titanium (Ti), niobium (Nb) and vanadium – stronger carbide-forming metals. The reason for micro-alloying is that the aforementioned metals will combine with carbon (in the given steel) preferentially thus leaving chromium and nickel uncombined in the ferrite thereby enhancing corrosion resistance of the alloy.

Additionally, while the removal of carbon from these low-alloy steels is likely to reduce their overall strength, the addition of silicon and nickel is used to restore the strength caused by the removal of carbon. Extensive metallurgical studies have led to confirmed results in terms of the degree of corrosion-resistance of each alloying element. A comparative trend is shown in Figure 2-10.

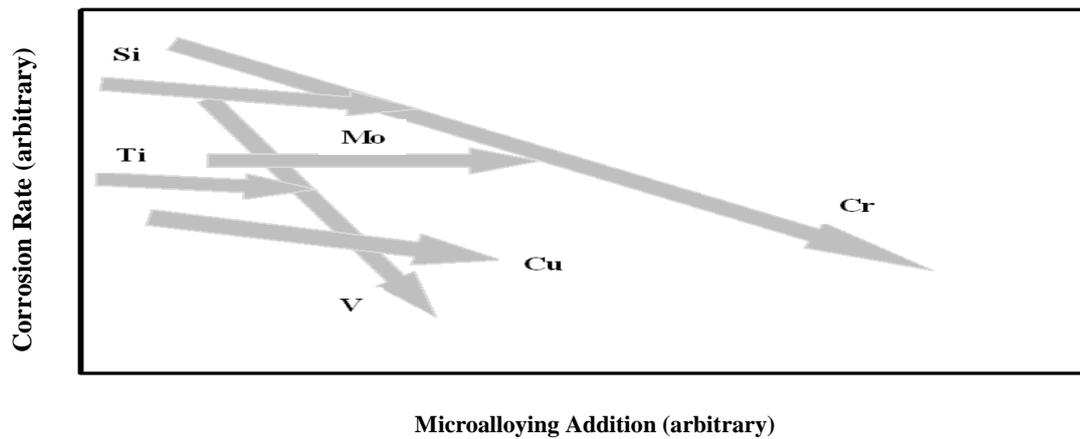


Figure 2-10. Plot showing comparative trend of elemental contributions to corrosion-resistance (Kermani et al., 2001), (Kermani et al., 2003)

The Influence of Chromium

The addition of chromium has a beneficial effect on the corrosion performance of low-alloy steels as seen in Table 2-3 and Figure 2-11.

The following table categorises the effect of chromium in terms of addition percentages.

Table 2-3. Chromium-content categories and expected corrosion performance levels

(Kermani et al., 2001; Kermani et al., 2003)

Cr-content (%)	Description
5	Offers lowest corrosion rates
3	Offers a 10-time reduction in corrosion rates
1.5	Insufficient to provide reliable corrosion performance
< 1.5	Generally not satisfactory for CO ₂ corrosion performance Highest corrosion rates are observed at 0.02% Cr

The following figure shows a schematic of CO₂ corrosion performance for each of the chromium-content categories discussed in Table 2-3.

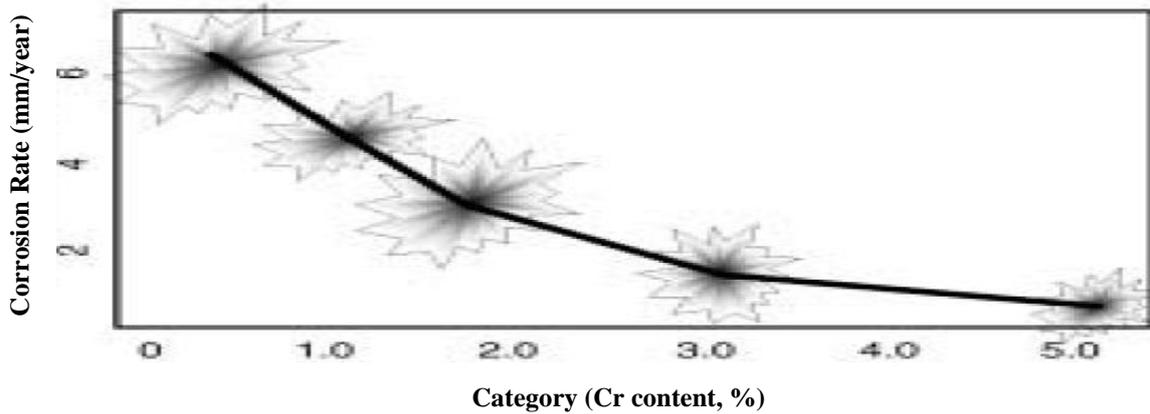


Figure 2-11. Plot showing the category of chromium content of steel and corresponding corrosion rate (Kermani et al., 2001; Kermani et al., 2003)

While optimum levels for the addition of chromium is still undetermined, a level between 2 to 3% Cr was considered essential to achieve good corrosion performance. It is also concluded that Vanadium-microalloyed steel containing Cr, Si, Mo and Cu is the most promising combination in terms of producing alloys with the desired features of strength and corrosion resistance (Kermani et al., 2001; Grau, 2000).

2.5 Established CO₂ Corrosion Models

Many research institutions and petroleum companies have developed CO₂ corrosion models over the years. Some of the oil companies that have contributed to the understanding of CO₂ corrosion through the development of pipeline corrosion models include: Shell, Statoil, Total and BP (Nyborg, 2010). In addition some research institutes have developed pipeline CO₂ corrosion models and these are: the Institute for Energy Technology (IFE), OLI Systems and Ohio University (Nyborg, 2010).

The models discussed in this Section include the HydroCor, Corpus, KSC and Multicorp models. Other models such as the De Waard, Norsok M-506 and Freecorp Models are discussed in detail in Chapter 3.

2.5.1 Hydrocor Model

This model was developed by Shell to combine corrosion and fluid flow modelling and is their preferred tool for CO₂ corrosion prediction in pipelines. Oil-wetting is assumed for flow velocities in excess of 1.5m/s and water cuts less than 40% implying there is no corrosion risk (Pots et al., 2002; Nyborg 2002; Nyborg 2010). Greater water cut percentages would change the wettability simulation conditions from oil-wet to water-wet hence a greater corrosion risk. The model assumes scale formation for condensed water but not for formation water due to scale instability. Porous mixture scales may form with little protection offered to the metal surface.

It should be noted that a separate CO₂ corrosion model is used to generate a corrosion rate profile for pipelines with multi-phase flow. Due to this fact, pH calculation is not included in the Hydrocor model it therefore does not account for iron and bicarbonate concentrations.

2.5.2 Corplus Model

Corplus is Total's own corrosion model and is a result of the merger between Elf's Cormed and Total's Lipucor (LI) models (Crolet and Bonis, 1991; Nyborg, 2010). It has a fluid flow model and a CO₂ potential corrosivity index. The latter is based on the vast experience of Total's engineers gained over the years of service in the field. The scale uses terminology like: very high, low, and so on, to classify the severity of CO₂ potential corrosivity.

It also has a separate model for computing corrosion rates. In general the Corpus model relies on huge amounts of data to work out corrosion rates. The full CO₂ corrosion description in this case comprises results from the CO₂ potential corrosivity index and the corrosion rate values. It must be noted that the model does not account for any instances of oil-wetting. Model operation is based on detailed water chemistry analyses including the effects of CO₂, organic acids and calcium.

It includes a facility for correcting calcium carbonate (CaCO₃) concentration. Where user-specified CaCO₃ concentrations are likely to lead to CaCO₃ super-saturation, the user is warned to check the calcium carbonate concentration to reduce its magnitude. If this message is ignored by the user, the program then proceeds to correct this by calculation of a pH of lower magnitude (Nyborg, 2010).

2.5.3 KSC Model

This model is mechanistic, so offers insight and thorough descriptions of the occurring physical phenomena in CO₂ corrosion. It was developed at IFE, Norway, and is based on the combination of an electrochemical and a transport model (Nesic et al., 1996; Nyborg, 2002).

The model simulates chemical reactions in the aqueous phase, electrochemical reactions on the metal surface and diffusion of species to and from the bulk solution, as well as diffusion through porous iron carbonate films. Corrosion rate calculations are carried out to include the risk of mesa attack. It includes a relatively strong effect of protective films, which is sensitive to pH and temperature. For this reason, there is a tendency for it to predict low corrosion rates for high temperatures and high pH (Nyborg, 2010).

2.5.4 Multicorp Model

This is a mechanistic model developed by Ohio University and was originally based on the KSC model using mechanistic simulation of chemical, electrochemical and transport processes that typically take place in CO₂ corrosion (Nyborg 2010).

Advancements to the original KSC model came by way of the development of a multiphase flow model with precipitation of iron carbonate films and effects of oil wetting. The model results were then further verified against laboratory and field data (Nesic et al., 2005; Nyborg 2010).

The model also includes the effects of organic acids such as acetic acid and H₂S, including iron sulphide film precipitation. It is based on detailed mechanistic modelling of the kinetics of chemical and electrochemical reactions of species in the bulk solution and on the metal surface respectively and the transient transport of species from the bulk solution to the metal surface. The model is well suited for facilitating an understanding of the various CO₂ corrosion mechanisms through the inclusion of corrosion inhibitor performance and the effects of multi-phase flow in addition to the other facets mentioned.

2.6 CO₂ Corrosion Mechanism between Low and High Partial Pressures

2.6.1 Carbon Capture and Storage (CCS)

In the CCS industries, carbon dioxide is captured and transported at elevated pressures in pipes for the purpose of controlling emissions of the gas in order to reduce its greenhouse gas effects in the atmosphere. The gas is transported at pressures above those characteristic of oil and gas pipelines; usually exceeding 20 bar, and as a single-component stream, to suitable storage sites at secure locations for geologically significant timescales (Downie et al., 2007). CO₂ can be transported on the liquid side or on the vapour side of the vapour/liquid line running between the critical point (74bar, 31°C) and the triple point (5bar, -56°C) while not allowed to cross the dashed line as shown in Figure 2-12 (IEA GHG, 2010). It is often transported as a dense-phase liquid at elevated pressures for convenience and efficiency (Downie et al., 2007). However assessing the corrosion risks for pipelines at these elevated pressures is challenging and potentially unpredictable due to the non-ideal dissolution of CO₂ in the liquid phase coupled with the presence of impurities such as SO_x and NO_x gases with traces of water.

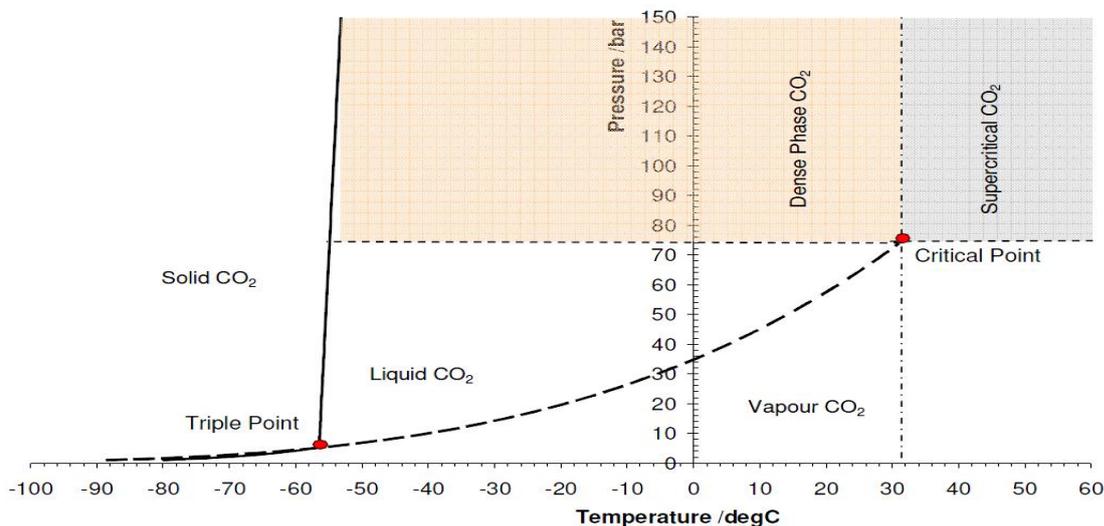


Figure 2-12. Phase diagram for pure CO₂ – the captured gas can be transported strictly on either side of the dashed line without cross over. Dense-phase region represents the most efficient means of pipeline transport. (IEA GHG, 2010)

It should be noted that prior to transport, the captured gas has to be elevated to pressures above its critical point however the presence of impurities in the stream influences the critical

temperature and pressure as well as the width of the phase envelope that consequently dictates the region of two-phase flow as shown in Figure 2-13.

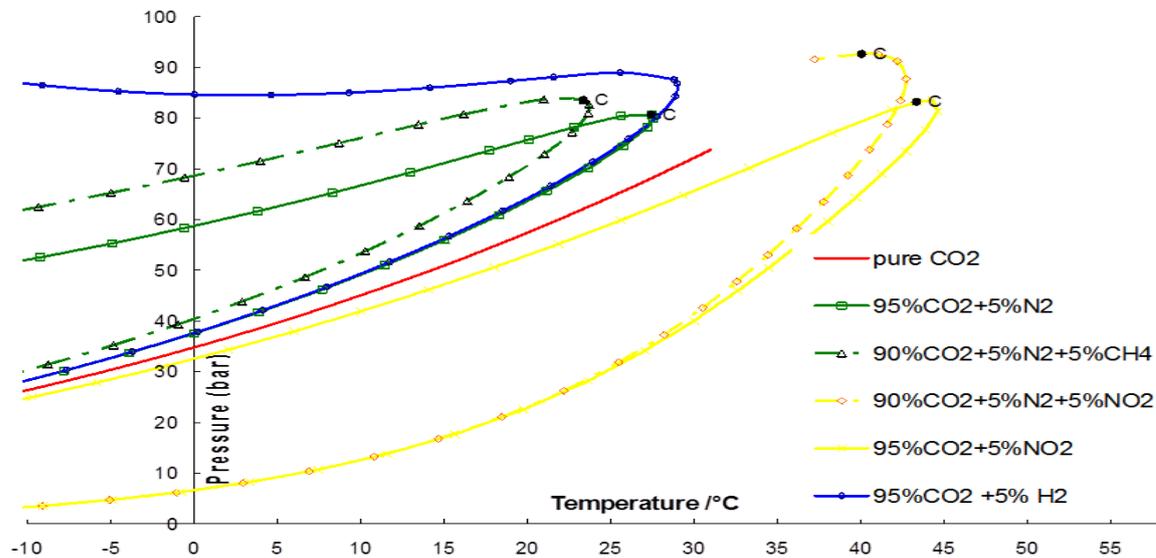


Figure 2-13. Phase envelopes of CO₂ with a combination of impurities

(Downie et al., 2007)

Increasing percentage content of impurities in the CO₂ stream is known to reduce the supercritical phase of the gas which impacts on the operating region of the pipeline thus the pipeline may have to be operated at higher pressures in order to maintain single-phase flow of the gas stream (supercritical/dense phase), for instance. The presence of impurities also affects the density and viscosity of the stream and the extent of their influence are dependent on the type, quantity and combination of impurities present (Downie et al., 2007). A well-documented implication of the presence of impurities is its effect on the temperature drop and consequent hydrate formation (Downie et al., 2007).

2.6.2 Research Directions

The objective of this research is to obtain the solubility values for a mixed CO₂-H₂O system using appropriate equations of state (EOS). An equation of state is a mathematical function that describes the pressure, volume and temperature (PVT) behaviour of a substance. The solubility results will first be determined by obtaining compressibility values from the given EOS and then using these to obtain fugacity coefficients for a mixed CO₂-H₂O system. The compressibility and fugacity coefficients are thermodynamic concepts that represent the

departure of real gases from ideal gas behaviour. The acidity of the aqueous solution is thus known by converting the solubility values to solution concentrations.

It is also possible to determine the acidity of the aqueous media for mixed $\text{CO}_2\text{-H}_2\text{S}$, $\text{CO}_2\text{-SO}_x$ and $\text{CO}_2\text{-NO}_x$, by application of the same principles stated for the $\text{CO}_2\text{-H}_2\text{O}$ system.

2.6.3 The Limitation of Henry's law for High Pressure Conditions

As previously discussed, in the oil and gas industry, CO_2 partial pressures often do not exceed 2MPa. Carbon dioxide is said to dissolve in water for partial pressures no greater than 20bar (2MPa). Such dissolution is considered to be ideal and Henry's law is applicable for these ranges of pressures however for carbon dioxide partial pressures that exceed 20bar, dissolution is no longer ideal thus Henry's law is inapplicable for such pressures, as shown in Figure 2-14 (Spycher et al., 2003).

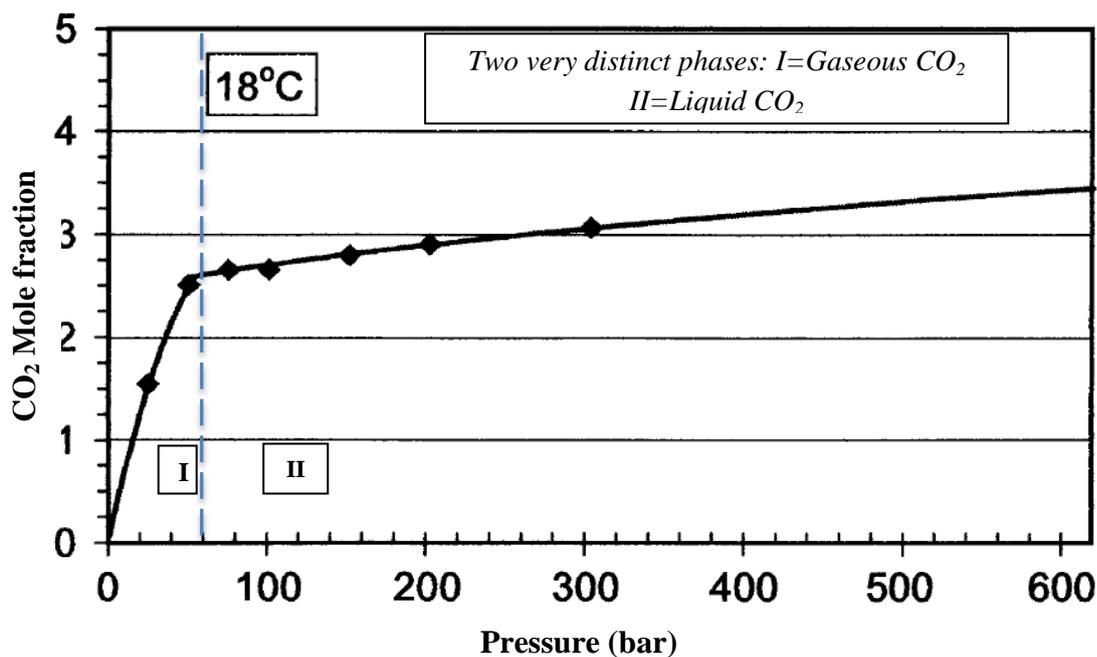


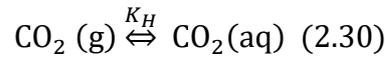
Figure 2-14. Solubility of CO_2 in H_2O (Spycher et al., 2003)

There is therefore a need to take into account the non-ideality of the gas phase as it dissolves in the aqueous phase. Due to the fact that co-solubility of carbon dioxide in water and water in carbon dioxide are the means by which CO_2 corrosion occurs, solubility of these species is thus investigated in terms of solution thermodynamics as is briefly outlined in the following Section. The methodology for accurate supercritical CO_2 corrosion modelling is discussed by (Choi and Nesic, 2009) as well as (Mohamed et al., 2011).

2.6.4 Methodology for CO₂ Corrosion Prediction at High Pressure Conditions

Equilibrium constants for solubility (K_{CO_2} and K_{H_2O}) were first obtained for the CO₂-H₂O system where necessary. Dissociation constants for the dissociation of water, dissolution of CO₂, the hydration of CO₂, carbonic acid dissociation and bicarbonate ion dissociation were also obtained (Mohamed et al. 2011).

Of all these reactions, the dissolution of CO₂ is briefly discussed because it was derived from basic thermodynamic relationships. The dissolution of carbon dioxide from its gaseous form to its aqueous form is summarised by the following reaction:



$$K_H = \frac{[CO_2(aq)]}{P_{CO_2}(g)} = \frac{[C_{CO_2}]}{P_{CO_2}} \quad (2.31)$$

K_H is the Henry's law constant and its unit is mol/ (litre x bar)

Where:

P_{CO_2} = partial pressure of gaseous CO₂ in bar

$[C_{CO_2}]$ = concentration of CO₂ in aqueous solution

Accounting for non-ideality of the gas phase, the following equations are derived from the definition of fugacity and applying necessary corrections for the pressure:

$$y_{H_2O} = \frac{K_{H_2O}(1 - x_{CO_2})}{\phi_{H_2O}P_{tot}} \cdot \exp\left\{\frac{(P - P^0)\bar{V}_{H_2O}}{RT}\right\} \quad (2.32)$$

$$x_{CO_2} = \frac{(1 - y_{H_2O})\phi_{CO_2}P_{tot}}{55.508K_{CO_2}} \cdot \exp\left\{\frac{-\bar{V}_{CO_2}(P - P^0)}{RT}\right\} \quad (2.33)$$

The correction for K is given by: $K_{(T,P)} = K_{(T,P^0)} \exp\left\{\frac{V_i(P - P^0)}{RT}\right\}$ (2.34), and serves as a correction for pressure at a given temperature in the two preceding equations.

Where:

y_{H_2O} is the mole fraction of water in the CO₂ phase

x_{CO_2} is the mole fraction of CO₂ in the water phase

ϕ_{H_2O} is the fugacity coefficient for water (dimensionless)

ϕ_{CO_2} is the fugacity coefficient for CO₂ (dimensionless)

\bar{V}_{H_2O} is the average partial molar volume of water in cm³/mol

\bar{V}_{CO_2} is the average partial molar volume of CO₂ in cm³/mol

P , P^0 and P_{tot} are the applied pressure, atmospheric pressure (0.101325 MPa) and the total system pressure respectively

R = Molar Gas Constant in J/mol.K

T = Absolute Temperature in K

These equations were then used to solve for the fugacity coefficients so that the volumes of the compressed CO₂ gas and H₂O are then calculated using an equation of state. In this case, the Redlich-Kwong equation was used and yielded cubic equations which were then solved using the Newton-Raphson numerical algorithm (Mohamed et al., 2011). The solubility constant, K_H was then computed. The Redlich-Kwong EOS is stated as follows (Redlich and Kwong, 1949):

$$P = \frac{RT}{V - b} - \frac{a}{V(V + b)T_k^{0.5}} \quad (2.35)$$

Where:

P = Pressure (in MPa)

V = Volume (in m³)

R = Molar Gas Constant (J/mol.K)

T_k = Absolute Temperature (in K)

a = EOS parameter → Measure of the attractive forces between molecules

b = EOS parameter → Related to size of the molecule

2.7 Neural Networks

Artificial neural networks are information-processing units that are quite similar in terms of mode of operation to biological nervous systems such as the brain and its highly complex network of neurons – artificial neural networks (ANNs) are inspired by their biological counterparts (Haykin, 1999; Krogh, 2008). Essentially, neural networks are models which can be ‘trained’ to forecast, by developing a correlation between a known set of input variables and output problem descriptors. They obtain a correlation from the use of transfer functions thereby assigning ‘weighted scalars’ to input data which is subsequently matched to a given output. Matching is achieved by adjusting and re-adjusting the weighted scalars accordingly in a continuous loop, thereby reducing the error between the target and output as seen in Figure 2-15. In so doing, the neural network learns and trains such that it utilises its ‘experience’ in predicting the outcome for a different set of input data.

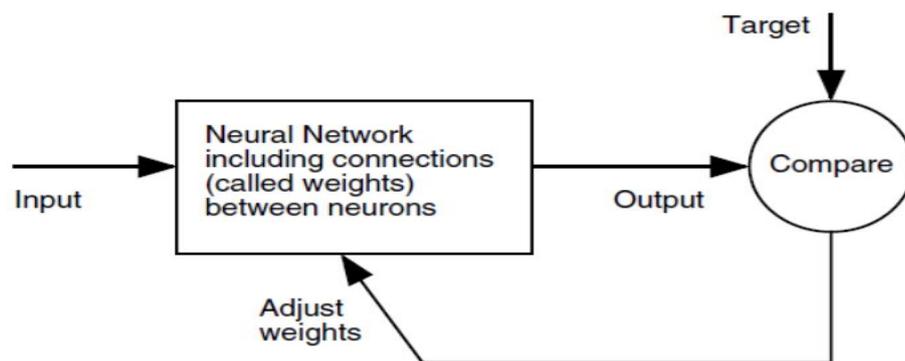


Figure 2-15. Flow loop of input to output via continuous adjustment of weights

(Demuth et al., 2009)

Often neural networks (NNs) are designed in the form of neural architectures. A typical layered structure of a NN is shown in Figure 2-16. A NN may contain as many layers as necessary depending on the task to be carried out. However, if a NN has been trained with a large number of layers on a small dataset, then there is the risk of over-fitting. When a NN is over-trained in this manner, it loses its ability to adapt and fit a wider range of datasets and as such is said to lose its ability to generalise (Radonja and Stankovic, 2002).

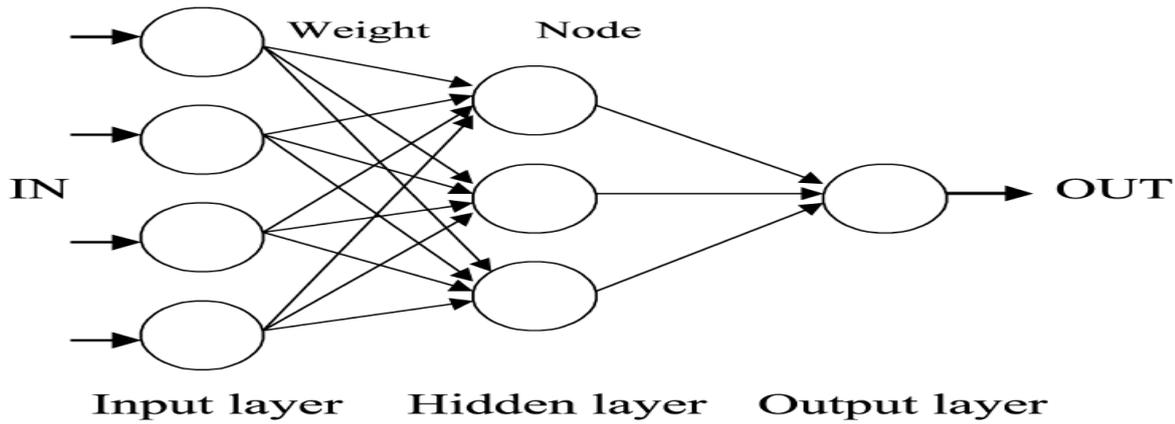


Figure 2-16. Basic layered structure of a three-layer artificial neural network

(Al-Fattah and Startzman, 2003)

As earlier stated, the transfer function determines the correlation that maps input data to output. For this reason, the type of transfer functions to be used should be based on the characteristics of the given input and output data, such as the range and magnitude of each data-value within the dataset (Vogl et al., 1988). The purelin and log-sigmoid transfer functions are shown in Figure 2-17, on the left and right respectively.

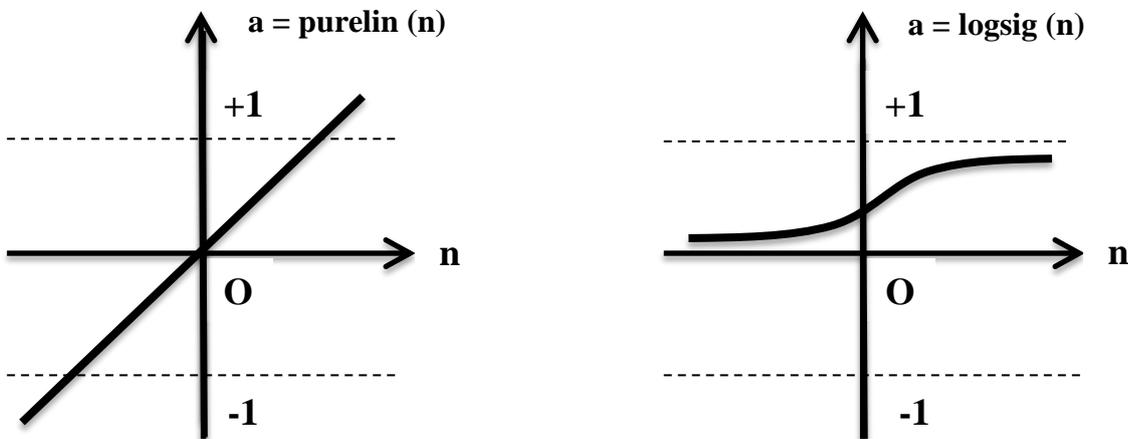


Figure 2-17. The purelin and log-sigmoid transfer functions (Beale et al., 2014)

The purelin transfer function is often used in the last layer of a multilayer NN as function approximator while the logsig transfer function is used in the hidden layer of a multilayer NN because it is differentiable (Beale et al., 2014). Training algorithms are a set of instruction code that controls the optimisation method for a given NN. They therefore control how the

weights of a NN are set (Beale et al., 2014). The definitions of some training algorithms are shown in Table 2-4.

Table 2-4. Definitions of the training functions used (Beale et al., 2014)

Training Function	Training Algorithm Group	(Definition) Updates network weights according to:
Traindm	Gradient Descent	Gradient descent with momentum
Trainrp	Gradient Descent	Resilient backpropagation (Rprop)
Trainscg	Conjugate Gradient	Scaled conjugate gradient method
Traincgf	Conjugate Gradient	Conjugate gradient backpropagation with Fletcher-Reeves updates
Trainbfg	Quasi-Newton	Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method
Trainlm	Quasi-Newton	Levenberg-Marquardt optimisation

2.7.1 Case Study on the Implementation of Neural Networks in Model Prediction

This research study is about a ‘purpose-built’ experiment that involved a 10-problem/variable domain in which pitting corrosion potential was investigated using a sample of austenitic type 304 stainless steel. The ten variables investigated were F^- , Cl^- , Br^- , I^- , CO_3^{2-} , OH^- , SO_4^{2-} , $S_2O_3^{2-}$, NO_3^- and temperature (Cottis et al., 1999).

The neural network employed was a simple 4-node network with a single hidden layer. Training of this network involved making use of two-thirds of the experimental data available while the remaining one-third of the data was used for validation purposes to signal the termination of training. This resulted in a network with a total of 75 weights and plots were then produced to observe the effect of each ion on pitting corrosion potential. Plots were produced in order to compare experimental results by (Man and Gabe, 1981), with those of the neural network where experimental results were measured using the same material and test method as was used to produce the training data (Cottis et al., 1999). One such plot is shown in Figure 2-18.

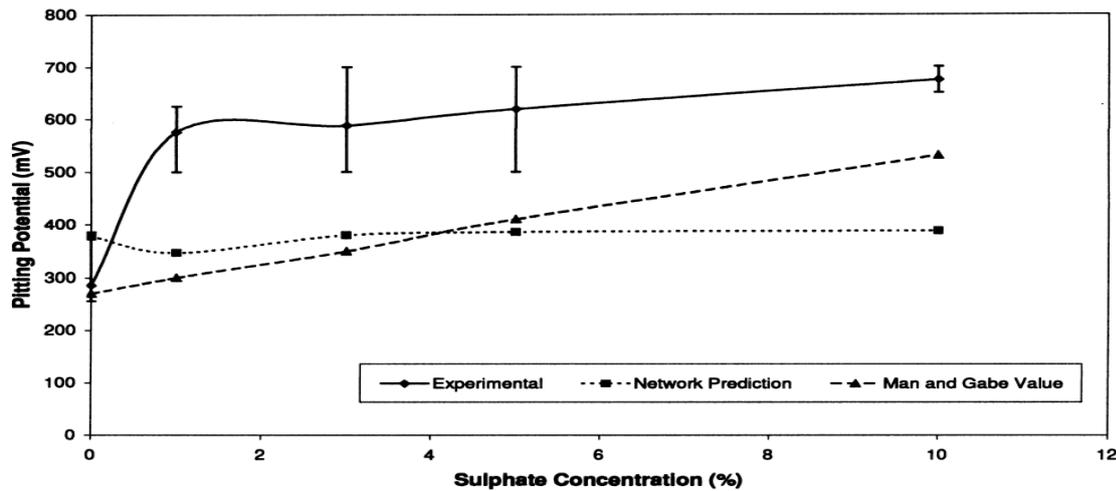


Figure 2-18. Plot showing the effect of sulphate ion on pitting potential of steel in 3% NaCl solution. (Cottis et al., 1999)

Overall, the training dataset was small therefore there was the need to design a small neural network such that it relies on generalisation as opposed to memorisation. The neural network performs reasonably well in situations where elements of extrapolation are required when modelling problems with small data sets.

Furthermore, based on observations from these studies it can be said that, the performance of neural networks is largely dependent on the quality of training data however given that corrosion data is largely inconsistent and is known to contain a substantial amount of ‘noise’, results are not likely to be meaningful as are likely to be models of the noise rather than that of the average behaviour (Cottis et al., 1999).

Noise in corrosion data may be as a result of the use of Monte Carlo techniques to fill gaps in summarisation plots, the use of faulty equipment, poor conduction of experiments and a failure to report significant variables while carrying out corrosion experiments (Cottis et al., 1999). Monte Carlo techniques, in this instance, are used to improve upon the quality of the reported data and the appearance of summarisation plots before the application of neural networks for corrosion prediction. In another study in which sulphuric acid corrosion of steel was investigated, the question of validation was raised (Helliwell et al., 1996). Validation had been carried out to a limited extent. It is also apparent that the degree of confidence that could be placed in prediction was questionable.

In conclusion, there is a need to state the confidence that can be placed in the prediction of a given neural network and it is also necessary to predict both model variability and mean behaviour though it will require copious amounts of training data sets.

2.8 Fuzzy Logic Systems

Fuzzy logic systems provide a language with syntax and semantics that can translate qualitative knowledge into numerical reasoning. Fuzzy linguistic variables include low, high, good and so on and when these are combined with fuzzy set operators such as and/or, the process is termed approximate reasoning (Hajizadeh, 2006).

Approximate reasoning is a term used when referring to fuzzy logic due to the fact that it does not deal in absolutes. In fact, it is based on binary logic, but it differs from it. Where binary logic assigns 0 or 1, for a false or true statement, fuzzy logic instead assigns an approximate value in the range: 0-1.

Essentially, fuzzy systems process input data converting it to output data according to the steps shown in Figure 2-19. Fuzzification involves assigning linguistic variables to input data, often applying 'if-then' rules as conditional statements. The 'if-then' rule is discussed more in Section 2.8.1. The implication phase assigns a membership function to the data. This is the mapping of the data into the range: 0-1, also known as categorising the input data into degrees of membership (Zadeh, 1965), as seen in Set X of Figure 2-19. It is worth noting that membership functions may take several forms: triangular, trapezoidal and continuous among others. These are discussed further in Section 6.3.

The membership function selected for a given input dataset depends upon the distribution of the data within the dataset. Aggregation is the process of putting all fuzzy sets together in an aggregate fuzzy set and defuzzification translates the fuzzy information back to readable form as an output, as seen in Figure 2-19.

There are two main types of fuzzy inference systems (FIS), and these are the Mamdani and Takagi-Sugeno systems. These will be discussed further in the next Section however the main difference between these two systems is in the defuzzification step. The Mamdani re-transforms fuzzy sets to output using a defuzzification technique – directly stripping away fuzzified data to reveal the output.

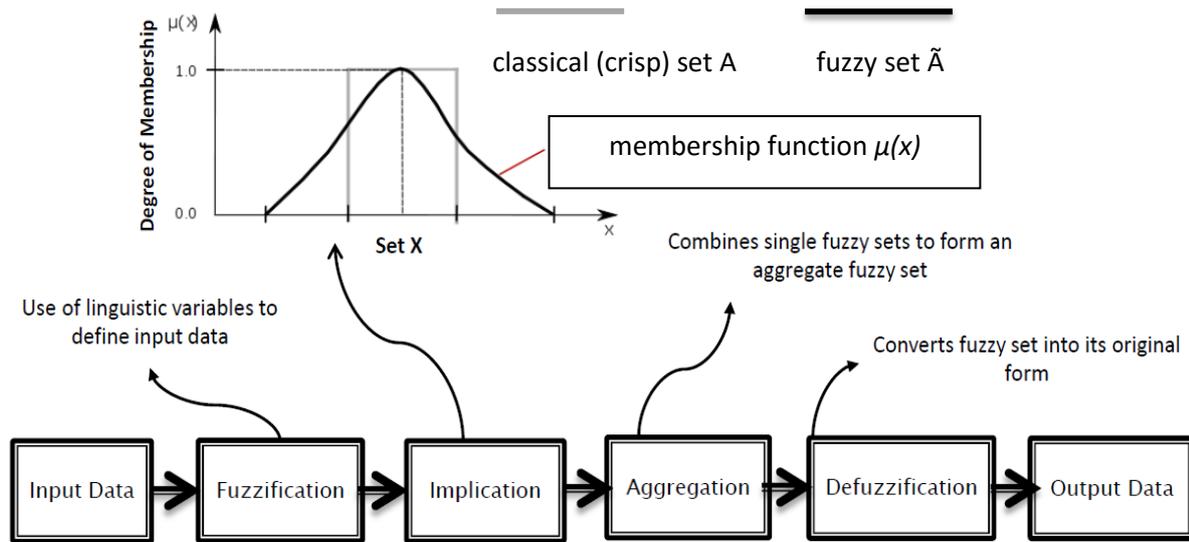


Figure 2-19. Steps in fuzzy inference process

The Takagi-Sugeno system on the other hand, makes use of a weighted average technique to convert fuzzified data into output (Kaufmann and Gupta, 1985; Fuzzy Logic Toolbox User's Guide, 2015). In general, fuzzy logic systems excel particularly during instances where gaps in numerical banks of data exist and where in such cases, exact measured process variables are considered too imprecise to justify the use of numbers. This is often quite common in several engineering problems (Hajizadeh, 2006).

2.8.1 Case Study on the Execution of the Fuzzy Logic System

In this Section, a case study on the use of fuzzy logic for CO₂ corrosion rate prediction is described (Hajizadeh, 2006). A graphical user interface (GUI) was set up with six input parameters: pressure, temperature, oil production rate, gas production rate, H₂S and CO₂ mole percentages. The single output is corrosivity (mils/year).

A simple set of if-then rules was used in the execution of a fuzzy set in which the input variables are classed as members or non-members given that a fuzzy set is defined as a collection of ordered pairs of the form $A = \{x, \mu(x)\}$, where A describes the relationship between an unknown quantity, x , and a membership function, $\mu(x)$. It should be noted that in general, fuzzy sets can be divided into crisp boundary and non-crisp boundary fuzzy sets but for each type, quantities are classed as members or non-members.

In terms of the execution of the if-then rules of a fuzzy system, the closer an argument is to the 'if' part, the greater the influence the 'then' part has on the mapping of input variables.

For instance, for the statement: ‘If temperature is low, then corrosion rate is low’; for small input temperature values, the corrosion rates will definitely be assigned small magnitudes. The fuzzy system then sums up the ‘then’ parts and implements a defuzzification system which yields an output. The Takagi-Sugeno (TS) inference system was used for this research study in which 50 different corrosion rate data points are used for training however a choice can be made between its use and the use of the Mamdani inference system. The main difference between these inference systems is that the former involves the use of only constant output membership functions however the TS system can be applied to mimic the commonly-used Mamdani by implementing a hyperbolic transfer function (Hajizadeh, 2007; Fuzzy Logic Toolbox User’s Guide, 2015).

Furthermore, three-dimensional (3-D) surface contour plots were produced for the given user-input parameters and the trends were found to be in general agreement with research literature. Two such plots are shown in Figure 2-20. In addition, the 3-D plots can be used in CO₂ corrosion rate prediction and can also serve as an informative tool which lends itself to the understanding of specific relationships between model input variables and the output, corrosion rate (Hajizadeh, 2007).

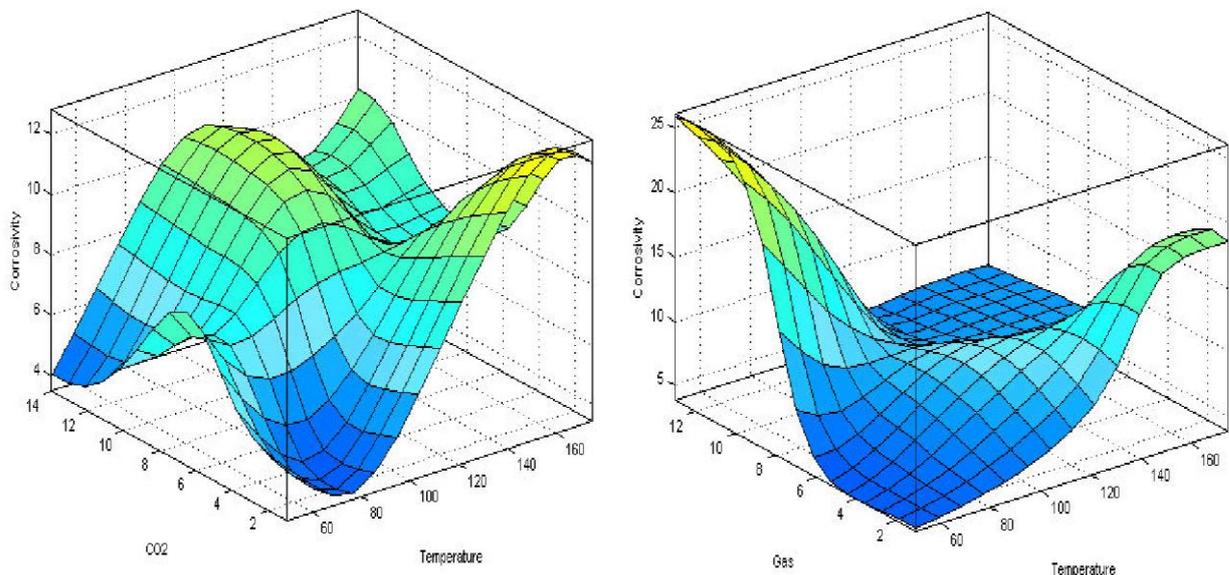


Figure 2-20. 3-D Surface contour plots showing the relationships between input variables (CO₂ mole % with temperature and gas production rate with temperature) and the output variable, CO₂ corrosion rate (Hajizadeh, 2007)

2.9 Conclusions

Since the discovery of CO₂ corrosion in gas wells located in Texas, USA, in the 1940's, several research undertakings on this subject have been carried out. CO₂ in the dry gas form is non-corrosive at oilfield and CCS conditions. It however becomes corrosive when it dissolves in the aqueous phase, forming carbonic acid. It is particularly more corrosive when localised mesa-type attacks take place. Mesa-type attacks occur in low fluid-flow environments, in which Fe₃C scale is somewhat prevalent. It is also very dependent on the composition of fluid flow. Fe₃C scales provide inadequate protection, thus expose the metal surface underneath to severe corrosive attack.

The major factors influencing CO₂ corrosion were also discussed and these were broadly categorised into: environmental, physical and metallurgical factors. The environmental factors were also discussed with respect to actual oil and gas producing fields in the North Sea; data contained in Table 2-1. Explanations were provided for the quoted corrosion rate estimates given in Table 2-1 in terms of the corrosion rate-determining factors, i.e. the environmental factors. An increase in the magnitudes of temperature and CO₂ partial pressures were observed to lead to elevated corrosion risks for pipelines. For pH, the reverse was the case, low pH implies high corrosivity while for fluid flow; very high velocities appear to cause corrosion rates to plateau, as seen in Figure 2-7D.

The condition for corrosivity with regards to a H₂S-containing environment is: $P_{CO_2}/pH_{2S} < 200$. Also, H₂S corrosion is fundamentally different from CO₂ corrosion, in that it is not electrochemical whereas the latter is. H₂S corrosion proceeds via direct combination of the gas with the metal in a 'solid state corrosion reaction' in which species transport from the bulk solution through boundary layers to the metal surface dictates the rate of corrosion.

For accurate modelling of CO₂ corrosion at high pressures, the application of appropriate equations of state is required. Some research directives were stated, one of which was the determination of the pH for a mixed CO₂-H₂O system.

The use of NNs for the prediction of low pressure CO₂ corrosion rates is well-documented however in this thesis; it is also used for the prediction of CO₂ corrosion rates under high pressure conditions. Some reference was made to the training of neural networks for small datasets in order to retain their ability to generalise. High pressure CO₂ corrosion datasets in open literature are very small in size therefore the need to adapt the NNs for this particular

purpose is crucial. NN modelling of high pressure CO₂ is discussed in the chapter on neural network models.

For fuzzy inference systems (FIS), the process in which input data is converted to output data was discussed. FIS employ approximate reasoning techniques to problem-solving through the use of human language descriptors and assigning values between: 0 to 1 to represent these descriptors. The way in which the assignments are carried out is governed by the use of the membership function. It is imperative to understand the data being modelled in order to be able to select an appropriate membership function. The advantage this modelling technique has over the others is its ability to directly visualise the various relationships between input data variables in both two-dimensional and three-dimensional space.

Chapter 3. CO₂ Corrosion Models

3.1 Introduction – CO₂ Corrosion Models

A large number of models have been developed by oil companies for the purpose of predicting pipeline CO₂ corrosion (Nyborg, 2002; Nyborg, 2010). In Section 2.5 of the previous chapter, some established CO₂ corrosion models were discussed such as Shell's HydroCor, Total's Corplus, IFE's KSC model and Ohio University's Multicorp models.

These models differ from each other because of the different philosophies behind their development. One way to classify these models is by conservatism. The concept of conservatism is explained by the scale formation effect. For those models described as conservative or known to over-predict corrosion rates, these models are said to take little or no account of the effect of the corrosion film (passive layer) on the overall CO₂ corrosion mechanism. It is known that, particularly at high temperatures, scale formation layers are more stable and compact hence the net effect on the surface of the metal is that corrosion is reduced.

The less conservative models often assume the influence of film and scale formation hence generally predict lower corrosion rates than conservative models. It is also said that the less conservative models also assume oil-wetting tendencies as opposed to the mostly water-wetting tendencies of conservative models (Nyborg, 2002; Nyborg, 2010). The aqueous phase is the continuous phase in contact with the metal surface for water-wetting systems hence corrosion risks are greater than for oil-wetting systems whose continuous phase is oil.

In general, these two factors – the effect of corrosion products/film layer and oil/water wettability are the main distinguishing features in the modes of operation of the several models developed for CO₂ corrosion prediction. In this chapter, the corrosion models discussed fall into one of two categories – empirical or mechanistic. The former implies models developed with the use of raw experimental or field data to extract underlying relationships between measured variables while the latter involves the development of a prediction system based on the mechanisms that underpin the physical phenomena being investigated. The 1975, 1991, 1995 De Waard and Norsok models are empirical models (Nyborg, 2010) while the Nescic-Postlethwaite-Olsen (NPO) model is purely mechanistic (Nesic et al., 1996).

Also, in Section 2.4.1, as part of the environmental factors affecting CO₂ corrosion, the presence of H₂S was discussed. The scenarios for the severity of corrosion risks in the

presence of H₂S as well as the mechanics and kinetics of formation of iron sulphide scale were also outlined. In this chapter, H₂S corrosion is discussed in terms of the use of Ohio University's freely accessible Freecorp software (Ohio University, 2008) to predict corrosivity for H₂S corrosion data in research literature (Lyle and Schutt, 1998; Nesic et al, 2008).

In the mechanics and kinetics of iron sulphide scale formation of Section 2.4.1, some H₂S scale types were listed, one of which was mackinawite. In Section 3.7.2, with the aid of the aforementioned Freecorp software, the H₂S solution concentration is expressed in terms of the distance of corroding species (H₃O⁺ and H₂S) from the steel surface. Mackinawite is said to constitute the inner film formed on the steel surface. The Freecorp model is also used to predict corrosivity for CO₂ corrosion data in research literature (Crolet et al, 1999; Guo et al, 2005) as it incorporates the electrochemical philosophy of the NPO model.

3.2 The 1975 De Waard-Milliams Correlation

In an attempt to come up with a correlation capable of accurately predicting CO₂ corrosion in oil and gas pipelines, the de Waard-Milliams equation was formulated as follows:

$$\log_{10}(V_{corr}) = 5.8 - \frac{1710}{T_k} + 0.67 \log_{10}(P_{CO_2}) \quad (3.1)$$

Where:

V_{corr} = Corrosion rate (mm/year)

T_k = Temperature (K)

P_{CO_2} = carbon dioxide partial pressure (MPa); $P_{CO_2} = \text{mol\% } CO_2 \times \text{total pressure}(P)$

(de Waard and Milliams, 1975; Zhang et al., 2012b)

The correlation expresses corrosion rates in units of mm/year in terms of temperature, T and carbon dioxide partial pressure, P_{CO_2} . A typical plot for the correlation is shown in Figure 3-1.

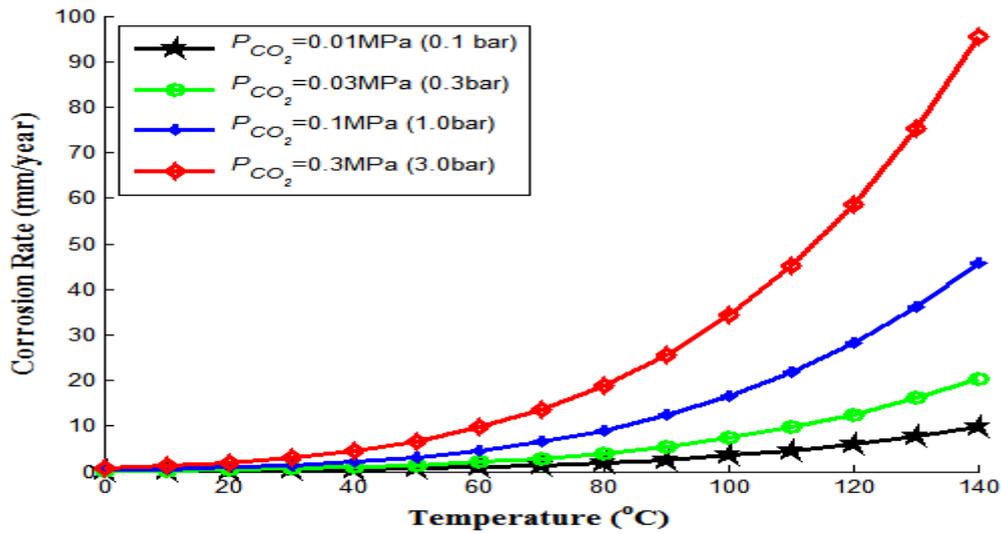


Figure 3-1. Plot of corrosion rate against temperature for varying carbon dioxide partial pressures, $P_{CO_2} = 0.01, 0.03, 0.1$ and 0.3 MPa (without inhibition from carbonate scale)

From Figure 3-1, it is observed that corrosion rates increase as a power function – power of 10, as the temperature increases. Herein lies the limitation of the model, in that typically plots of corrosion data from laboratory results as well as field data show a distinct peak at approximately 60-90°C followed by a decrease in magnitude of corrosion rate with a further increase in temperature beyond 90°C, as shown in Figure 3-2.

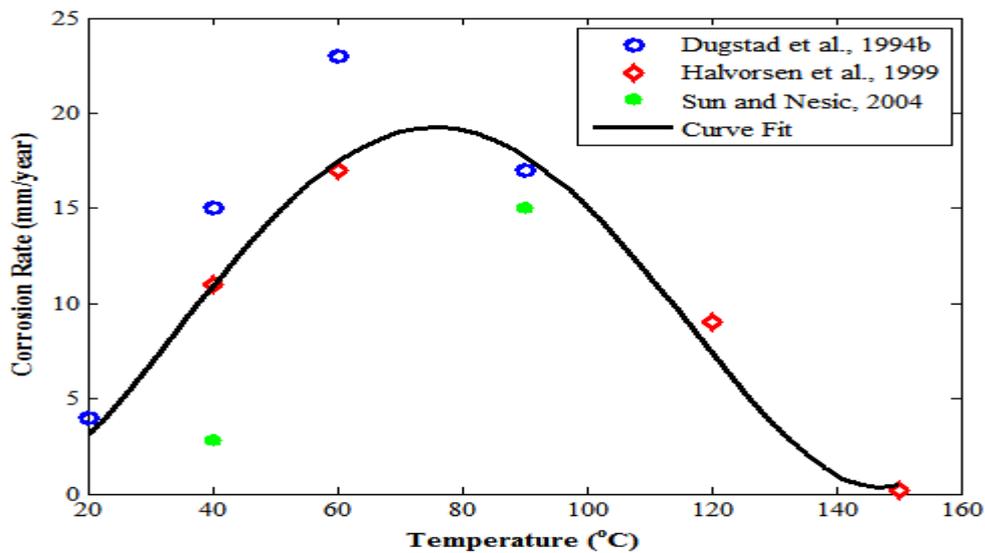


Figure 3-2. Typical plot of corrosion rate against temperature from experimental CO₂ corrosion data (Dugstad et al., 1994b; Halvorsen et al., 1999; Sun and Netic, 2004)

However for the plot in Figure 3-1, there is no peak for this range of temperatures hence the reason a subsequent correlation was formulated in order to account for the film formation effect.

The other limitation of this correlation is the use of carbon dioxide partial pressures instead of fugacity. At higher CO₂ pressures, such as for pressures that exceed 20bar, for instance, a more appropriate and accurate approach will involve the use of the concept of fugacity as opposed to partial pressure (de Waard et al., 1991).

3.3 The 1991 De Waard-Lotz-Milliams Correlation

The 1991 correlation is basically the same as the 1975 equation in terms of statement and variables with the exception of the fugacity term, f_{CO_2} , which replaces the previous carbon dioxide partial pressure term, P_{CO_2} . The equation is stated as follows (de Waard et al., 1991):

$$\log_{10}(V_{corr}) = 5.8 - \frac{1710}{T_k} + 0.67 \log_{10}(f_{CO_2}) \quad (3.2)$$

Where:

T_k = Temperature (K)

f_{CO_2} = Carbon dioxide fugacity (MPa)

Application of this model involves the use of correction factors which are associated with a physical or chemical effect that causes a slight change in the corrosion prediction result (de Waard et al., 1991). The physical or chemical effect includes parameters affecting CO₂ corrosion such as pH and scale formation, which are not stated directly in Equation 3.2 as variables. Therefore, separate correction factors for each of these parameters are determined

Generally, for any physical or chemical effect being corrected for, the correction factor, which is usually less than unity (<1), is multiplied by the result obtained from the Equation 3.2 and this often yields corrosion rates of decreased magnitudes as compared to the original predicted results from Equation 3.2, which are considered to be over-conservative (de Waard et al., 1991). Specifically, the correction for pH will tend to yield correction factor results that exceed unity (>1) for media that are under-saturated with corrosion products, Fe₂CO₃ or Fe₃O₄. The reason being that for under-saturated systems, there is a steady anodic dissolution

rate of iron on its surface and this proceeds until the system is sufficiently saturated after which the rate of dissolution slows down. Thus corrosion rates are higher for under-saturated systems because film formation has not yet proceeded to the point where it inhibits further corrosion at the metal surface therefore any pH correction factors obtained will exceed unity.

3.3.1 Accounting for High-temperature Protective Films

It is known that the precipitation of iron carbonate, FeCO_3 , or ferrous oxide, Fe_2O_3 , ultimately leads to the formation of protective film layers however they are not the only requirement for stable protective film formation (de Waard et al., 1991). Temperature has a direct influence – increasing temperatures increase the rates of several known chemical reactions as explained by the Arrhenius theory and as such in CO_2 corrosion, this implies that the carbonate layers will be formed at a much faster rate due to an increased rate of reaction between ferrous ions and carbonate ions to form FeCO_3 (siderite).

These FeCO_3 films are formed and adhere onto the metal surface and reduce the surface area of the metal exposed to corrosive attack thereby reducing corrosion rates and resulting in temperature maximum (peak) for corrosion rates. The temperature corresponding to the maximum corrosion rate is the scaling temperature and is given by Equation 3.3. At this temperature, ferrous ion concentration (Fe^{2+}) formed at the metal surface and local pH are such that a protective film is formed. (de Waard et al., 1991)

$$T_{scale} = \frac{2400}{6.7 + 0.6 \log_{10}(f_{CO_2})} \quad (3.3)$$

Where:

T_{scale} = Scaling Temperature (K)

f_{CO_2} = Carbon dioxide fugacity (MPa)

While film formation can occur at any temperature, its protection of the metal surface is greater at higher temperatures. At lower temperatures, for instance, temperatures less than 60°C , the corrosion product film has a smudge-like appearance and is easily removed by flowing liquids while at higher temperatures, the texture is different, usually coarser and is less easily washed away (de Waard et al., 1991). It is also worth noting that flowrate has an effect on scaling temperature; a higher flowrate will result in a higher scaling temperature. Also, a higher bulk pH will tend to lower this temperature.

Figure 3-3 shows the plot of corrosion rate against temperature for varying CO₂ fugacities.

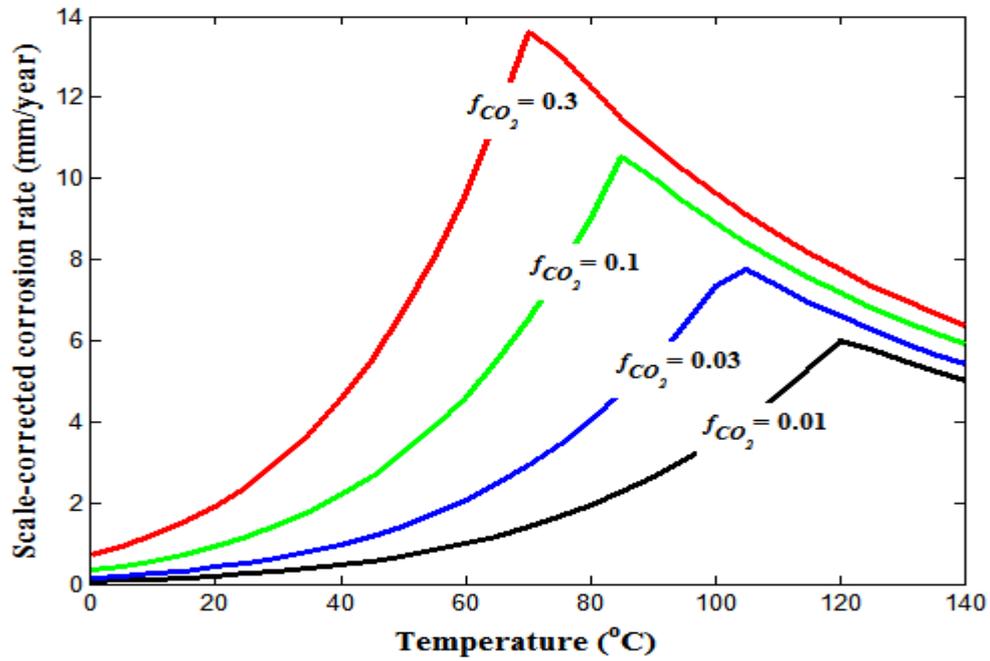


Figure 3-3. Plot of corrosion rate against temperature for varying CO₂ fugacities, $f_{CO_2} = 0.01, 0.03, 0.1$ and 0.3 MPa

For the ascending part of the curve, Equation 3.2 is applied, however for the descending part of the curve, Equation 3.5 is applied. Equation 3.5 is derived as follows:

From Equation (3.3), re-arranging:

$$\frac{2400}{T_k} - 0.6 \log_{10}(f_{CO_2}) - 6.7 = 0 \quad (3.4)$$

Multiplying the right hand side of Equation (3.2) by the left hand side of Equation (3.4), yields the result for calculating the new corrosion rate, Equation (3.5):

$$\log_{10}(V_{corr}) = \left\{ 5.8 - \frac{1710}{T_k} + 0.67 \log_{10}(f_{CO_2}) \right\} \times \left\{ \frac{2400}{T_k} - 0.6 \log_{10}(f_{CO_2}) - 6.7 \right\} \quad (3.5)$$

The scaling temperature is evaluated from Equation 3.3 using the given CO₂ fugacity, to determine where the ascending part of the curve ends and where the descending part begins. It is observed from Figure 3-3, that the scaling temperatures for each of the CO₂ fugacities is different, the higher the CO₂ fugacity, the smaller the magnitude of its corresponding scaling temperature and this is due to the fact that at lower temperatures, any scale formation is

washed away. This therefore results in greater corrosion rates for high CO₂ fugacities as the rate at which scale is being washed away is also high. These equations were implemented in Matlab and verified against results published in research literature (Woollam and Hernandez, 2006).

3.4 The 1995 De Waard-Lotz-Dugstad Correlation

This model is currently being used by Shell for predicting CO₂ corrosion rates in the field (Kermani and Morshed, 2003) and accounts for fluid flowrate as this is known to have an influence on corrosion rates. It comprises a simple resistance model which is the combination of the 1991 de Waard-Lotz-Milliams model with a flow-dependent CO₂ mass-transfer model (Woollam and Hernandez, 2006). The model was developed to take into consideration the fact that mass transfer rates need to keep up with reaction kinetics of the corrosion reaction. Hence the established equilibrium reaction is stated as follows (Woollam and Hernandez, 2006; de Waard et al., 1995):

$$C_{CORR} = \frac{[CO_2]}{\frac{1}{k_r} + \frac{1}{k_m}} \quad (3.6)$$

Where:

[CO₂] is CO₂ concentration which is related to CO₂ fugacity, f_{CO_2}

k_r and k_m are rate constants associated with reaction kinetics of the corrosion reaction, the charge transfer reaction and the mass transfer of dissolved CO₂ from the bulk of solution to the surface of steel respectively.

Equation 3.6 can be expressed as Equation 3.7, shown below. Equation 3.7 is the overall resistance model.

$$C_{CORR} = \frac{1}{C_R} + \frac{1}{C_{MT}} \quad (3.7)$$

Where:

C_R is the highest possible reaction rate, i. e. when mass transfer is infinitely fast

C_{MT} is the highest possible mass transfer rate of corrosive species

Hence C_R , also known as the contribution from activation reaction kinetics is associated with ‘worst-case’ corrosion and takes the form of the previously established 1991 de Waard-Lotz-Milliams correlation:

$$\log_{10}(C_R) = 4.93 - \frac{1119}{T_k} + 0.58 \log_{10}(f_{CO_2}) \quad (3.8)$$

Also, C_{MT} is given by: $C_{MT} = 2.45 \frac{U^{0.8}}{d^{0.2}} \quad (3.9)$

Where:

U is liquid velocity (in m/s)

d is pipe diameter (in m)

f_{CO_2} is Carbon dioxide fugacity (MPa)

The pH term stated in the correlation in research literature, is neglected here as this correlation was used for condensed water whose pH is determined by the solubility and dissociation of CO_2 (Woollam and Hernandez, 2006). The plot of corrosion rate against temperature in Figure 3-4 shows the shape and trend for each CO_2 fugacity value.

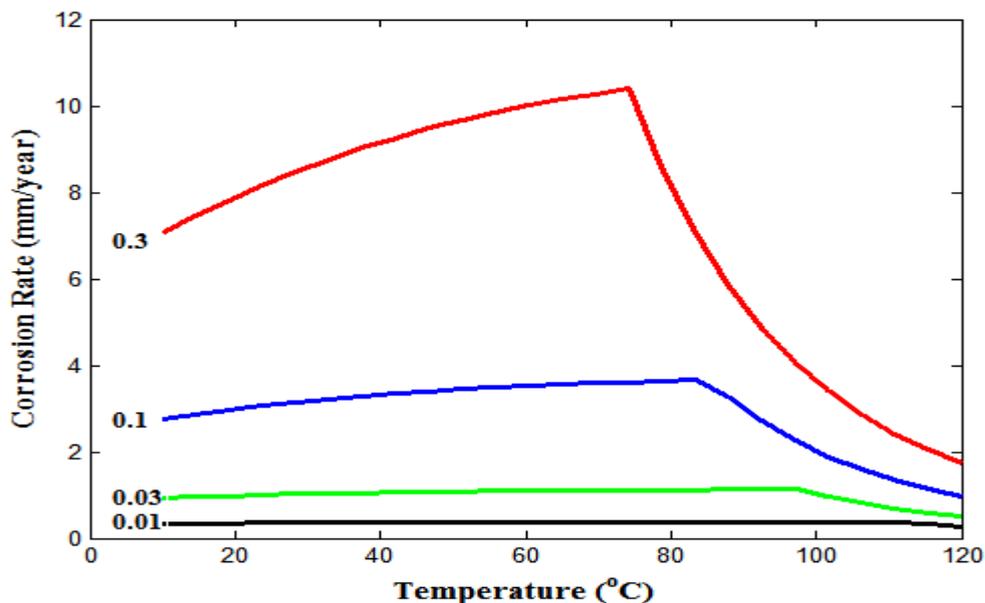


Figure 3-4. Plot of corrosion rate against temperature for varying CO_2 fugacities

$$f_{CO_2} = 0.01, 0.03, 0.1 \text{ and } 0.3 \text{ MPa}$$

It is observed from Figure 3-4, that corrosion rates increase as CO₂ fugacity increases from 0.01 to 0.3MPa (0.1-3bar). The reason being that greater CO₂ fugacities translate into greater concentrations of the gas in bulk solution and consequently greater carbonic acid concentrations. This in turn leads to greater corrosion rates hence greater dissolution rates of the metal at the anode.

Also, it is observed that the scaling temperatures increase as CO₂ fugacity decreases from 3 bar to 0.1 bar. This is due to the fact that at higher temperatures, more stable film layers are formed that are considerably harder to wash away (de Waard et al., 1991). At lower temperatures, scale takes up a smudge-like texture and does not offer protection to the metal surface hence a greater proportion of the metal surface is exposed to corrosive attack leading to greater corrosion rates. Scaling is more likely to occur at lower temperatures for greater CO₂ fugacities because corrosion rates are higher, therefore the bulk solution is likely to get super-saturated more readily before the effect of high temperatures set in – solubility decreases with increasing temperature.

3.4.1 Parameter Study on the 1995 De Waard-Lotz-Dugstad Correlation – Investigating Changes in Flow Velocity and CO₂ Fugacity on CO₂ Corrosion Rate

The plot in Figure 3-5 shows the corrosion profile with temperature for $f_{CO_2}=1\text{MPa}$ (10bar), $T=10\text{-}100^\circ\text{C}$, $\text{pH}=5.0$ and varying flow velocities. It is observed that corrosion rates increase with increasing temperatures regardless of the fluid flowrate velocity until the scaling temperature is reached ($\sim 63^\circ\text{C}$) after which corrosion rates decrease gradually. It is generally known that corrosion specie solubility decreases with increasing temperature and upon reaching the scaling temperature of 63°C , supersaturation takes place and as temperatures proceed to increase further, there is said to be sufficient scale build-up on the metal surface such that the effective area of the metal exposed to corrosive action is reduced hence corrosion rates fall even though it follows from Arrhenius theory, that increasing temperatures lead to an increase in reaction rates.

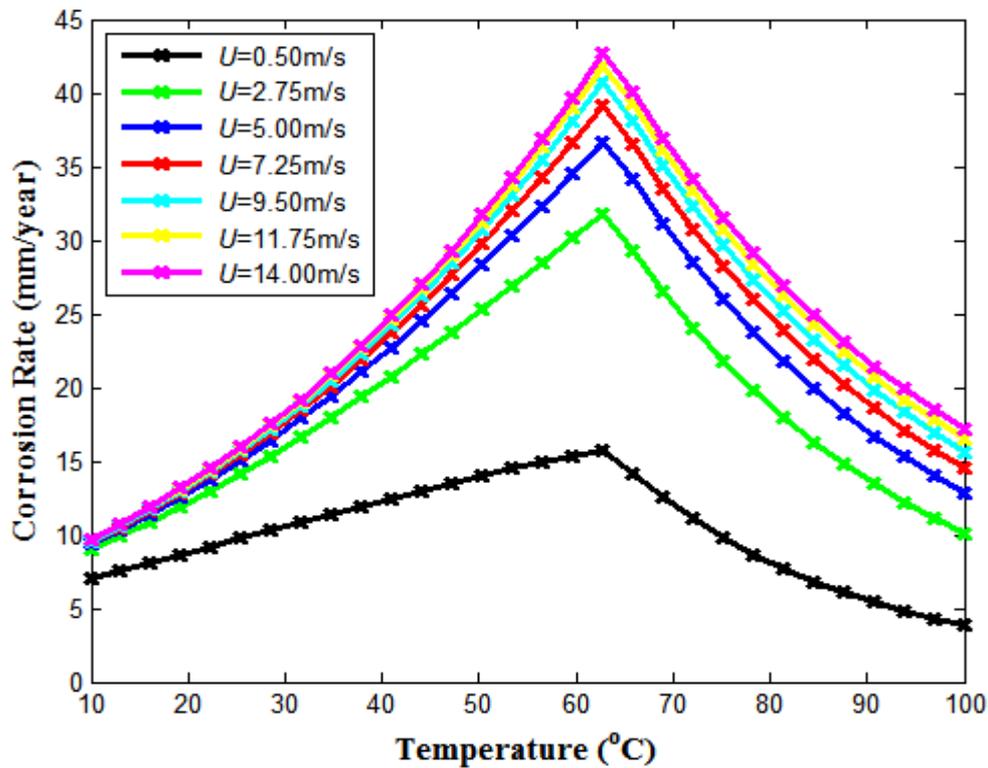


Figure 3-5. Plot of corrosion rate against temperature (Conditions are: $T=10-100^{\circ}\text{C}$, $f_{CO_2}=1\text{MPa}$, $\text{pH}=5.0$ and for varying flowrate velocity)

For the plot of corrosion rate against temperature for $T=10-100^{\circ}\text{C}$, $U=6.0\text{m/s}$, $\text{pH}=5.0$ and varying carbon dioxide partial pressures, Figure 3-6, there is an increase in corrosion rates with increasing temperatures until the scaling temperature is reached after which there is a drop in the magnitudes of corrosion rates. Carbon dioxide partial pressure to the power of 0.7 is said to have a direct proportional relationship with corrosion rate (Dugstad et al., 1994a), as seen in Figure 2-7A, Section 2.4.1 – there is an increasing trend for each curve in that the greater the CO_2 partial pressure, the greater the corresponding corrosion rate and this is a result of greater concentrations of carbonic acid forming. Given that carbonic acid readily dissociates into the bicarbonate ion and hydrogen ion, it implies that a greater concentration of carbonic acid will result in a greater concentration of hydrogen ions in solution thereby making the solution more acidic thus enhancing CO_2 corrosion.

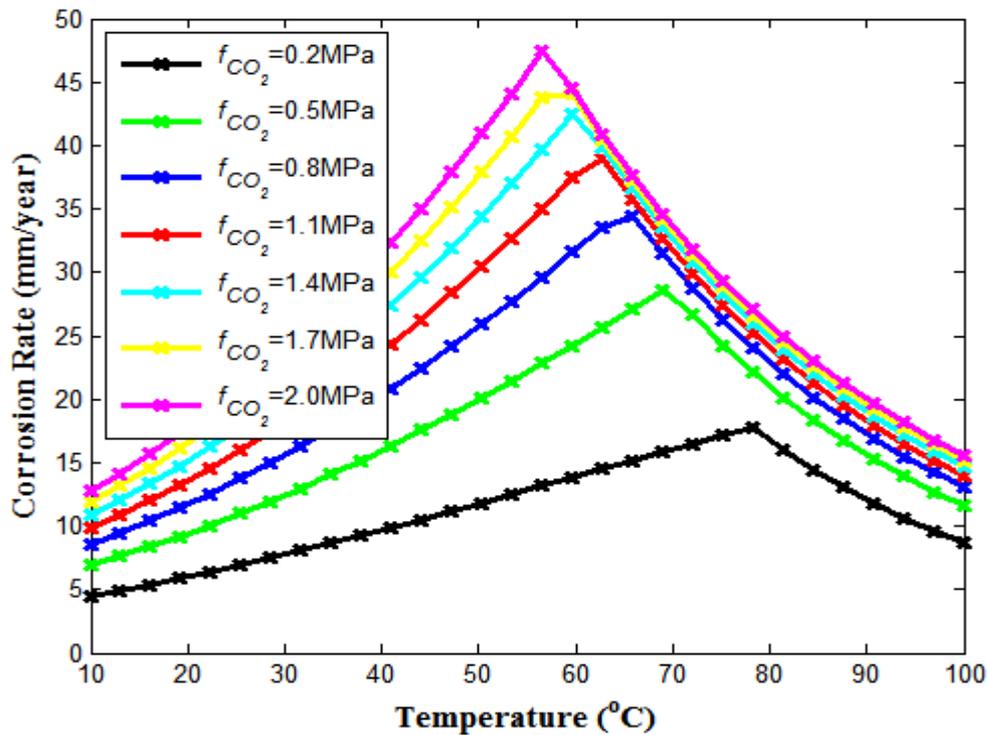


Figure 3-6. Plot of corrosion rate against temperature (Conditions are: $T=10-100^{\circ}\text{C}$, $U=6.0\text{m/s}$, $\text{pH}=5.0$ and for varying carbon dioxide fugacities)

3.5 The NORSOK M-506 Model

This is an empirical model developed by the Norwegian oil companies, Statoil, Norsk Hydro and Saga petroleum. It is originally based on laboratory data that was previously used to calibrate the de Waard model, however in addition it is capable of estimating corrosion rates at temperatures of up to 150°C (Nyborg, 2010). The model is said to yield ‘worst-case’ CO_2 corrosion rate results since the flow loop experiments upon which it is based assume low ferrous (Fe^{2+}) ion concentration in the aqueous phase (Olsen et al., 2005). The model equation can be summarised as follows Woollam and Hernandez (2006), Norsok Standard M-506, (2005):

$$C_{NOR} = K_t f_{\text{CO}_2}^{0.62} \left(\frac{S}{19} \right)^{0.146+0.0324 \log(f_{\text{CO}_2})} f(\text{pH})_t \quad (3.10)$$

Where:

K_t is a constant dependent on temperature

f_{CO_2} is CO_2 fugacity

S is wall shear stress (Pa)

$f(pH)_t$ is a complex function of pH and temperature

The model takes account of the formation of protective films at higher temperatures as seen in the plot in Figure 3-7. This Figure was obtained from the open-source Norsok M-506, accessed by a Microsoft Excel spreadsheet (coloured lines with markers). The NORSEK source code was then written, implemented and plotted separately using Matlab 2012a software (coloured lines without markers). These two plots were then superimposed together. Corrosion rates are observed to increase with increasing temperature after which there are peaks at a temperature of about 78°C followed by a dip in magnitudes. The peaks in corrosion rate are due to the effect of the presence of protective film layers on the surface of the metal which have the dual action of reducing the number of sites on the metal surface available to corrosive attack as well as forming a compact shield of ferrous carbonate on the metal surface that is neither easily swept away by fluid flow nor does it offer channels through its structure for the possibility of localised attack.

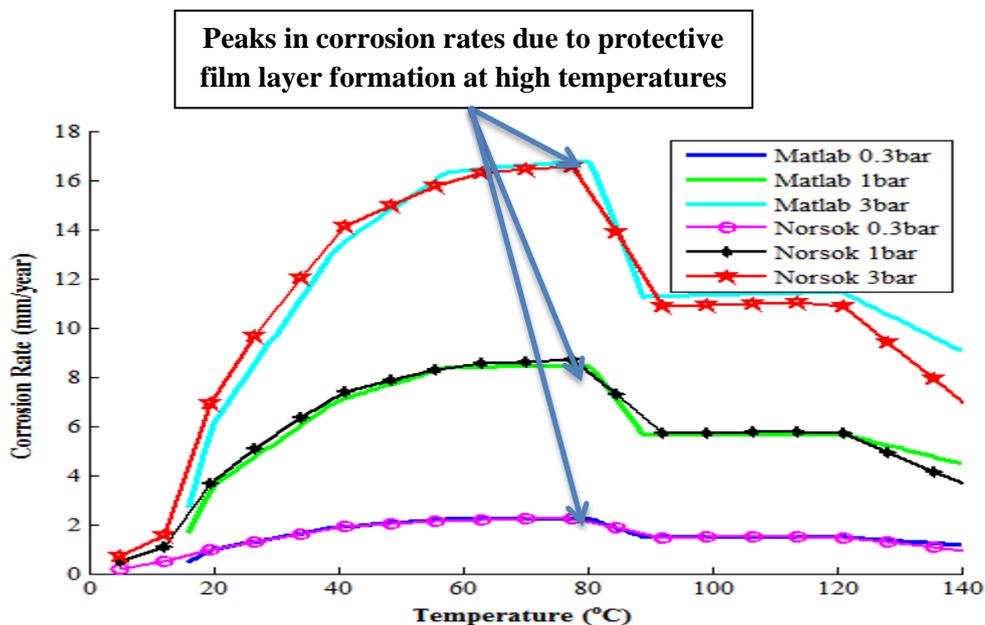


Figure 3-7. NORSEK M-506-generated and Matlab 2012a-generated plots of corrosion rate against temperature for varying CO_2 fugacities: 0.03MPa (0.3bar), 0.1MPa (1bar) and 0.3MPa (3 bar). Conditions are: pH=4, CO_2 mole =100%, Shear stress=2Pa

One limitation of the model is that it does not account for oil-wetting and its use is not applicable to systems in which pH stabilisation is used for corrosion control (Nyborg, 2010).

3.5.1 Matlab Model Results based on NORSOK Correlations

Figure 3-7 shows the Matlab 2012a model results (shown as blue, gree and cyan lines) overlaid on the Norsok M-506-generated results (shown as magenta circles, black crosses and red stars). The code was written based on identical parameters as indicated in published research literarture to validate Matlab 2012a-generated results. (Woollam and Hernandez, 2006). Other conditions required to obtain the plot shown in Figure 3-7 are tabulated in Table 3-1 as follows:

Table 3-1. Summary of variables and their respective magnitudes required to obtain results in figure 3-7

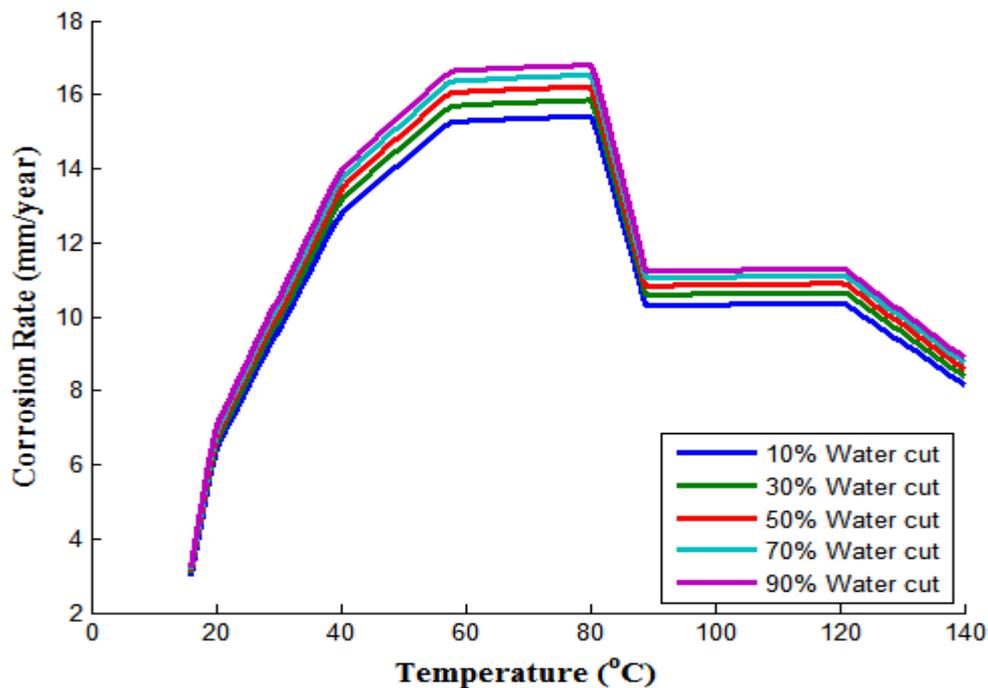
Parameter/unit	Magnitude
pH	4.0
Pipe Diameter (mm)	100
Water cut (%)	20
*Liquid volumetric flowrate (m ³ /s)	0.0307
CO ₂ mole (%)	100
*Superficial liquid velocity (m/s)	3.9063

* These parameters have been assumed. Superficial liquid velocity is calculated from an assumed value of liquid volumetric flowrate using the formula: $Velocity, U \left(\frac{m}{s} \right) = \frac{Volumetric\ flowrate\ (m^3/s)}{Area\ (m^2)}$. T_k/T_{std} was assumed to be approximately equal to unity given that the instantaneous temperature of the gas in Kelvin, T_k , is almost identical to the temperature of the gas at standard conditions, T_{std} .

All other parameters such as pipe roughness, oil specific gravity, density and viscosity of water, density and viscosity of oil as well as compressibility have been assigned identical values as the default Norsok M-506 corrosion rate model. There are slight differences in the superimposed plots and these are likely due to the assumed figures used in modelling.

3.5.2 Parameter Study on the NORSOK M-506 Model – Investigating Changes in Water Cut Percentage and CO₂ Mole Percentage on CO₂ Corrosion Rate

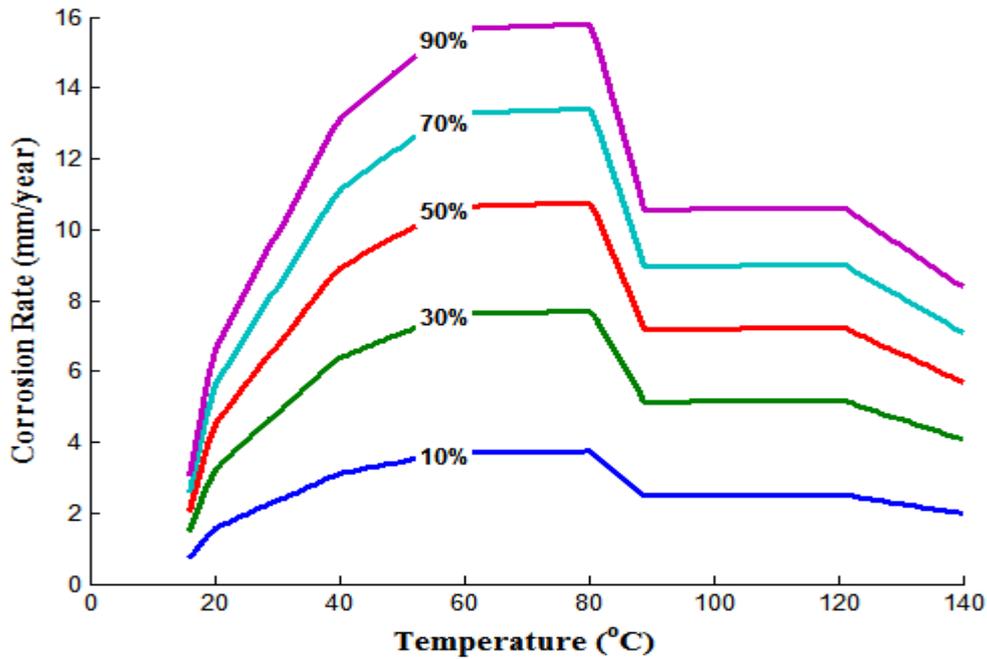
Upon observation of the plot in Figure 3-8, corrosion rates increase with increasing temperature as expected due to the reasoning behind the Arrhenius theory (Section 3.7.3); there is a temperature dependence on the rate constant hence the chemical reaction, CO₂ corrosion in this context is dependent on temperature change.



**Figure 3-8. Plot of carbon dioxide corrosion against temperature
(For varying water cut percentages)**

In general, the greater the water cut percentage in the fluid, the greater the rates of corrosion. This is explained by the relative ease of formation of carbonic acid when the water cut percentages are greater and the fact that the pressure or shear stress exerted on the metal surface or pipe walls is greater for conditions of high water cut. Fluid flow with a great water cut percentage, say 70%, implies that a bigger force is exerted per unit area of the pipe internal wall than the same fluid flow with 30% water cut, for instance. For the plot shown in Figure 3-9, corrosion rates increase with increasing CO₂ mole percentages due to the fact that there is an increased likelihood for the formation of carbonic acid in solution of a greater

concentration hence a CO₂ mole percentage of 90% would induce greater corrosion rates than one of 70%, for instance.



**Figure 3-9. Plot of carbon dioxide corrosion against temperature
(For varying CO₂ mole percentages)**

3.6 The Nescic-Postlethwaite-Olsen Model (NPO Model)

This is a mechanistic model that combines the equations for the cathodic reactions which include the reduction of H⁺ and the reduction of H₂CO₃ against a single current equation for the anodic dissolution of Fe. Hence, when the sum of cathodic current equations is said to be equal to the anodic current equation, therefore the rate of corrosion can be determined (Nesic et al., 1996; Woollam and Hernandez, 2006):

$$i_{Fe} = i_{H^+} + i_{H_2CO_3} \quad (3.11)$$

i_{H^+} , $i_{H_2CO_3}$ are the cathodic current terms for hydrogen ion and carbonic acid respectively

i_{Fe} is the anodic current term for the dissolution of the metal (Fe)

3.6.1 Cathodic Reactions

For these reactions, the electrochemical process is modelled by a resistance model, in which the inverse of the current function is equal to the sum of the inverses of the activation current and limiting current (Woollam and Hernandez, 2006).

$$\frac{1}{i_{(H^+)}} = \frac{1}{i_{\alpha(H^+)}} + \frac{1}{i_{lim(H^+)}}^d \quad (3.12)$$

Where:

$i_{\alpha(H^+)}$ is the activation controlled current

$i_{lim(H^+)}^d$ is the limiting current

The term, $i_{lim(H^+)}^d$, is due to mass transfer kinetics playing an important role in the transport of species to the metal surface from bulk solution.

$$i_{lim(H^+)}^d = k_m F [H^+]_b \quad (3.13)$$

Where:

k_m = Mass transfer constant

F = Faraday constant

$[H^+]_b$ = Concentration of hydrogen ions in bulk solution

The activation controlled current is given by the following expression (Woollam and Hernandez, 2006):

$$i_{\alpha(H^+)} = i_{o(H^+)} \cdot 10^{-\eta/b_c} \quad (3.14)$$

η is the overpotential (Volts, V)

$i_{\alpha(H^+)}$ is the exchange current density (A/m²)

b_c is the cathodic Tafel slope on the Tafel plot

The cathodic current equation for carbonic acid is expressed in a similar manner as that of hydrogen ions, H⁺.

3.6.2 Anodic Reactions

This reaction is assumed to be strictly activation-controlled.

$$\text{Hence, } i_{(Fe)} = i_{o(Fe)} \cdot 10^{\eta/b_a} \quad (3.15)$$

η is overpotential (Volts, V)

$i_{o(Fe)}$ is exchange current density (A/m^2)

b_a is anodic Tafel slope on the Tafel plot

These equations are used to derive the plot shown in Figure 3-10. This is a Tafel plot of potential against the log of current density and the corrosion rates are read off from the point of intersection of $E_{corr} = -513\text{mV}$ and the anodic dissolution line (A1). The current density is then read off from the horizontal axis as the value of i_{corr} . This current is then converted into units of corrosion rate in mm/year by the following expression in order to obtain the plot shown in Figure 3-11.

$$\text{Conversion of Current density to Corrosion rate: } \frac{1.155A}{m^2} = \frac{1\text{mm}}{\text{year}} \quad (3.16)$$

The plot shown in Figure 3-10 was produced for a temperature of 100°C and a CO_2 fugacity of 0.3 MPa (3bar). The temperature was then changed accordingly in the model to determine the corresponding corrosion rates. The CO_2 fugacity was then changed to 0.1MPa (1 bar), 0.03MPa (0.3 bar) and 0.01MPa (0.1 bar) and the same procedure was repeated to obtain the plot shown in Figure 3-11.

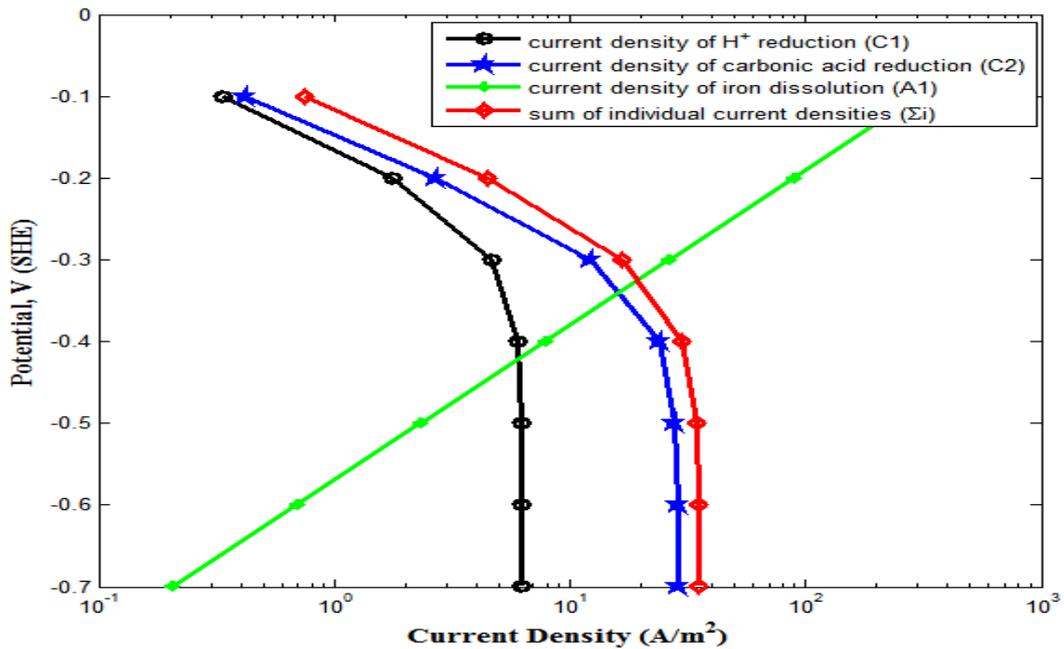


Figure 3-10. Nescic-Postlethwaite-Olsen (NPO) model output for $T=100^{\circ}\text{C}$ – Tafel plot of potential against \log_{10} (current density)

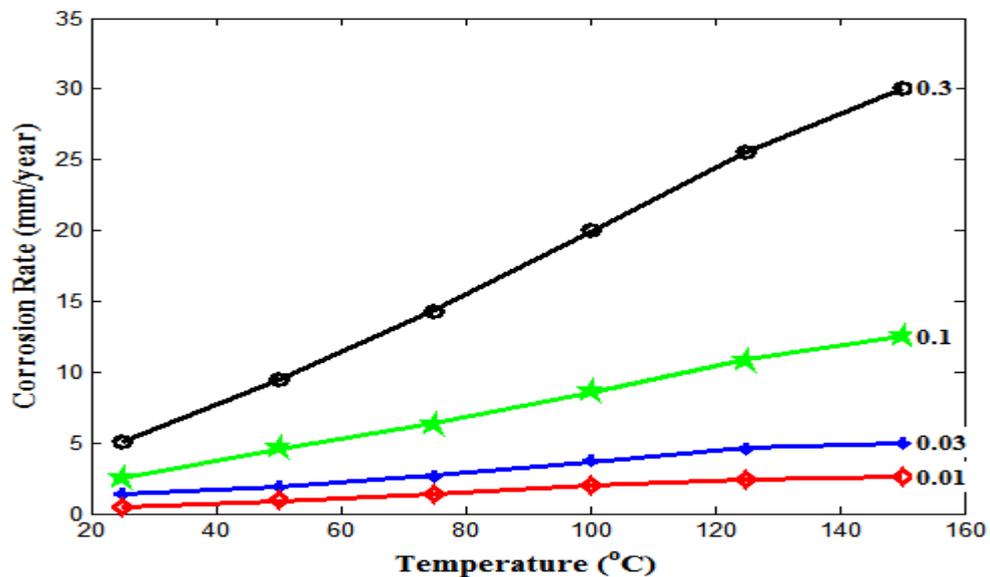


Figure 3-11. Plot of corrosion rate against temperature for the NPO model for CO₂ fugacities: 0.01MPa (0.1bar), 0.03MPa (0.3bar), 0.1MPa (1bar) and 0.3MPa (3 bar). Other conditions where applicable are: pH=4, CO₂ mole =100%, Shear stress=2Pa

From Figure 3-11, it is observed that corrosion rates generally increase with increasing temperature which is as expected following the explanation given by the Arrhenius theory for the effect of increasing temperature on increasing the rate of chemical reactions. It is also observed that the corrosion rates are greater for CO₂ fugacities of higher magnitudes in the following order: f_{CO_2} (at 0.3MPa) f_{CO_2} (at 0.1MPa) > f_{CO_2} (at 0.03MPa) > f_{CO_2} (at 0.01MPa).

The reason for this is that higher magnitudes of CO₂ fugacity translate into greater concentrations of the gas in solution and consequently higher concentrations of carbonic acid hence enhanced corrosion risks and higher anodic dissolution rates of the metal.

3.7 Introduction to Freecorp

The Freecorp version 1.0 corrosion model is a simple point model developed exclusively based on freely accessible information by corrosion researchers at Ohio University (Ohio University, 2008). A point model is a simulator that is capable of predicting uniform corrosion rates for the following species: carbon dioxide (CO₂), oxygen (O₂), acetic acid (CH₃COOH or HAc) and hydrogen sulphide (H₂S) at a single point (Nyborg, 2002) within the given environment as opposed to a combined fluid flow-corrosion model which would be capable of predicting the corrosion profile at different locations in a pipeline or corrosion system (Nyborg, 2002).

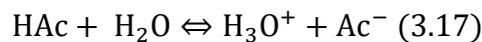
In terms of Freecorp version 1.0's ability to accurately estimate corrosion, it has been designed with a mechanistic background at its core (Ohio University, 2008). The model is capable of indicating the relative contributions to corrosion as well as the dominant corrosion mechanism based on the corrosive species input by the user. In addition, polarisation sweeps and polarisation curves for each individual electrochemical reaction are displayed.

3.7.1 Organic Acid Corrosion

Acetic acid (CH₃COOH) is the most commonly occurring of the organic acids that are typically seen in the water phase during oil and gas production. Other forms of organic acids include formic, acetic and propionic acids. It is known that there are only small significant differences between the corrosiveness of the aforementioned acids hence acetic acid corrosion is considered to be the broad representative for these organic acids since it is the most prevalent.

There is documented evidence of acetic acid contributing to top-of-line corrosion (TLC) (Nesic et al., 2009). TLC is a form of CO₂ corrosion that occurs along the top of wet gas pipe internals as a result of condensation of water followed by the rapid saturation of corrosion products in the aqueous medium (Dugstad et al., 1994a). One limitation of Freecorp is that while it is able to predict CH₃COOH corrosion, it makes no provision for TLC estimation.

Acetic acid is of a major concern in top-of-line corrosion due to its ability to provide an additional source of hydrogen ions brought about by its weak dissociative property. It partially dissociates into a hydrogen ion and an acetate ion as shown below:



Apart from being a reservoir for hydrogen ions, it has also been discovered experimentally that the undissociated CH₃COOH specie can be directly reduced after adsorbing onto the metal surface (Nesic et al., 2009).



Thus, the presence of free CH₃COOH becomes problematic, in that, particularly at low pH ranges, the equilibrium of the dissociation reaction shifts to the left. It is also said that the reduction of free acetic acid is strongly affected by the velocity of fluid flow indicating that it is a mass transfer-controlled process and this implies that corrosion rate is dependent upon the acetic acid concentration and the kinetics of transport of the species from the bulk solution to the metal surface (Nesic et al., 2009).

Figure 3-12 shows a scatterplot of Freecorp model results for acetic acid corrosion against experimental results. The two datasets, Crolet et al., 1999 and Guo et al., 2005, are the results of the investigation of CO₂-CH₃COOH corrosion on X65 carbon steel and N80 carbon steel respectively. Corrosion rates were determined by varying the concentrations of acetic acid for both studies. Experiments were carried out at temperatures of 22°C for Crolet et al., 1999 and 50°C for Guo et al., 2005. Although the CO₂ concentration in the former was constant, 1bar (0.1MPa), there were two magnitudes for CO₂ concentration in the latter case: 1 (0.1MPa) and 10bar (1MPa).

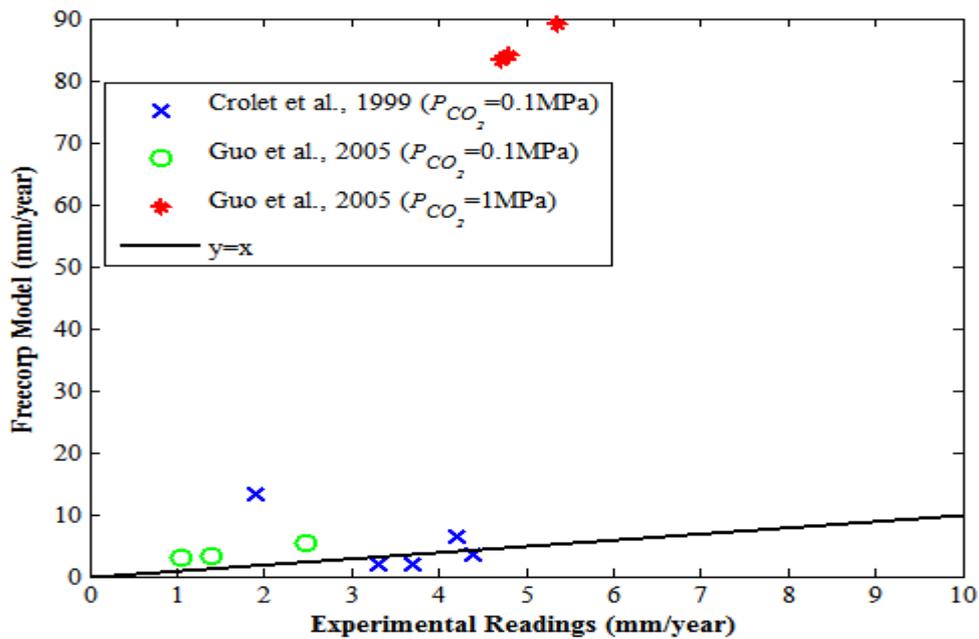


Figure 3-12. Scatterplot of Freecorp model readings against experimental readings for two datasets: Crolet et al, (1999) and Guo et al, (2005)

The model over-predicts all data points of the Guo et al. (2005) study while the $y=x$ line splits the Crolet et al. (1999) dataset roughly into two, particularly for high HAc concentrations. High HAc concentrations correspond to the experimental readings of the greatest magnitude (≈ 5 mm/year) and as experimental readings increase in magnitude, disproportionately high readings are observed for model predictions. The Freecorp model predictions for the Guo et al., 2005 dataset for 1MPa (shown in red crosses) greatly exceed the corresponding experimental readings even though higher than normal corrosion rates are expected as a result of the CO_2 partial pressure being equal to 1MPa (10bar).

3.7.2 Hydrogen Sulphide Corrosion

H_2S corrosion proceeds via solid state reaction – the initial and final states of iron exist in the solid state and unlike CO_2 and HAc corrosion, it is not electrochemical. There is no significant separation of the oxidation and reduction reactions at the steel surface. Hence, no current flows between the cathode and anode (Nesic et al., 2009).



Freecorp version 1.0 models H_2S as a chemical reaction showing the species concentration profile for H_2S concentration from the metal surface across the liquid boundary layer to the bulk solution as shown in Figure 3-13. The following concentration profile is representative of the following conditions: $T=20^\circ C$, $P_{total}=0.1MPa$ (1bar), $P_{CO_2}=0.01MPa$ (0.1bar), $H_2S(g)=40ppm$ and $pH=5$.

In general, H_2S corrosion is limited to the kinetics of diffusion of corroding species (H_3O^+ and H_2S) from the bulk solution across the boundary layer and onto the metal surface. A concentration gradient is set up across the boundary layer whereby actively corroding species at the metal surface induce a drive resulting in the influx of more corroding species from the bulk solution to the metal surface.

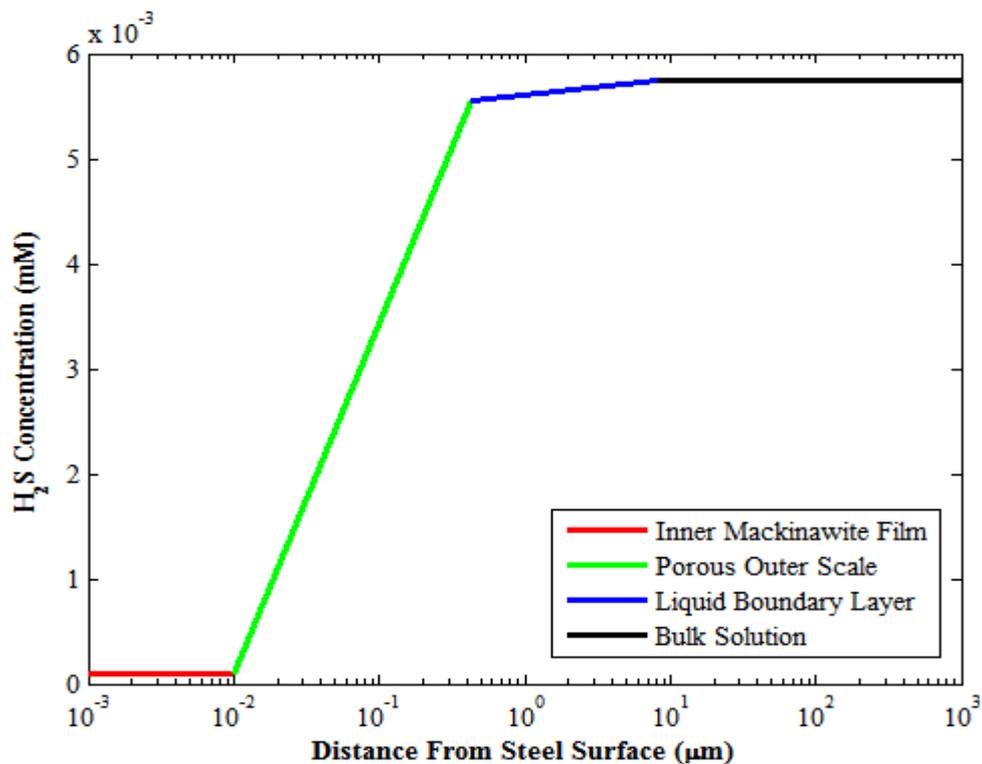


Figure 3-13. Species concentration profile - H_2S concentration as a function of the distance from steel surface

Mackinawite, a thin layer of sulphide scale formed by a rapid reaction between H_2S and iron acts as a solid state diffusion barrier however diffusion of corroding species does not cease altogether, instead as corrosion proceeds, more mackinawite is formed thus thickening the scale layers which eventually results in spalling and microcracking and so a cyclic process of growth, cracking and delamination ensues (Sun and Netic, 2007; Netic et al., 2009).

As corrosion proceeds over time, mackinawite scales thicken and become more protective of the metal surface beneath and in some instances, may lead to the formation of pyrrhotite, another less soluble form of ferrous sulphide scale. In cases where, the H₂S concentrations are high, pyrite and elemental sulphur form on the metal surface (Nesic et al., 2009).

The reason for there being different forms of iron sulphide is attributed to the fact that they vary in terms of thermodynamic stability. Troilite and pyrrhotite are more stable than mackinawite, for instance, combined with the fact that the equilibrium ferrous ion concentration and pH are greater on the metal surface than in the bulk solution (Benning et al, 2000; Criaud et al, 1989).

Figure 3-14 shows a scatterplot for two sets of data: Lyle and Schutt (1998) and Nesic et al (2008). While overall, the model estimates corrosion fairly accurately, for both datasets, it tends to under-predict experimental readings of larger magnitudes. For instance, the model predicts less accurately for the last two data points (shown in green crosses) from the Lyle and Schutt (1998) study, which investigated H₂S and pitting corrosion for a constant temperature of 15.6°C, at H₂S pressures of the range: 0-0.013MPa (0-0.13bar), CO₂ partial pressures (0-0.069MPa), for a pH range of 3.85 to 4.75 and for a duration of 14 days. The less accurate model predictions correspond to the data points with low pH values (3.8-4.0) as well as with the high H₂S gas phase pressures (0.0069-0.013MPa) while the varying CO₂ pressures do not appear to have any significant impact on trends. Hence, the model struggles with corrosion estimations for high H₂S pressures and very low pH ranges.

A similar explanation can be used to describe model predictions on the Nesic et al., 2008 study, as well. Here, H₂S corrosion was investigated at a constant temperature (20°C), pH (5) and CO₂ partial pressure (0.01MPa), with varying H₂S concentrations (0-180ppm) and for a period of 24 hours. The main difference between the two studies is that the pH was constant for the Nesic et al (2008) study while it was varied for the Lyle and Schutt study. While the CO₂ partial pressure varies for the former and is constant in the latter, CO₂ does not influence the trends in these studies.

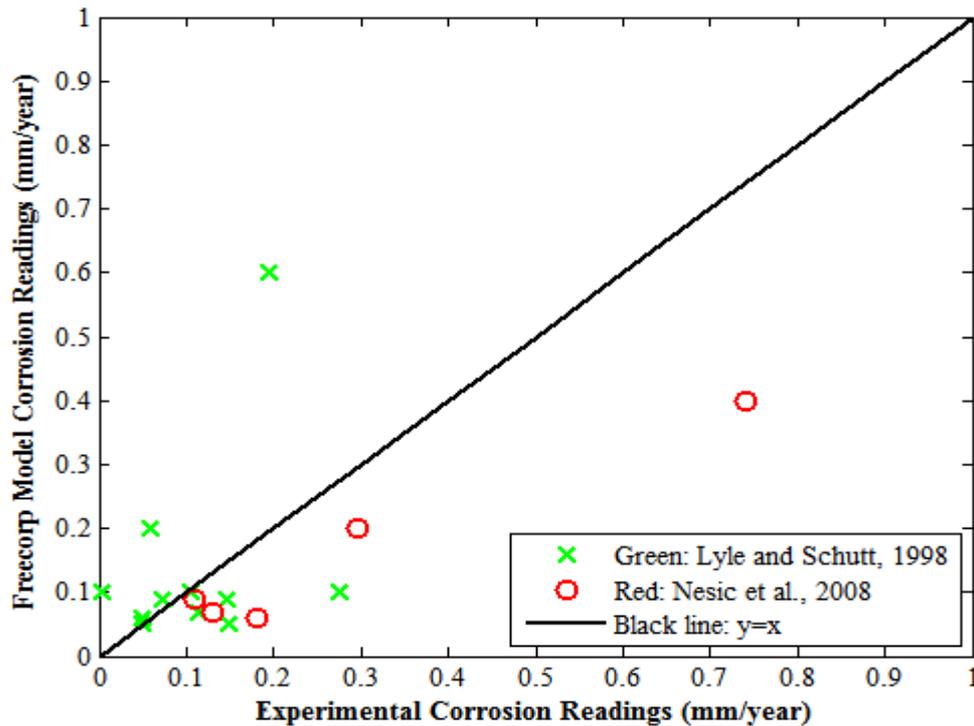
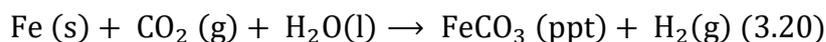


Figure 3-14. Scatterplot of Freecorp model readings against experimental readings for two datasets: Lyle and Schutt (1998) and Nestic et al. (2008)

3.7.3 CO₂ Corrosion Rate Trends and Parametric Study on Velocity of Fluid Flow

Corrosion rates increase with increasing temperature. This is explained by the kinetic theory in that greater temperatures provide larger amounts of energy to colliding molecules in a given reaction hence the frequency of collisions and the likelihood of the formation of a product increases. In terms of CO₂ corrosion, this implies that an increase in temperature results in increased reaction rates, thus an increased tendency to form corrosion products such as the evolution of hydrogen gas, the dissolution of iron to ferrous ions as well as the formation of carbonate scales. This reaction is summarised as follows (Nestic and Vrhovac, 1999):



The Arrhenius theory also follows from the Kinetic molecular theory and further emphasises this point (Tan and Chan, 2011). The Arrhenius equation is stated as follows:

$$k = Ae^{-\frac{E_A}{RT_k}} \quad (3.21)$$

Where:

k = Reaction Rate Coefficient

A = Pre – exponential factor

E_A = Activation Energy (KJ/mol)

R = Molar Gas Constant

T_k = Absolute Temperature (K)

By the Arrhenius theory, reacting molecules will have to acquire a minimum threshold energy (activation energy, E_A) in order to form products and temperature provides this energy (Tan and Chan, 2011). Thus, from the definition of the Arrhenius equation, the natural logarithm of the reaction rate constant, $\log_e k$, is proportional to the inverse of the absolute temperature of the reaction, implying that the rate constant for any given reaction increases with increasing temperature (Tan and Chan, 2011).

It is observed from Figure 3-15, that corrosion rates generally increase with temperature and this is in accordance with the kinetic theory and the Arrhenius equation since it accelerates all processes involved in corrosion – species transport, electrochemical and chemical reaction rates.

Corrosion rates increase quite markedly as temperatures increase without accounting for the effect of scale formation which results in peaks usually between 60-80°C depending on flow conditions and water chemistry as well as the formation of Fe_3O_4 at higher temperatures. It is also worth noting that even when ferrous ion concentrations are included in the simulation, $[\text{Fe}^{2+}] = 50\text{-}120\text{ppm}$, the result is simply a decrease in the overall magnitude of the corrosion rates with no distinctive peak whatsoever (de Waard et al., 1991).

It is also observed that from Figure 3-16, the greater fluid velocities result in greater magnitudes of corrosion rate (Corr Rate), thus Corr Rate at $U=1.0\text{m/s} >$ Corr Rate at $U=0.5\text{m/s} >$ Corr Rate at $U=0.1\text{m/s}$. The reason for this is due to the fact that iron carbonate scales deposited on the surface of the metal have an inhibitory effect to corrosion as they build up and thicken however with greater fluid velocities, there is a greater tendency for these scales to be eroded and washed away hence exposing a greater area of the metal surface to further corrosive attack.

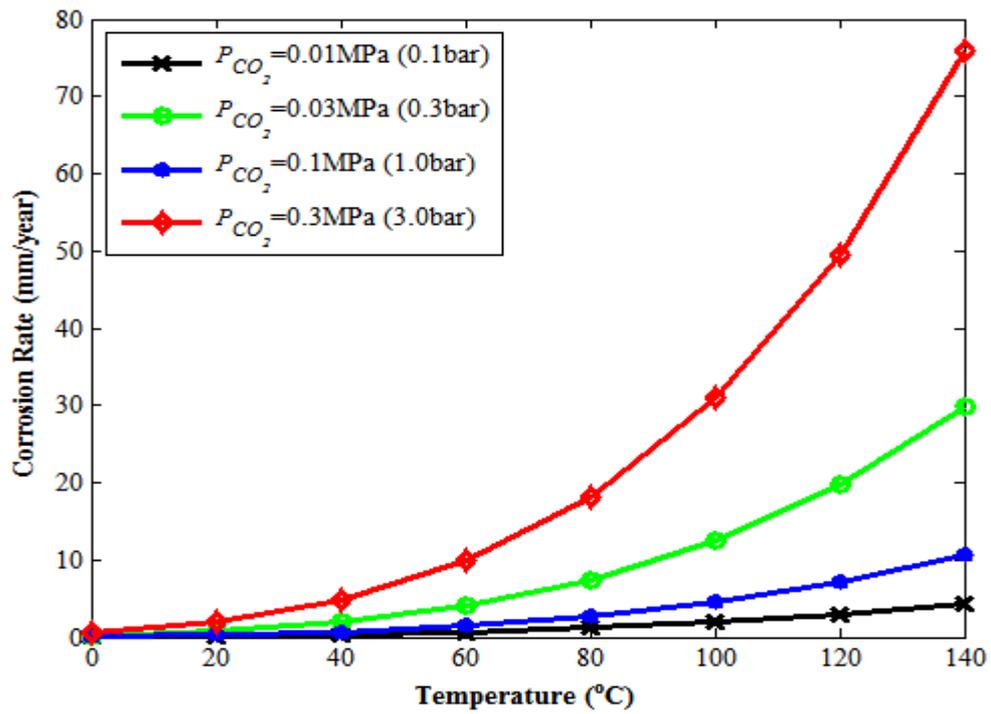


Figure 3-15. Plot of corrosion rate against temperature for varying CO₂ partial pressures for a fluid velocity, $U=0.1\text{m/s}$

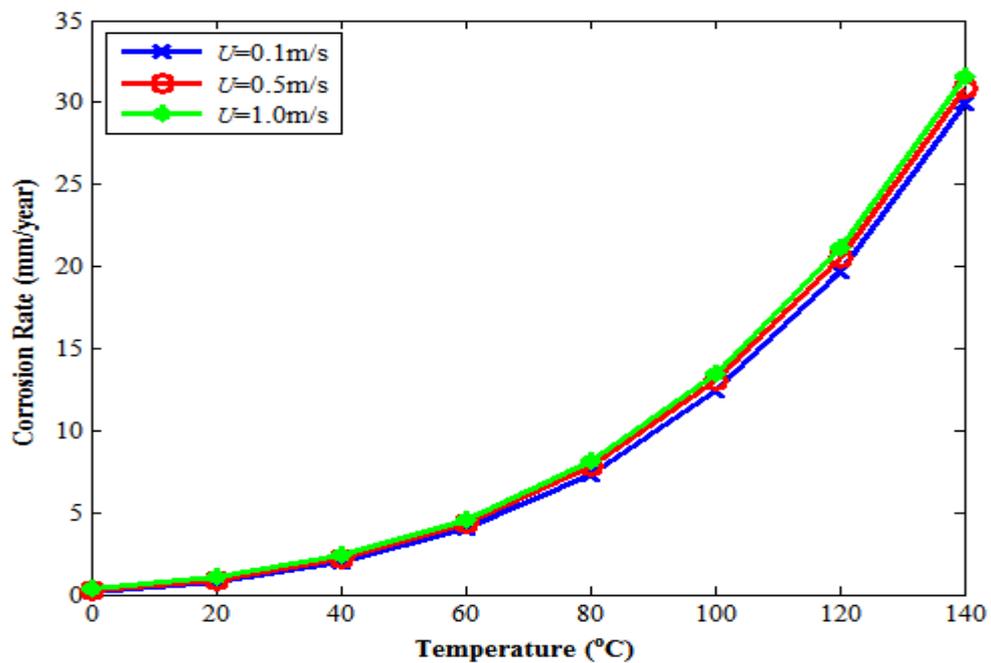


Figure 3-16. Plot of corrosion rate against temperature for varying fluid velocities

3.8 Conclusions

The comparison of the different models discussed was based on identical values for the underlying physical and chemical conditions that affect CO₂ corrosion – the parameters common to all these models were the temperature, CO₂ fugacity (partial pressure), flow velocity and pH.

The 1975 De Waard correlation is the most basic function. CO₂ corrosion increased with increasing temperatures, as do all of the other models. However it does not account for mid-temperature scaling and this is its main limitation. The 1991 and 1995 De Waard correlations account for mid-temperature scaling but the former does not explicitly contain a velocity term in its function whereas the latter incorporated a velocity term as part of its resistance model. The 1995 De Waard model also made use of a pH correction as part of the activation reaction kinetics term.

The Norsok model also exhibited similar CO₂ corrosion rate-temperature profile plots as the De Waard models whilst accounting for shear stress in pipe flow. An interesting feature was the scaling temperatures and how they varied across the different models. Table 3-2 shows the scaling temperatures for each of these models.

Table 3-2. Scaling temperature maxima for the various models at $f_{CO_2}=0.3\text{MPa}$

Model	Peak temperature, T _{scale} (°C)
1975 De Waard-Milliams	Continuous Increase
1991 De Waard-Lotz-Milliams	71
1995 De Waard-Lotz-Dugstad	75
Norsok	78
NPO	Continuous Increase

For the 1991 and 1995 De Waard correlations, as the CO₂ fugacity decreased from 0.3MPa to 0.01MPa, the scaling temperatures increased consequently, which is depicted in Figures 3-3 and 3-4 respectively. Film formation is not included in both the 1975 De Waard and NPO models hence there are no peaks in their corrosion rate-temperature profiles, as indicated in Table 3-2. Film formation is included in the Norsok model correlations, and so a peak is depicted in its corrosion rate-temperature profile. However, this peak remains fairly constant regardless of changes in the magnitude of CO₂ fugacity, as seen in Figure 3-7. While for the

1991 and 1995 De Waard models, changes in the magnitude of CO₂ fugacity consequently leads to changes in the magnitude of the corresponding scaling temperatures as seen in Figures 3-3 and 3-4, respectively. A comparison of all the investigated corrosion models is shown in Figure 3-17.

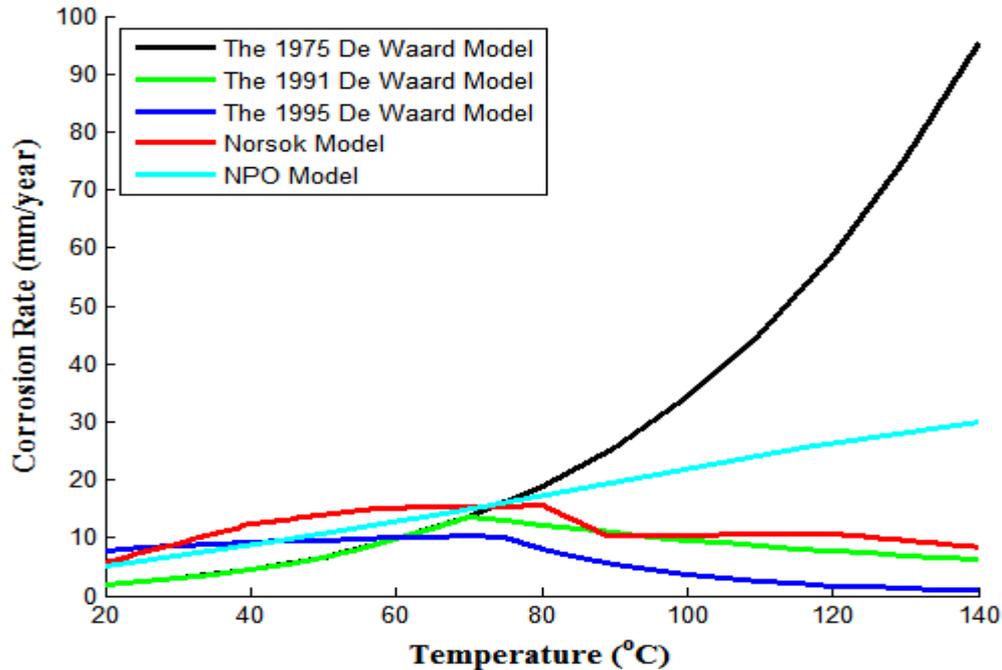


Figure 3-17. Plot of CO₂ corrosion rate against temperature for the various models for $f_{CO_2}=0.3\text{MPa}$ (3bar)

From the plot, all the model predictions appear to be in close proximity to each other for the temperature range of 20-80°C but after 80°C, their behaviours are markedly different. While the 1991 De Waard, 1995 De Waard and Norsok model predictions decrease once their respective scaling temperatures are exceeded, the 1975 De Waard and NPO models continue to increase due to the fact that as previously mentioned, their correlations do not account for film formation. Therefore, their accuracy in prediction is limited to the low to mid-temperature range only.

In addition, the Norsok model's predicted results are consistently higher than those of the De Waard's correlations for most temperatures, due to the fact that the applied shear stress of 2Pa, as seen in Figure 3-7, is in the lower end of its applicable range of 1-150Pa. The NPO model is mechanistic and electrochemical in its basis and as such has a tendency for over-

prediction, particularly for temperatures exceeding 80°C. However, it must be said that while the empirical models such as the De Waard correlations and the Norsok model may appear to predict more accurately than the mechanistic NPO, drawing conclusions from results predicted outside their range of applicability is questionable. Mechanistic models are more likely, in general, to better model a wider range of parameters than their empirical counterparts. Overall, based on the corrosion rate vs temperature plot trends, the 1991, 1995 and Norsok models are the more realistic ones, with porous Fe₂O₃ porous films forming before peak temperatures and Fe₃O₄ passive films forming beyond peak temperatures.

For the Freecorp version 1 model, in predicting CO₂ corrosion, the model does not account for Fe₃C, FeCO₃ and Fe₃O₄ scaling and the implication of this is that corrosion rates continuously increase with increasing temperatures. Despite a ferrous ion concentration, [Fe²⁺], provision in the model, corrosion rates still increase with increasing temperatures even though [Fe²⁺] in excess of 80ppm should lead to supersaturation that will eventually cause the build-up of scale (de Waard et al., 1991), culminating in a peak-like trend as depicted by the Norsok model, for instance. The model is also limited to uniform corrosion hence it is incapable of estimating pitting or any form of localised corrosion. Also, the range of CO₂ prediction is from 1Pa (0.01mbar) to 1MPa (10bar).

In HAc corrosion prediction, model predictions are fairly accurate provided HAc concentrations do not greatly exceed 0.01M (600ppm), for which it grossly over-predicts. Also, the model makes no provision for TLC.

In H₂S corrosion prediction, the model seems to perform fairly well although as with HAc, the greater the concentrations of H₂S, the less accurate are the model predictions. The model's maximum limit is a H₂S partial pressure of 1MPa (10 bar). The model does not account for the precipitation of ferrous sulphide hence is incapable of identifying the likeliest form of ferrous sulphide scale for a given flow condition given that there is a high tendency for the formation of a mixture of different scales on the steel surface. The composition of the scale will affect the species concentration profile from the metal surface across boundary layer to the bulk solution because the scale acts as a barrier to the transport of corroding species. These models were assessed in terms of their accuracy of predicted corrosion rates using a separate test dataset. The results are discussed in Section 8.2.

Chapter 4. Statistical Analysis

4.1 Introduction

In the previous chapter, established CO₂ corrosion models in research literature such as the De Waard-Milliams correlations, Norsok and NPO models were discussed. Though these models are satisfactory for their intended purpose – low pressure CO₂ corrosivity prediction, their limitations are their unsuitability for high pressure CO₂ corrosion prediction (Mohamed et al., 2011) and when more complex effects are present such as the growth of protective scale (Nesic et al., 2008).

In this chapter, low and high pressure CO₂ corrosion datasets are analysed using various statistical techniques. The statistical techniques applied include descriptive statistics, principal component analysis (PCA), variable interactions, regression and response surface modelling. PCA is a mathematical procedure which transforms potentially correlated data into an orthogonal system of linearly uncorrelated principal components (Suryanarayana and Mistry, 2016). PCA is carried out in such a way that the first principal component accounts for much of the variability within the dataset. The datasets comprise the primary environmental factors: temperature (T), CO₂ partial pressure (P_{CO_2}), flowrate velocity (U), pH and corrosion rate (Corr Rate). A smaller test dataset was then separately selected from related sources and used as unseen data in the prediction of CO₂ corrosion rates. Assessments of the performance of each model were then made by plotting model vs experimental scatter diagrams and determining the difference (error) in magnitudes between each of the plotted points. In addition, the R^2 -value and 95% confidence intervals were also used in the evaluation of model accuracy and reliability.

4.2 Low Pressure CO₂ Corrosion Data

4.2.1 Background on Datasets and Descriptive Statistics

For the low pressure CO₂ data (Dugstad et al., 1994b, Nordsveen et al., 2003) a parametric research investigation was carried out on CO₂ corrosion rates of ferritic-pearlitic carbon steel St-52. As reported in the literature, experiments were carried out under strict control of the water chemistry in a high pressure corrosion testing loop. Parameters such as temperatures (20-90°C), CO₂ partial pressures (0.04-2.10 MPa), pH (3.4-4.15) and flow velocities (0.1-13.0 m/s) were varied under conditions in which protective iron carbonate films did not form. The test loops were carried out in an 80mm internal diameter (ID) high-velocity flow loop. In all, 75 data-points were collected.

A summary of the descriptive statistics for each variable is displayed in Table 4-1.

Table 4-1. Descriptive statistics of the CO₂ corrosion sample data

Variable	Range	Mean	First Quartile (Q1)	Median (Q2)	Third Quartile (Q3)	Skewness
Temperature (°C)	20-90	52.50	25.00	50.00	82.50	0.25
CO ₂ Partial Pressure, P_{CO_2} (MPa)	0.04-2.10	0.51	0.13	0.22	0.63	1.54
Velocity (m/s)	0.1-13.0	7.20	3.10	8.50	13.00	-0.08
pH	3.40-4.15	3.80	3.63	3.80	4.05	0.20
Corrosion Rate (mm/year)	0.5-70.0	17.03	4.85	9.00	23.00	1.46

The boxplots in Figure 4-1 shows the distribution of each of these variables. The boxplots for temperature and pH data have approximately symmetrical data distributions, implying that the data within each of these variables is distributed evenly around the measures of central tendency (mean, median and mode). In general, the input variables: T , P_{CO_2} , pH and the output variable, Corr Rate, exhibit positive skewness, implying that their means are not substantially greater than their respective medians and modes. For pH, the skewness is 0.20 therefore, the mean, median and mode are approximately equal in magnitude. Velocity exhibits a negative skew, hence its mode is greater than its respective median and mean values. In the boxplots of Figure 4-1 and Figure 4-2, the closer the horizontal centre-lines of each variable to the bottom of the box is, the greater the degree of positive skewness and the closer the horizontal centre-line is towards the top of the box, the greater the degree of negative skewness. The red plus symbols above the corrosion rate boxplot indicate that the outliers for this dataset are in the 60-70mm/year range. There are outliers for the P_{CO_2} variable as shown in Figure 4-2, however the range is not as significant as that of the corrosion rate variable.

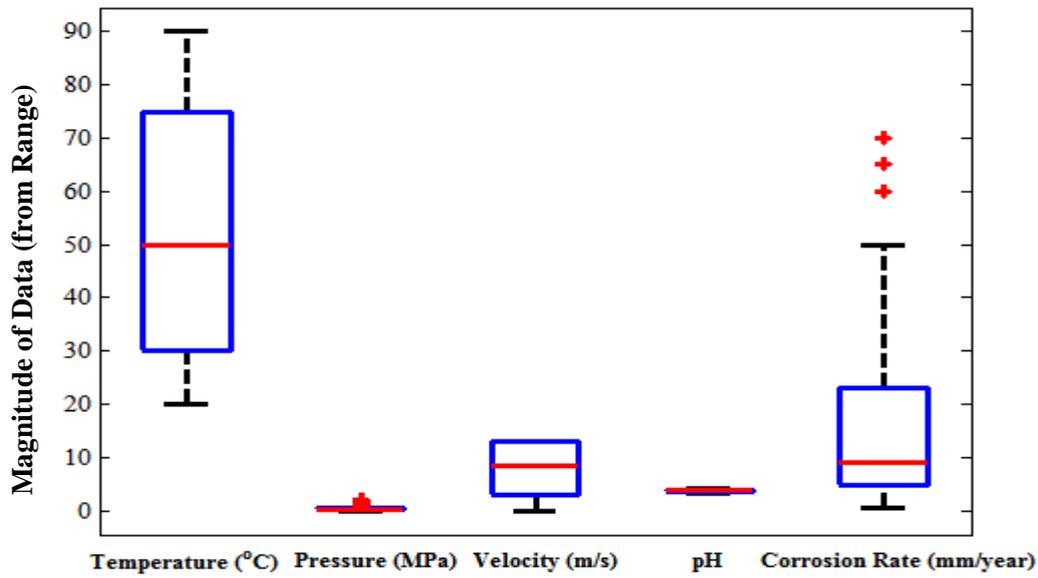


Figure 4-1. Boxplot showing the overall range (box and whiskers), inter-quartile range (white space in blue edged box), median, skewness and outliers (red plus symbols) in each of the variables in the dataset

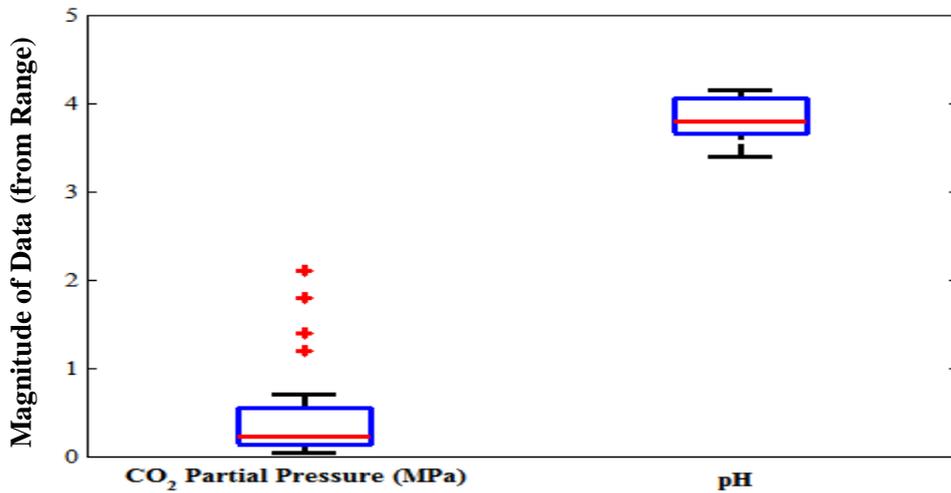


Figure 4-2. Boxplot showing the range of values, inter-quartile range, median and skewness of the P_{CO_2} and pH variables (Outliers are red plus symbols)

A summary of the test dataset used in this study is shown in Table 4-2. The number of data-points in this dataset is 15. This dataset is used for validation of the model and is not included in the development of the final model.

Table 4-2. Test data variables and their corresponding ranges

Variable	Range
Temperature, T (°C)	20-90
P_{CO_2} (MPa)	0.13-0.65
Fluid flowrate, U (m/s)	3.1-13.0
pH	3.60-3.90
Corrosion Rate (mm/year)	6.7-60.0

This dataset is a randomly selected group of CO₂ corrosion data (Dugstad et al., 1994b; Nordsveen et al., 2003). It is used as unseen data in each of the developed models in order to assess their performances in CO₂ corrosion prediction.

4.2.2 Principal Component Analysis

A weighted principal component analysis of the dataset was carried out to account for variations in units and scales of each of the variables: T , P_{CO_2} , U and pH. In principal component analysis, the first 2-3 components account for the majority of the variation in the dataset (Jackson, 1991), (Jolliffe, 2002), for instance, as shown by the magnitudes of the eigenvalues of the four principal components in Table 4-3 and the Pareto chart in Figure 4-3.

Table 4-3. Principal component analysis

Principal Component	Principal Component 1 (PC1)	Principal Component 2 (PC2)	Principal Component 3 (PC3)	Principal Component 4 (PC4)
Eigenvalue (Variance)	2.016	1.023	0.811	0.151
Proportion	0.504	0.256	0.203	0.038
Cumulative	0.504	0.760	0.962	1.000
Cumulative Percentage (%)	50.400	76.000	96.200	100.000

It is seen that from the cumulative figures, 96.2% of the variation in this dataset, is explained by the first 3 principal components, and so the fourth is not necessary to describe the variation in this particular dataset. This further implies that three variables are sufficient in describing CO₂ corrosion rate. The Pareto chart shown in Figure 4-3 illustrates this point further.

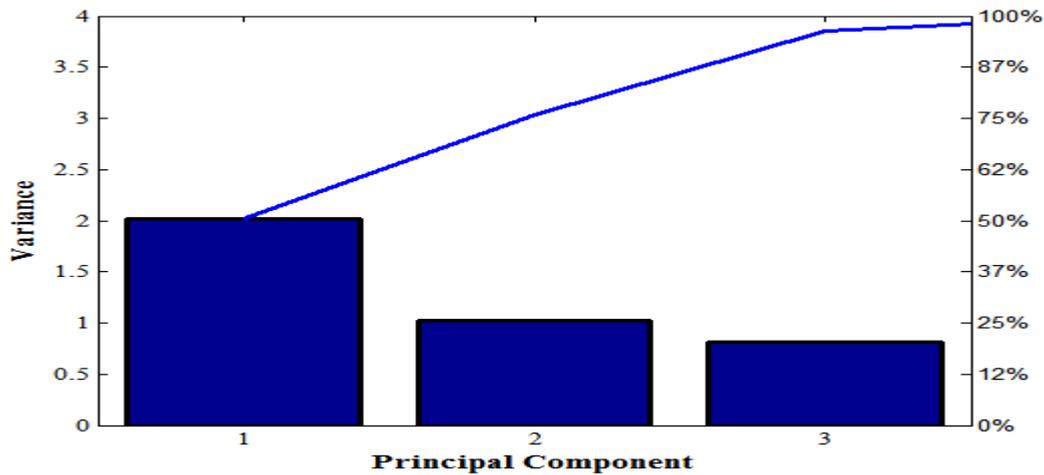


Figure 4-3. Pareto chart showing the spread of the data (variance) among each of the principal components

Pictorially, the biplot shown in Figure 4-4, below, illustrates the relative influence of each variable in two-dimensional (2-D) space.

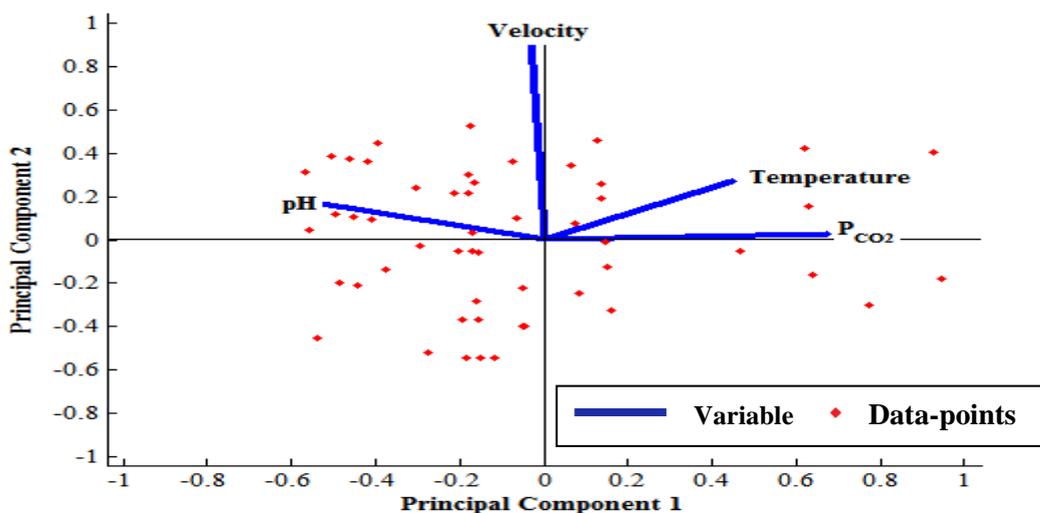


Figure 4-4. Two-dimensional biplot showing the orientation (magnitude and direction) of the variables in the principal component axes

The 2-D biplot shows that CO₂ partial pressure, temperature and velocity are three most dominant variables in terms of variation within the dataset and as such are likely to have a greater influence in terms of the prediction of CO₂ corrosion rates, due to their respective lengths and directions in the first and second component axes. The CO₂ partial pressure and temperature variables point in the positive direction of the first component (PC1) axis while the velocity and pH variables do not. The velocity is considered more significant than the pH in terms of its effect on the variance because its angular orientation to the positive direction of the PC1 axis is comparatively smaller.

The scree plot shown in Figure 4-5 is essentially a line graph plot of the eigenvalues of all the principal components joined together. The slope of the line joining PC1 to PC2 is quite steep while the following line joining PC2 to PC3 is less steep, while the slope of the line joining PC3 to PC4 increases again. This serves to indicate that three principal components are required to describe this dataset.

Figure 4-5, reinforces the fact that three principal components are responsible for the greatest variation in the dataset, and as indicated by the biplot in Figure 4-4, the principal variables are Temperature, P_{CO_2} and Velocity. This deduction is used as a basis in the determination of a suitable multiple linear regression model shown in the following section.

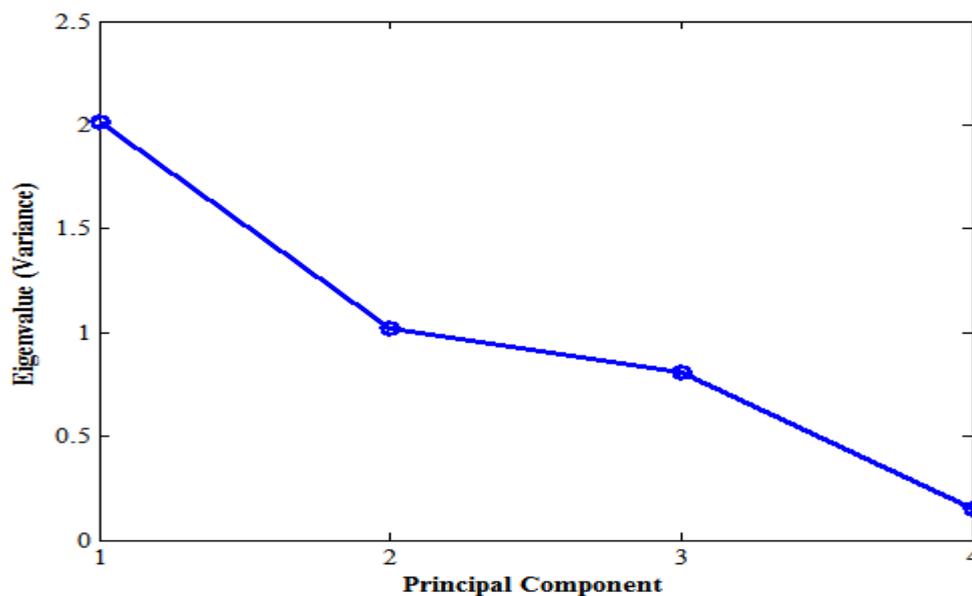


Figure 4-5. Scree plot showing decreasing eigenvalues (variances) for all the principal components

4.2.3 Multiple Linear Regression

Multiple linear regression is a term used to describe the modelling of a relationship between two or more explanatory variables and a single response variable by fitting a linear equation to observed or experimental data (Chatterjee and Hadi, 1986).

The general form of a multiple linear regression model is represented as follows:

$$y_i = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n \text{ for } i = 1, 2, \dots, n \quad (4.1)$$

Where:

$\beta_0, \beta_1, \beta_2, \dots, \beta_n$ are constants

x_1, x_2, \dots, x_n are explanatory variables (Temperature, P_{CO_2} , Flow Velocity, pH)

y_i is the response variable (CO_2 corrosion rate)

Given that the principal variables of CO_2 corrosion rate for this sample data as determined by principal component analysis are temperature and P_{CO_2} , the multiple linear regression model shown in Figure 4-6 comprises these variables. The following table shows the estimated coefficients for the simple multiple linear regression model.

Table 4-4. Regression model terms and statistics

Regression model term	Estimate	t-stat	p-value
Intercept	-4.7680	-0.9945	0.3243
T	0.2549	2.8549	0.0060
P_{CO_2}	36.7875	3.3849	0.0013
$T: P_{CO_2}$	-0.2954	-2.0851	0.0416

The R -squared and adjusted R -squared values are 0.507 and 0.480 respectively for $N=60$ (number of data-points) and 56 error degrees of freedom. The t -stat statistic and p -value from Table 4-4 indicate the level of importance of the given term. The p -values of all the regression model terms with the exception of the intercept are less than the critical p -value of 0.05, thus implying that these terms are significant. The implication for the intercept term is that there is no significant difference in the regression model should this term be included or excluded completely.

For the t -stat values in Table 4-4, the absolute t -stat values for each regression model term is considered. The absolute t -stat values for each of the regression model terms with the exception of the intercept, exceeds the critical t -statistic of 2.004 (see Statistical t -tables in Appendix A4.1), for 56 degrees of freedom and the 0.025 significance level. Therefore, the intercept is the only term that is not significant in the model – the same conclusion as that of the p -value statistic.

Based on the estimate values in Table 4-4, the regression equation is given by:

$$\text{Corr Rate} = -4.7680 + (0.2549 \times T) + (36.7875 \times P_{CO_2}) - (0.2954 \times T \times P_{CO_2}) \quad (4.2)$$

The polygonal plane through the points in the three-dimensional scatter plot in Figure 4-6, is a graphical representation of the corrosion rate regression model derived above. Alternate views for this plot are shown in Appendix A4.2

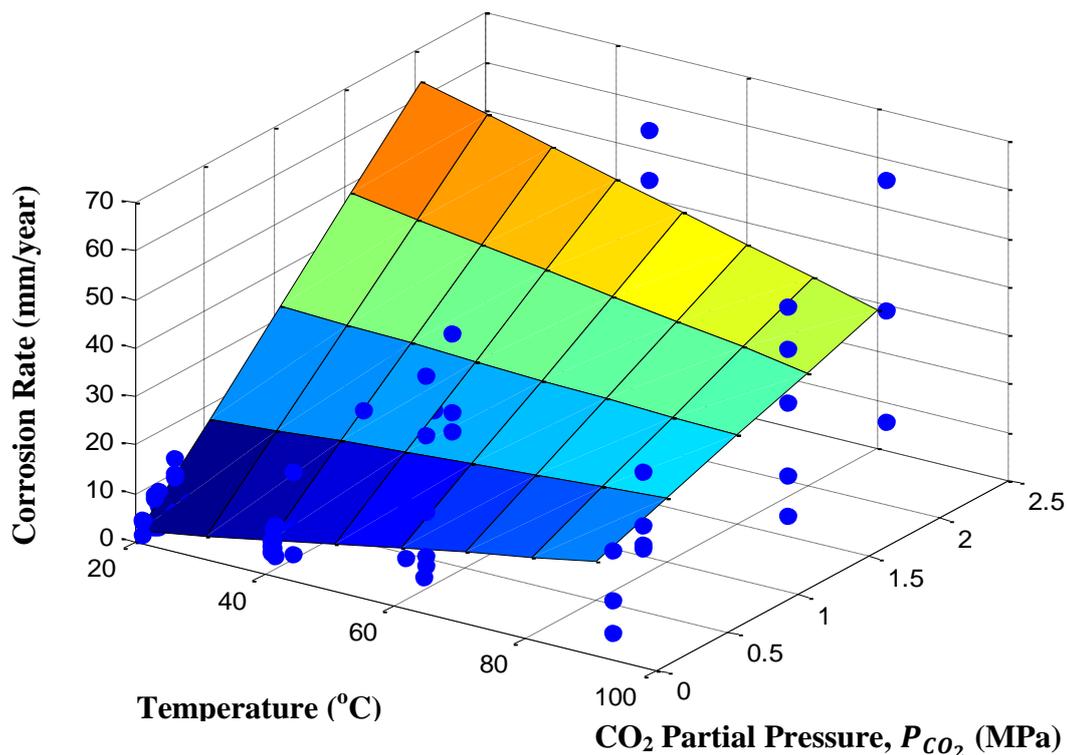


Figure 4-6. Three-dimensional plot of corrosion rate against temperature and CO₂ partial pressure (P_{CO_2}) showing regression model through the points

Another linear regression model was derived with seven terms given by the following relationship.

$$\begin{aligned} \text{Corr Rate} = & 78.20 + (0.99 \times T) - (2.00 \times P_{CO_2}) + (0.98 \times U) - (25.00 \times \text{pH}) \\ & - (7.00 \times 10^{-3} \times T^2) + (5.30 \times 10^{-3} \times T \times P_{CO_2} \times U \times \text{pH}) \quad (4.3) \end{aligned}$$

The R -squared and adjusted R -squared coefficients are 0.79 and 0.77 respectively. This model was used to establish the variable interactions shown in the following section. The plots in Figure 4-7 show the performance of the model. Predictions from the model are made using the test dataset and the results are plotted against experimental data. Errors are calculated by subtracting the model predictions from the experimental corrosion rates. The R^2 -statistic is a common criterion for goodness of fit for regression models and indicates how well the mathematical model predictions match up against the experimental data (Draper and Smith, 1998; Abyaneh, 2014). The definition of the R^2 -statistic and its derivation based on arbitrary data-points are covered in Section 5.6.

There is a good positive correlation between the model and experimental data. The data-points are also in close proximity to the 95% confidence intervals.

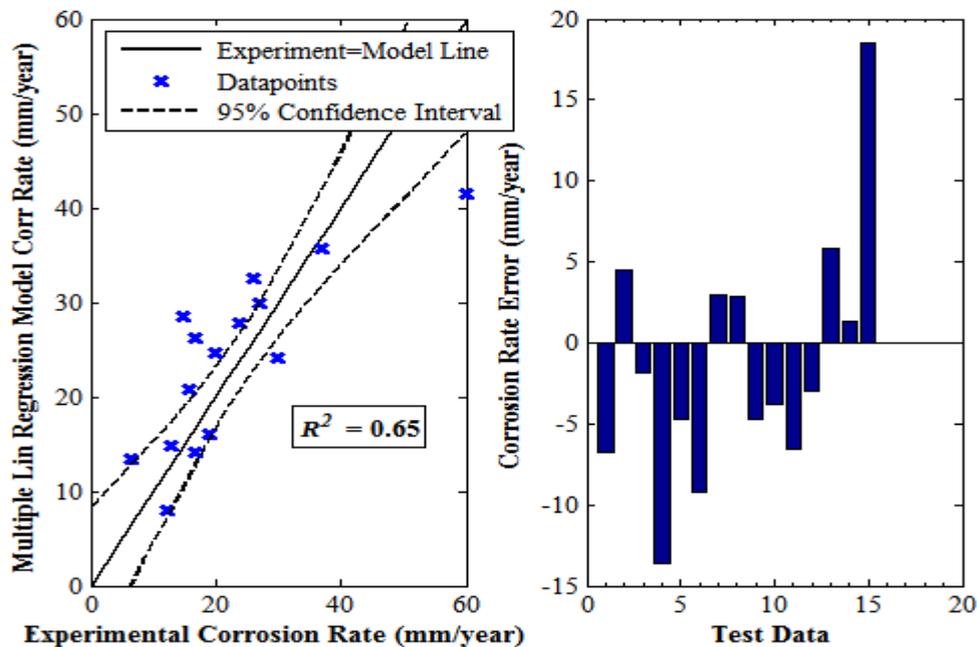


Figure 4-7. Multiple linear regression model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate

Variable Interactions

Interaction prediction plots between single input variables, temperature, P_{CO_2} , U and pH, are simulated to examine their effect on corrosion rate. Adjusted corrosion rate responses are a function of the x-axis variable, shown in each of these plots, with the other variables fixed at constant values – low, medium and high magnitudes. These plots make it possible to examine whether the effect of one variable depends on the value of the other variable, and if so, to what extent.

Figure 4-8A shows the prediction interaction between temperature and P_{CO_2} . It is observed that at $T=20^\circ\text{C}$, the corrosion rate is low, barely exceeding 5mm/year while corrosion rates are greater for $T=55^\circ\text{C}$ than for $T=90^\circ\text{C}$, at lower CO_2 partial pressures. The corrosion rate for $T=90^\circ\text{C}$ is greater than that of $T=55^\circ\text{C}$ when P_{CO_2} exceeds 1MPa. It is generally the case that the higher temperatures induce greater corrosion rates because the rates of chemical reactions increase with increasing temperatures (Tan and Chan, 2011).

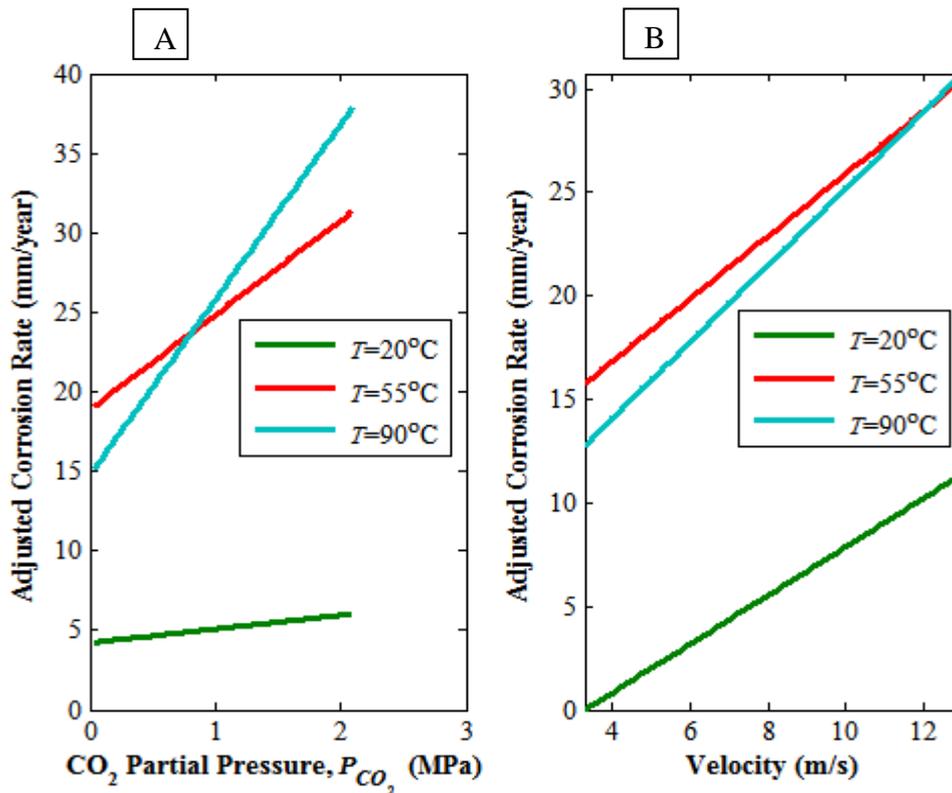


Figure 4-8. Variable interactions between:

A. P_{CO_2} and temperature B. Velocity and temperature

Figure 4-8 B shows the prediction interaction plot for temperature and velocity. The $T=20^{\circ}\text{C}$ line (shown in green) results in the least rate of corrosion. In general, corrosion rates increase from 0 to 11mm/year as the velocity increases from 0 to 13m/s. Initially, the $T=55^{\circ}\text{C}$ line (shown in red) exceeds the $T=90^{\circ}\text{C}$ (shown in blue) in terms of the magnitude of corrosion rate, i.e. $16 > 14\text{mm/year}$, however as velocity increases to a value of 12m/s, the two lines intersect and then the $T=90^{\circ}\text{C}$ line exceeds the $T=55^{\circ}\text{C}$ line at $U=13\text{m/s}$.

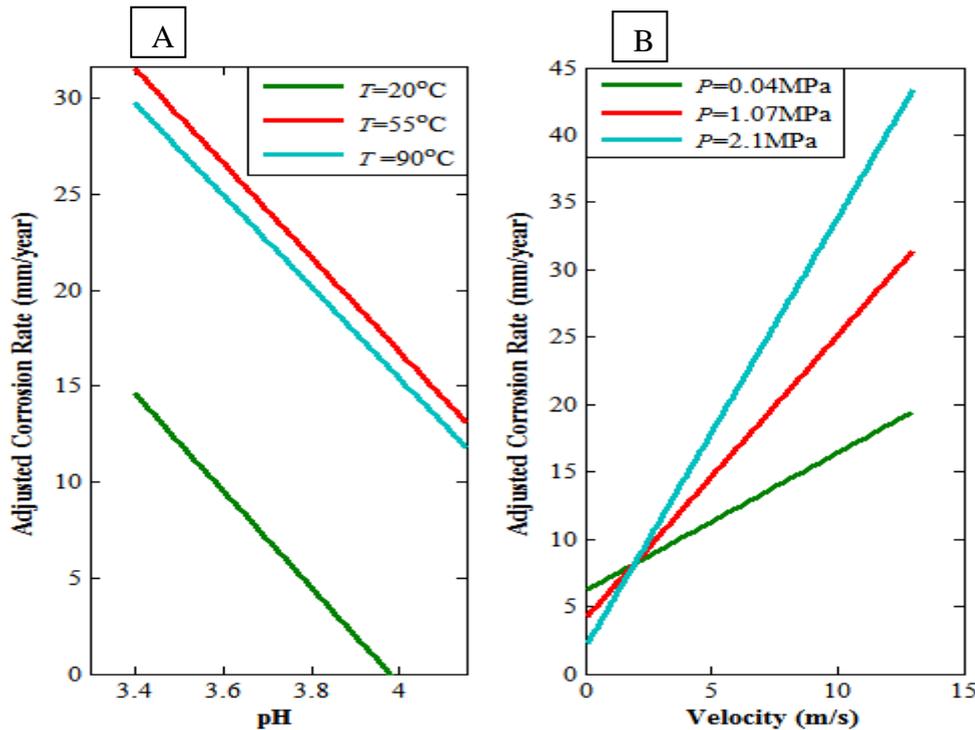


Figure 4-9. Variable interactions between:

A. pH and temperature B. Velocity and P_{CO_2}

Figure 4-9A shows the interaction plot for temperature and pH. Here, all the lines have a negative gradient. This indicates that low pH values have a greater effect on corrosion rate than higher ones as the concentration of the acidic medium is greater at low pH causing a faster rate of dissolution of the metal. The $T=20^{\circ}\text{C}$ line induces the least corrosion rate while the $T=55^{\circ}\text{C}$ induces the greatest corrosion rate – the reason being that corrosion rate versus temperature plots often show a distinct peak at mid-temperatures (Dugstad et al., 1994a).

Figure 4-9B shows the prediction interaction plot for P_{CO_2} and velocity. It is observed that higher P_{CO_2} values result in greater corrosion rates, as greater acid concentrations within the fluid medium in contact with the metal surface become more prevalent. Initially,

$P_{CO_2}=0.04\text{MPa}$ yields the greater rate of corrosion, i.e. corrosion rate at $P_{CO_2}=0.04\text{MPa}$ > corrosion rate at $P_{CO_2}=1.07\text{MPa}$ > corrosion rate at $P_{CO_2}=2.1\text{MPa}$, for $U=0\text{m/s}$. However, at $U=2\text{m/s}$, the three lines intersect and then the order of adjusted corrosion rates is reversed: corrosion rate at $P_{CO_2}=2.1\text{MPa}$ > corrosion rate at $P_{CO_2}=1.07\text{MPa}$ > corrosion rate at $P_{CO_2}=0.04\text{MPa}$.

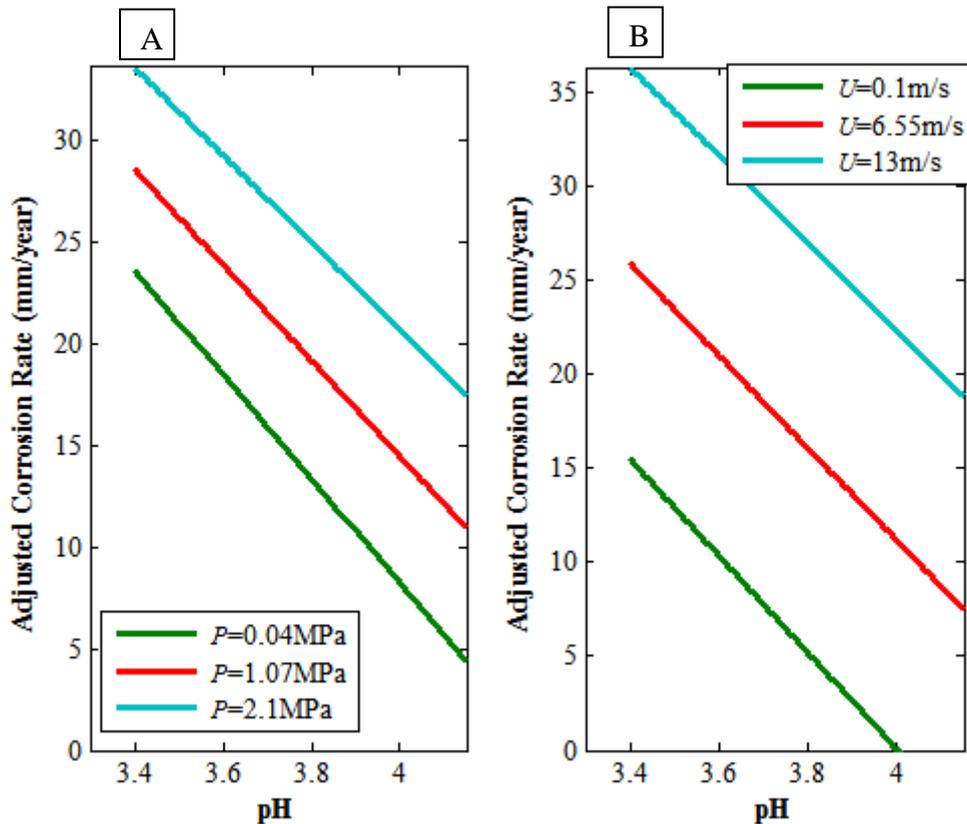


Figure 4-10. Variable interactions between:

A. pH and P_{CO_2} B. pH and velocity

Figure 4-10A shows the prediction-interaction plot for P_{CO_2} and pH. Each of the lines in the plot has negative gradients. Corrosion rates commence at approximately 32, 28 and 24mm/year for $P_{CO_2}=2.1$, 1.07 and 0.04MPa respectively and decrease steadily until they reach 17, 11 and 4mm/year.

Figure 4-10B shows the prediction-interaction plot for velocity and pH. Corrosion rates fall for each of the line plots with decreasing pH, thus negative slopes. The magnitudes of adjusted corrosion rates are in the following order: corrosion rate at $U=13\text{m/s}$ > corrosion rate

$U=6.55\text{m/s}$ corrosion rate at $U=0.1\text{m/s}$. This order is so because higher velocities have a greater tendency to sweep away any protective films on the metal surface thereby inducing greater corrosion rates. Initially, corrosion rates commence at ~36, 26 and 16mm/year at a pH of 3.4 and then these rates decrease to 19, 7.5 and 0mm/year respectively for $U=13, 6.55$ and 0.1m/s respectively, at $\text{pH}=4.15$.

4.2.4 Nonlinear Regression

Parametric nonlinear regression is carried out on the dataset and is an attempt to characterise the relationship between several continuous predictor variables (input) and a continuous response variable (output) with the use of nonlinear parameters. The model takes the following form (Chatterjee and Hadi, 1986):

$$y = f(x) + \varepsilon_r \quad (4.4)$$

Where:

y is the response variable (CO_2 corrosion rate)

f is the function that transforms predictor variables (input) to response variables (output)

x is the predictor variable or input variable (Temperature, PCO_2 , Velocity and pH)

ε_r is a set of independent random disturbances

In particular, the Hougen-Watson equation has proven to be a very useful nonlinear equation. The Hougen-Watson equation is a derived equation which expresses the rate of a chemical reaction in terms of the concentration of reacting species (Carberry, 2001; Staelens et al., 2002). This relation is as follows:

$$y_i = \frac{\beta_1 x_2 - x_3 / \beta_5}{1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3} \quad (4.5)$$

Where:

y_i = CO_2 corrosion rate (mm/year)

x_1, \dots, x_3 are the predictor variables (temperature, PCO_2 and velocity)

β_1, \dots, β_5 are predictor variable coefficients (determined by nonlinear regression)

In this study, the predictor variables replace the concentrations of reacting species. From the formulation, it is clear that the corrosion rate can be expressed in terms of three variables. The R -squared values for this equation using the study dataset with a combination of temperature, P_{CO_2} and velocity, as predictor variables is 0.7; while the combination of temperature, P_{CO_2} and pH as predictor variables yields an R -squared value of 0.45.

The magnitudes of the R -squared values provide supporting evidence to the deduction from principal component analysis, which is that temperature, P_{CO_2} and velocity are the principal variables for this dataset.

The plots in Figure 4-11 show the performance of the model. The R^2 -value is moderately high with majority of the data-points falling within the 95% confidence interval. This indicates that the model is reasonably accurate and reliable.

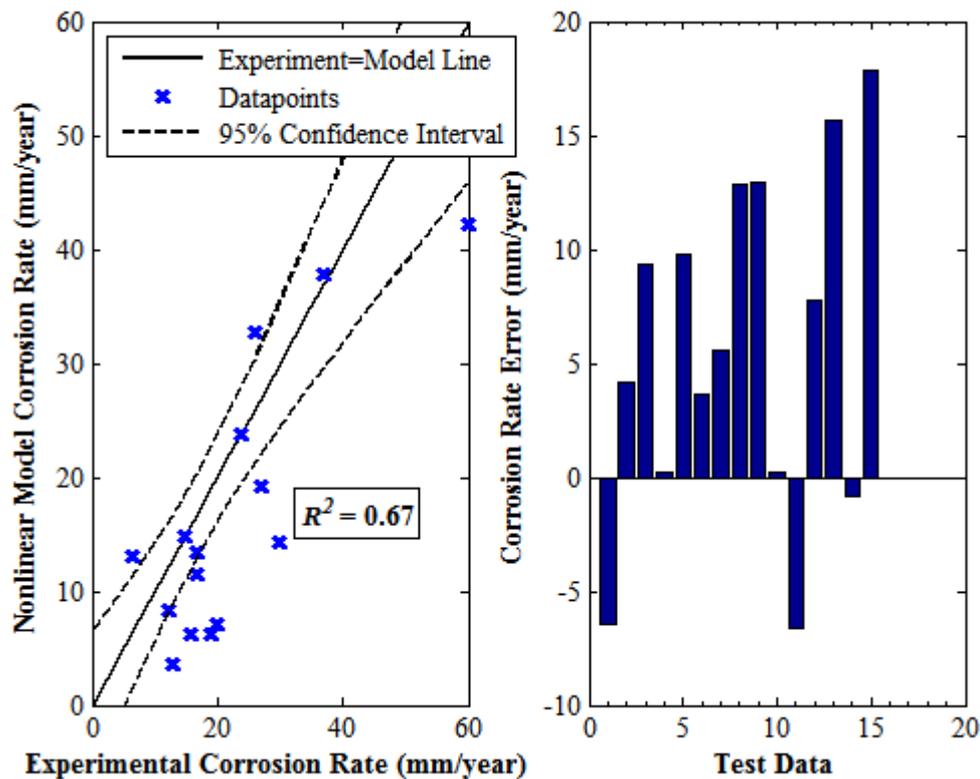


Figure 4-11. Hougen-Watson nonlinear model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – Error is the difference between experimental and model corrosion rate

The plots in Figure 4-12 show how well the model fits the original data. The leverage plot indicates that a few outliers are present (points above the dashed line). There is a possibility that these high-leverage datapoints have greater random errors associated with their experimental measurements than the rest of datapoints in this particular dataset. Nevertheless, majority of the datapoints lie well below the dashed line (Chatterjee and Hadi, 1986).

The residuals plot shows that there is an even spread of the datapoints about the zero mark (dashed line). This implies that the model is being sufficiently explained by the three predictor variables.

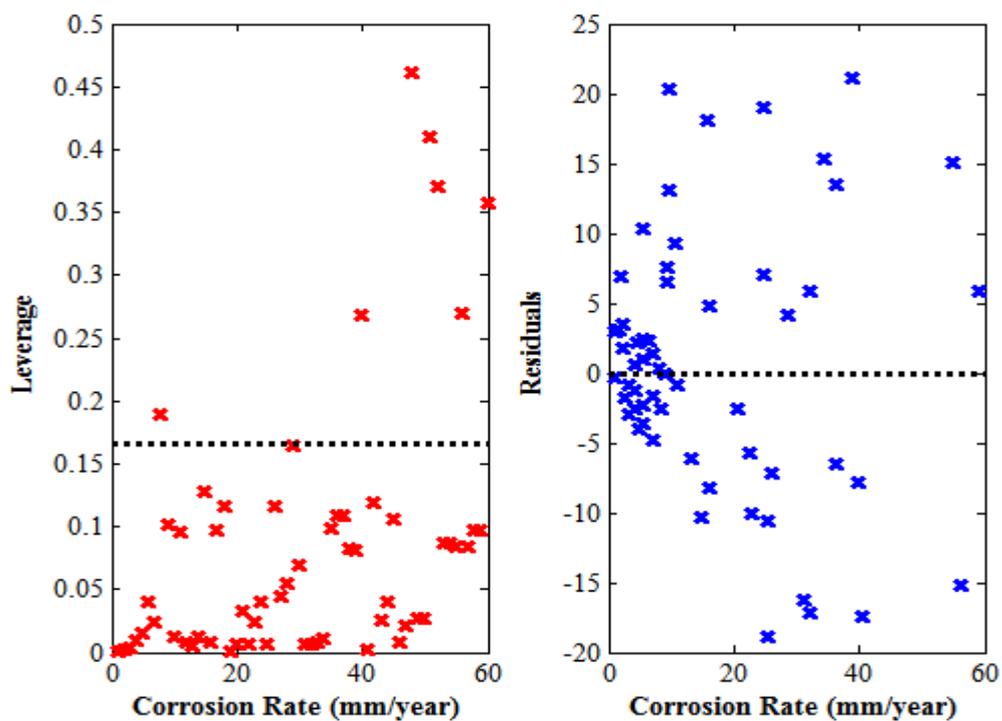


Figure 4-12. Leverage-CO₂ corrosion data plot and residuals-CO₂ corrosion data plot

4.2.5 Response Surface Plots

Response surface modelling plot slices can be used to examine a change in CO₂ corrosion rate by interactively changing the magnitude of any chosen predictor variable as shown in Figure 4-13. The advantage this model has over the Hougen-Watson model is that, it permits the use of all variables in the dataset.

The response surface plots also show trends for each variable, for instance, greater magnitudes of all the variables except pH lead to an increase in CO₂ corrosion rate (shown by

the centre straight line shown in each plot slice). CO₂ corrosion rate outputs are predicted using 95% confidence limits. The performance of the response surface model is summarised in the plots of Figure 4-14. The response surface tool produced a low value for the R^2 -statistic and this is because of a combination of the variation within the dataset as well as the presence of outliers.

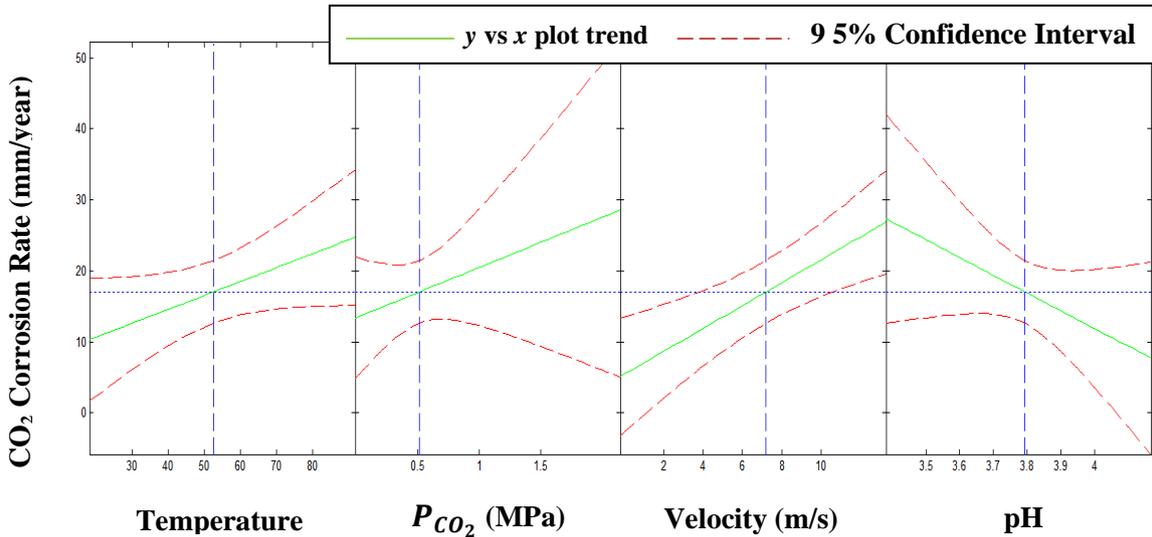


Figure 4-13. Response surface modelling plots – CO₂ corrosion rate can be investigated by changing the magnitude of each predictor variable

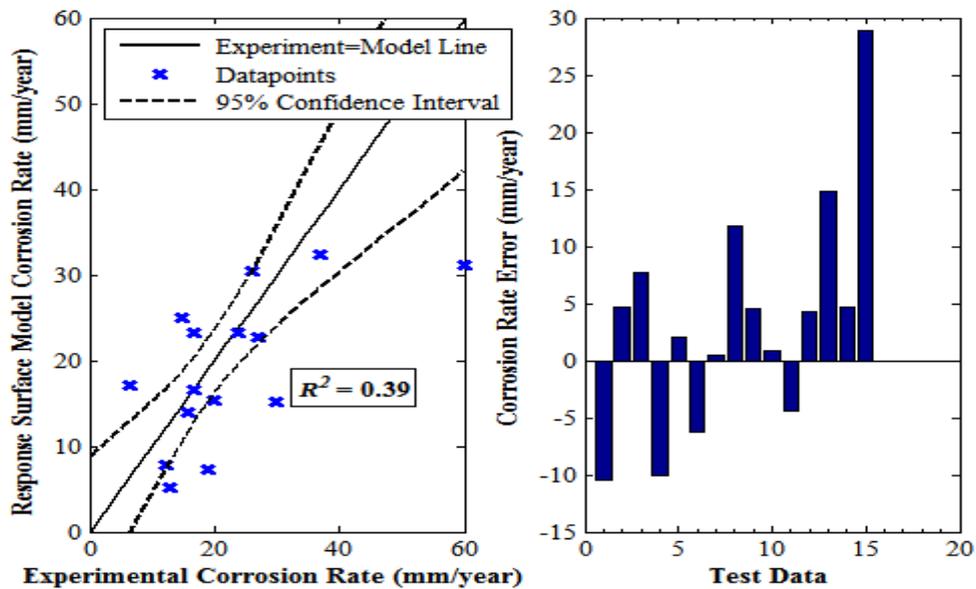


Figure 4-14. Response surface model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate

4.3 High Pressure CO₂ Corrosion Data

4.3.1 Background on Datasets and Descriptive Statistics

Data from multiple sources in open literature was used in the neural network modelling of high pressure carbon dioxide corrosion. All data sources determined corrosion rates experimentally by weight loss using autoclaves. For the Hesjevik et al (2003) study, a Hastelloy C-276 (UNS N10276) nickel-alloy was used and for the Choi and Netic (2009) study, an X65 carbon steel sample was used. For the Zhang et al (2012a) study, several samples of steel were used, including martensitic carbon steel, a pipeline X65 steel as well as three chromium-containing corrosion-resistant alloys (CRA). For modelling purposes, only carbon steel corrosion rate results were used in order to maintain consistency as corrosion rate measurements for CRA would affect the final model. For the Cui et al (2006) study, samples of API P110, N80 and J55 casing/tubing carbon pipe steels were used. Following gathering of all the data, a summary of the variable ranges is as follows: temperatures of 24-150°C, CO₂ partial pressures of 3.5-23.3 MPa, velocities of 0-4 m/s, pH values of 3.1-6 and corrosion rates of 0.9-19 mm/year. Overall there are 22 data-points. The data was then statistically analysed and summarised in Table 4-5.

Table 4-5. Descriptive statistics of the CO₂ corrosion sample data

Variable	Range	Mean	First Quartile (Q1)	Median (Q2)	Third Quartile (Q3)	Skewness
Temperature (°C)	24-150	80.25	50.00	60.00	125.00	0.46
CO ₂ Partial Pressure, P_{CO_2} (MPa)	3.5-23.3	13.17	8.07	12.10	20.30	0.32
Velocity (m/s)	0-4	2.00	0.00	2.00	4.00	0.00
pH	3.1-6	3.96	3.21	3.95	4.08	1.15
Corrosion Rate (mm/year)	0.9-19	10.78	8.32	9.75	14.50	-0.17

From the boxplot shown in Figure 4-15, temperature has the widest range of values (24-150°C) while pH has the smallest range of values (3.1-6). All of the variables excluding the corrosion rate are said to be positively skewed, implying that their means are greater than their respective medians and modes but not by substantial amounts as the skewness

magnitudes (Table 4-5), are only slightly greater than zero. The pH distribution, shown in Figure 4-16, is the only distribution with outliers (indicated by the red plus symbol). The boxplot of the pH variable indicates that the outlier is that of pH=6. This is the single greatest value for pH in the dataset. The relatively long top whisker of the pH boxplot also indicates it is highly positively skewed. The pH skewness of 1.15, from Table 4-1 confirms this.

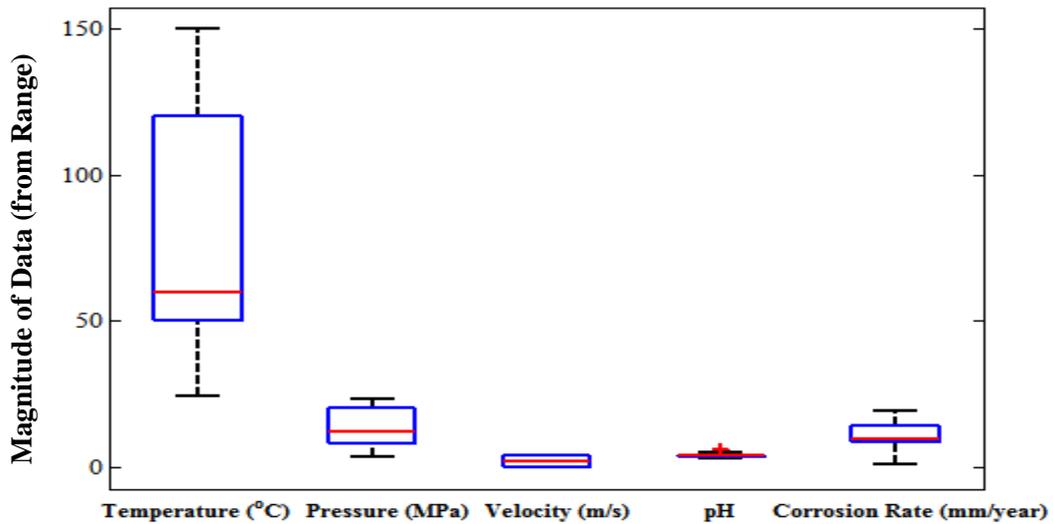


Figure 4-15. Boxplot showing the overall range (box and whiskers), inter-quartile range (white space in blue-edged box), median, skewness and outliers (red plus symbols) in each of the variables in the dataset

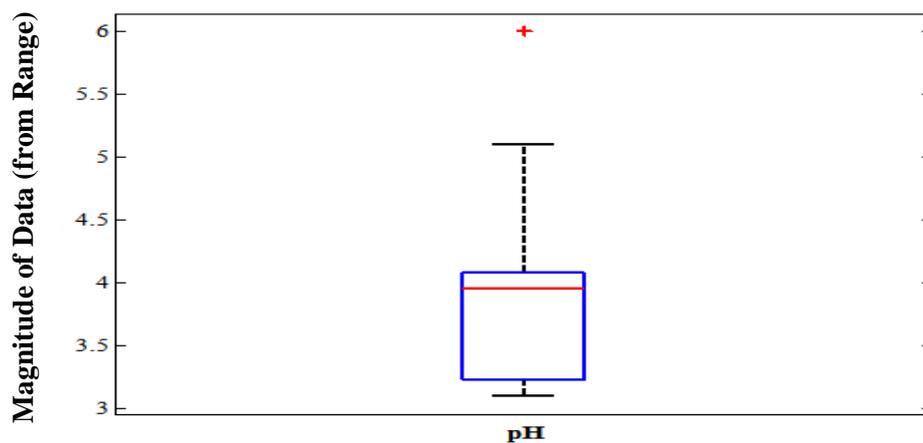


Figure 4-16. Boxplot showing the range of values, inter-quartile range, median, skewness and outliers of the pH variable

A summary of the test dataset is shown in Table 4-6. The size of this dataset is small as the actual dataset for the study also has a small size (N=16). This dataset is used for validation of the model and is not included in the development of the final model.

Table 4-6. Test data variables and their corresponding ranges

Variable	Range
Temperature (°C)	40-140
P_{CO_2} (MPa)	6.0-23.3
Fluid flowrate (m/s)	0-4
pH	3.1-5.0
Corrosion Rate (mm/year)	1.3-20

4.3.2 Principal Component Analysis

A weighted principal component analysis was carried out on the high pressure CO₂ dataset. The following Table shows the cumulative contribution of each of the principal components to the variance of the data. The cumulative percentages indicate how well the principal components explain variation within the data. The cumulative share for principal components 1 and 2 is 81.5% which is sufficient to explain variation in the dataset.

Table 4-7. Principal component analysis

Principal Component	Principal Component 1 (PC1)	Principal Component 2 (PC2)	Principal Component 3 (PC3)	Principal Component 4 (PC4)
Eigenvalue (Variance)	2.121	1.139	0.619	0.122
Proportion	0.530	0.285	0.155	0.030
Cumulative	0.530	0.815	0.970	1.000
Cumulative Percentage (%)	53.000	81.500	97.000	100.000

The Pareto chart shown in Figure 4-17 shows the relative significance of the principal components to the variance of the data.

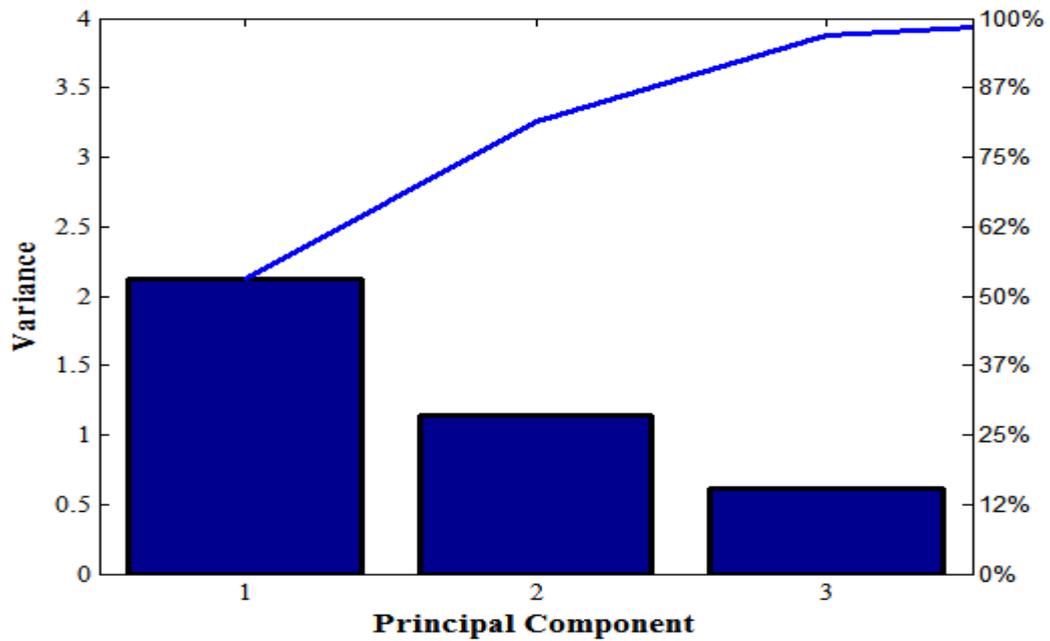


Figure 4-17. Pareto Chart showing the spread of the data (variance) among each of the principal components

The relative influence of each of the variables to the first and second principal components is depicted in the two-dimensional biplot in Figure 4-18.

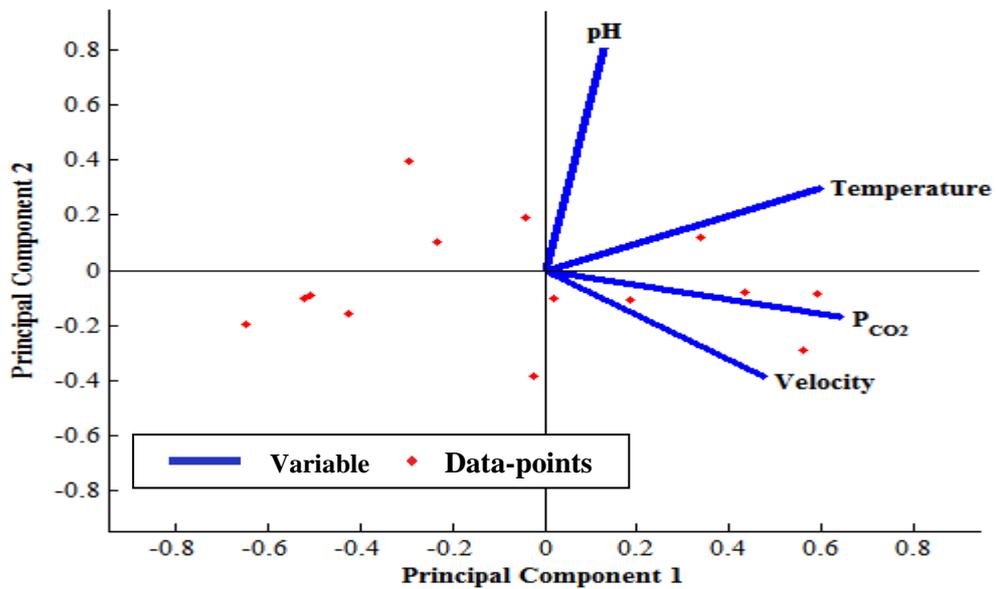


Figure 4-18. Two-dimensional Showing the orientation of the variables in the principal component axes

The 2-D biplot (Figure 4-18) shows that the CO₂ partial pressure (P_{CO_2}) variable is the most dominant variable in terms of accounting for the variation within the dataset followed by the temperature, velocity and pH respectively. This is because the direction of these three variables shows they are within close proximity to the Principal component 1 (PC1) axis. The length of each variable indicates their relative contribution to each principal component.

The scree plot shown in Figure 4-19 is a line graph of the eigenvalues from Table 4-7. The slopes change from PC1 to PC2, PC3 and PC4. However, the most significant change in slope occurs between PC1 and PC2, afterwards, the slope from PC2 to PC4 is uniform. This indicates that PC1 and PC2 are sufficient to describe the variation within the dataset.

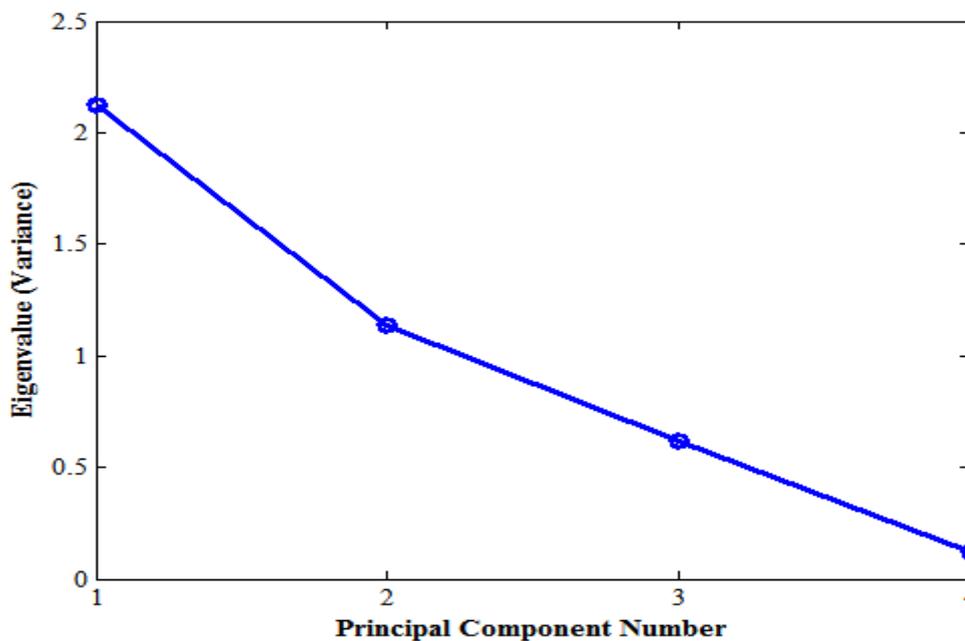


Figure 4-19. Scree plot showing the decreasing magnitudes of eigenvalues for all the principal components

4.3.3 Multiple Linear Regression

Multiple linear regression technique was used to model CO₂ corrosion rates in terms of the predictor variables: temperature, P_{CO_2} , flow velocity and pH. The regression equation (Equation. 4.1) was applied to the high pressure CO₂ data for two variables, T and P_{CO_2} . This equation is graphed in Figure 4-20 as a 3-dimensional plot. However this equation has an R^2 -coefficient of 0.28, which is very small. Therefore, it is necessary to include more predictors in the model.

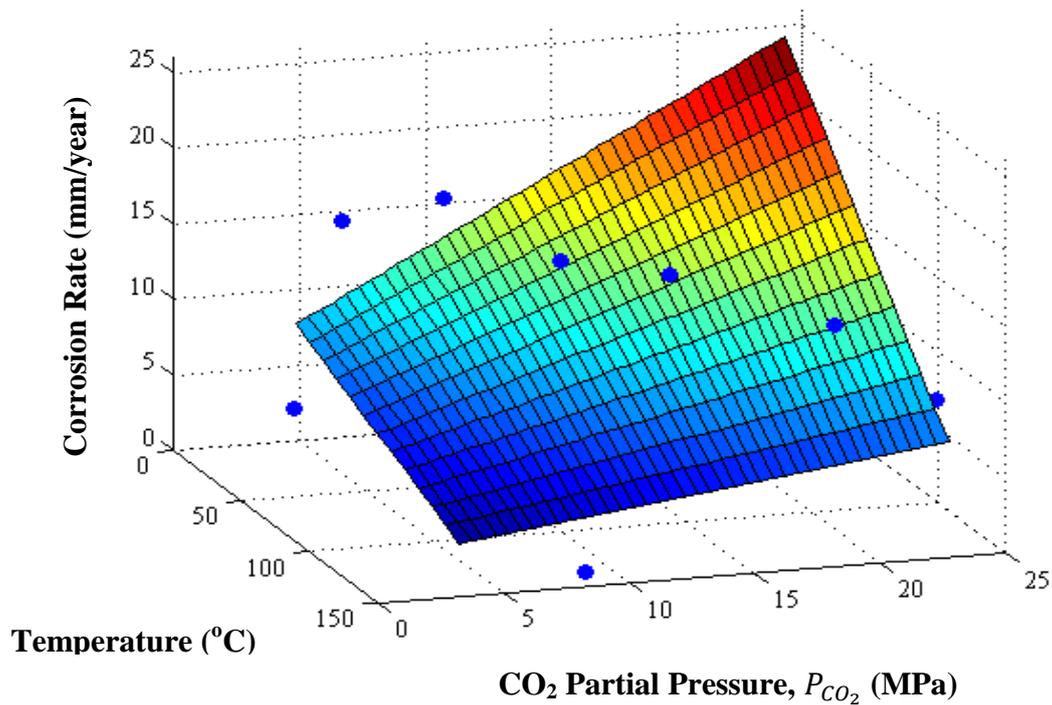


Figure 4-20. Three-dimensional plot of corrosion rate against temperature and CO₂ partial pressure (P_{CO_2}) showing regression model through the points

Table 4-8 shows the model terms for a suitably derived regression equation.

Table 4-8. Regression model terms and statistics

Regression model term	Estimate	<i>t</i> -stat	<i>p</i> -value
Intercept	-1.4267	-0.2893	0.7789
<i>T</i>	0.8445	5.6704	0.0003
P_{CO_2}	-0.5504	-2.6466	0.0266
<i>U</i>	-0.2325	-0.4330	0.6752
pH	-3.2793	-3.8158	0.0041
T^2	-0.0044	-5.9486	0.0002
$T \times P_{CO_2} \times U \times \text{pH}$	1.545×10^{-5}	0.1994	0.8464

For N=16 observations (data-points) and 9 degrees of freedom at the 95% confidence limit, the *p*-values of all the regression terms with the exception of the intercept and velocity terms, is less than the critical *p*=0.05 value. This indicates all these terms are significant to the model except the intercept, velocity and $T \times P_{CO_2} \times U \times \text{pH}$ terms. For the *t*-stat values, the absolute,

non-negative figure is considered. All t -stat values for the terms are greater than 2.262 with the exception of the intercept, velocity and the $T \times P_{CO_2} \times U \times pH$ terms. The value: 2.262 in this case is the critical t -statistic for 9 degrees of freedom at the 0.025 significance level (see Statistical t -tables in Appendix A4.1). Therefore the t -stat analysis arrives at the same conclusion as the p -value. Hence the intercept, velocity and $T \times P_{CO_2} \times U \times pH$ terms can be neglected altogether. This equation is given by:

$$\begin{aligned} \text{Corr Rate} = & -1.43 + (0.84 \times T) - (0.55 \times P_{CO_2}) - (2.23 \times U) - (3.28 \times pH) \\ & - (4.40 \times 10^{-3} \times T^2) + (1.55 \times 10^{-5} \times T \times P_{CO_2} \times U \times pH) \quad (4.6) \end{aligned}$$

The performance of the model is summarised in Figure 4-21. An R^2 -value of 0.70 was obtained for the high pressure CO_2 corrosion data. Also, the data-points lie in close proximity to the 95% confidence bounds.

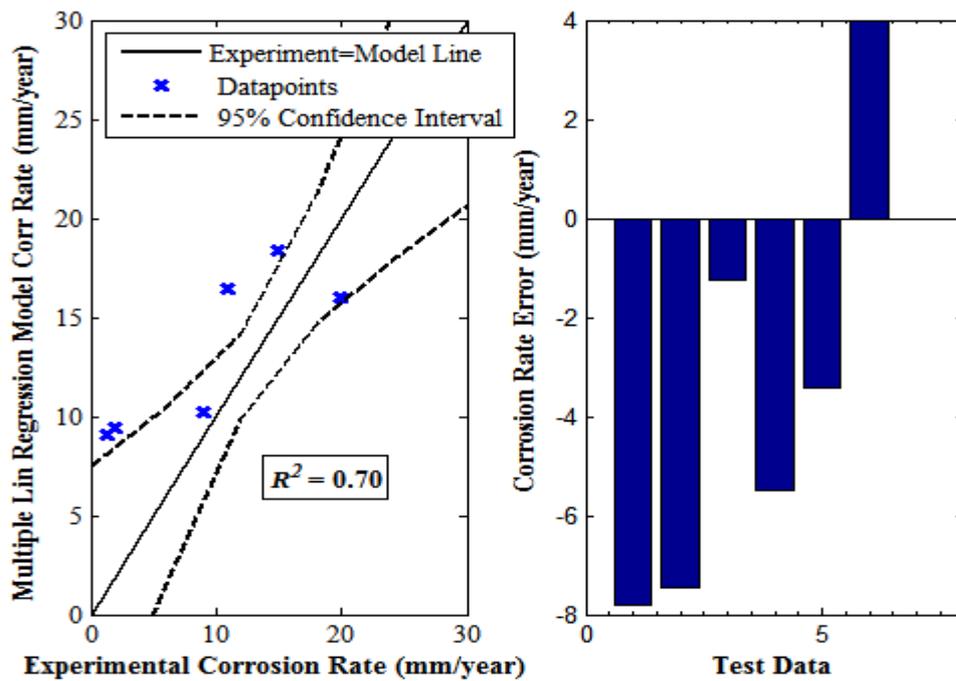


Figure 4-21. Multiple linear regression model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data

Variable Interactions

The interaction between input variables is discussed by the use of variable interaction plots. By varying the x-axis variable and keeping a second variable constant at three magnitudes: low, medium and high, the changes in adjusted corrosion rate can be investigated.

In Figures 4-22A and 4-22B, the interactions between temperature and CO₂ partial pressure and velocity are shown as straight lines with negative gradients. These generally indicate that an increase in the x-variable leads to a decrease in the adjusted corrosion rate. For the temperature interaction with CO₂ partial pressure, Figure 4-22A, the higher the CO₂ partial pressures, the smaller the magnitude of the corrosion rate. This might be due to an increased effect of the thicker scale formation at higher pressures forming a protective barrier on the metal surface against corrosion (Zhang et al., 2012b). The changes in temperature from the low to medium and high values indicate that corrosion rates peak at the medium temperature ($T=87^{\circ}\text{C}$), starting off and ending with small corrosion rates for low and high temperatures respectively. This is true for high pressure CO₂ corrosion (Zhang et al., 2012b).

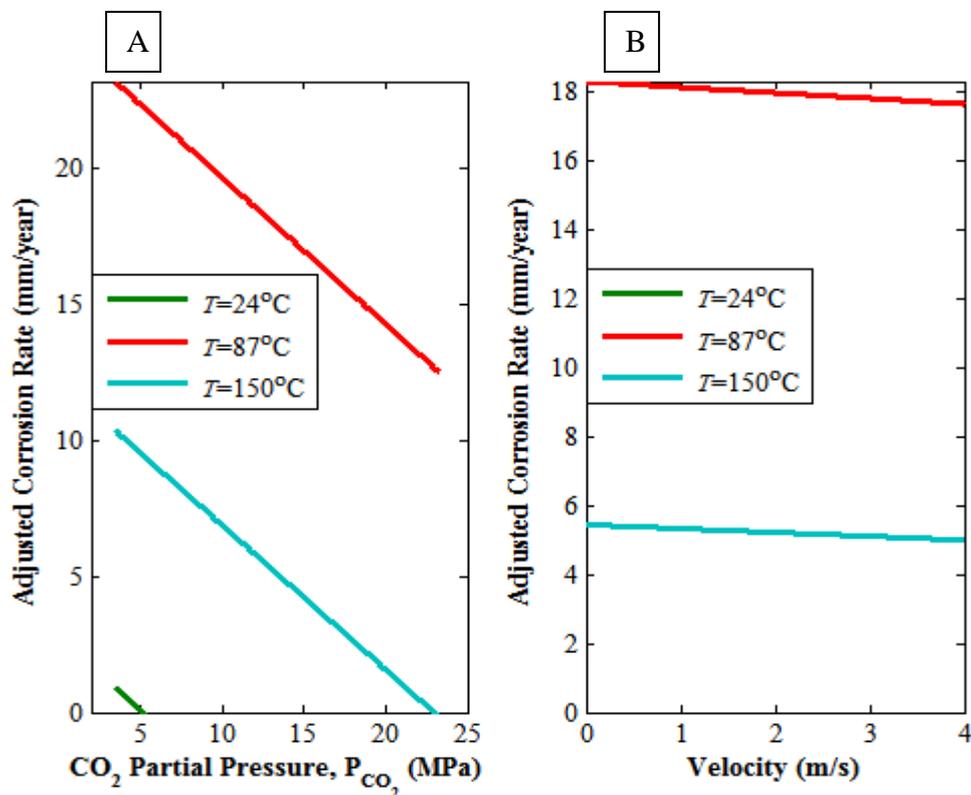


Figure 4-22. Variable interactions between:

A. P_{CO_2} and temperature B. Velocity and temperature

For Figure 4-22B, higher flow velocities have a decreased effect on corrosion rates but only very slightly and this may be because for high CO₂ pressures, thicker scales are formed and flow velocities of 0 to 4m/s in magnitude are insufficient to induce the washing away of these

scales from the metal surface. For Figure 4-23A, the interaction between temperature and pH results in straight lines with negative gradients.

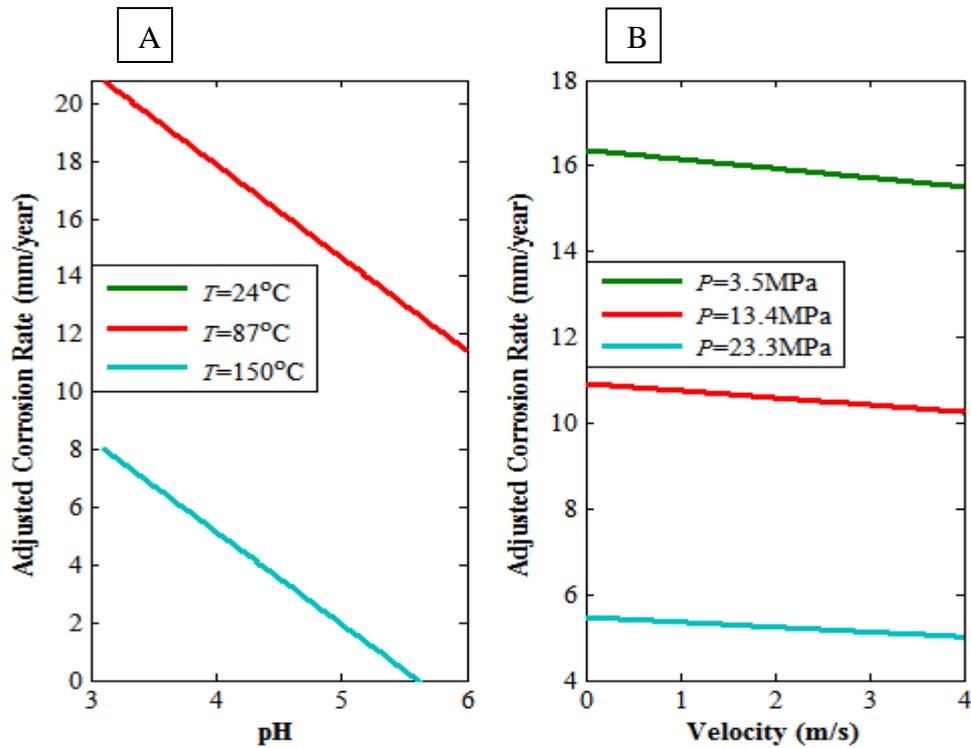


Figure 4-23. Variable interactions between:

A. pH and temperature B. Velocity and P_{CO_2}

An increase in pH leads to a drop in the corrosion rate, which is because the higher the pH, the lesser its influence on CO_2 corrosion as the acidity of the aqueous medium decreases. The $T=24^{\circ}\text{C}$ line is not visible on this plot because its effect on corrosion rate is very small. Again, the mid-temperature value of 87°C is responsible for the greatest corrosion rates. For Figure 4- 23B, increasing flow velocities leads to an overall slight decrease in the adjusted corrosion rates and this may be attributed to the tendency for thicker scale formation for high pressure CO_2 conditions. It is also observed that as the CO_2 partial pressures increase from 3.5-23.3MPa, the adjusted corrosion rates fall. It is worth noting that at high CO_2 pressure conditions, the formation of dense, more compact and thicker iron carbonate scale is favoured (Zhang et al., 2012b). Invariably, the greater the CO_2 partial pressures, the thicker the scales formed thus exerting an inhibitory effect on corrosion of the metal beneath.

For both Figures 4-24A and 4-24B, the adjusted corrosion rates decrease with increasing pH. The negative gradients of the lines are expected due to decreased effect that the aqueous medium has on corrosivity at high pH conditions. For Figure 4-24A, increasing CO₂ partial pressure has an inhibitory effect on the corrosion rate due to the formation of thicker scales at P=13.4 and P=23.3MPa. For Figure 4-24B, greater corrosion rates are observed at U=0m/s than at U=4m/s.

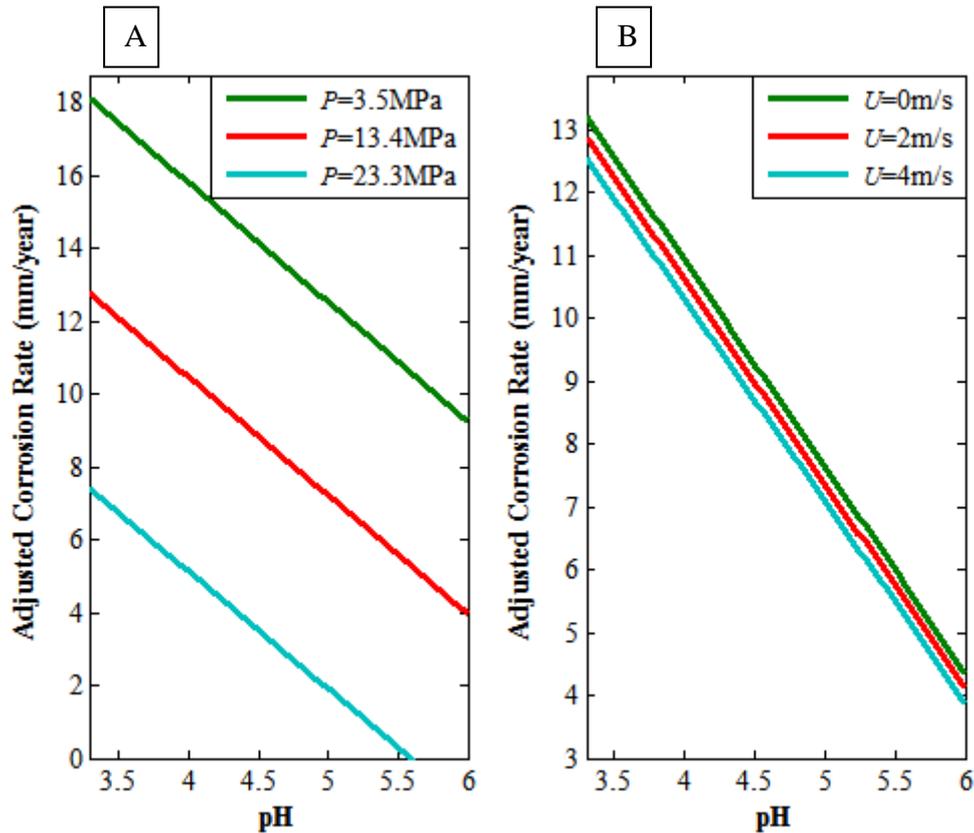


Figure 4-24. Variable interactions between:

A. pH and P_{CO_2} B. pH and velocity

In general, the greater the flow velocity, the greater the corresponding corrosion rate due to the washing away of protective layers formed by scale on the metal surface. However, when low fluid velocities are causing high corrosion rates, there is the likelihood that some degree of localised corrosion is taking place as opposed to the more common uniform corrosion, which is more characteristic of high fluid flow velocities.

4.3.4 Nonlinear Regression

Parametric nonlinear regression means the expression of a response variable in terms of a combination of several predictor variables. The Hougen-Watson equation was used as the nonlinear equation for modelling high pressure CO₂ corrosion rates. The equation has three predictor variables and based on the results of principal component analysis, the temperature, P_{CO_2} and flow velocity are the variables to be considered for modelling.

The plot in Figure 4-25 shows the performance of the model. In general, this model is a poor fit for the data. The R^2 -value was very low though nonlinear modelling is tricky especially when there is a shortage of data-points in the given high pressure CO₂ corrosion dataset.

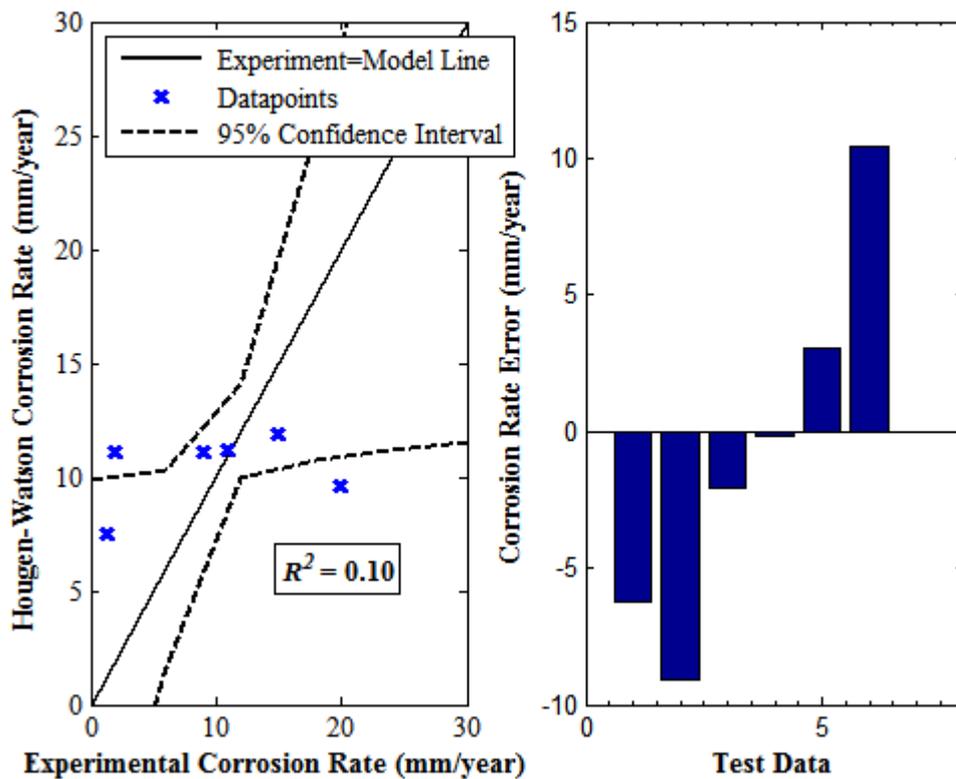


Figure 4-25. Hougen-Watson nonlinear model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data

The leverage plot in Figure 4-26 indicates that there only two notable outliers, which is in partial agreement with the descriptive statistics of the dataset that suggests there are no outliers. The residulas plot of Figure 4-26, indicates that the there is an even spread of the data about the zero dotted line, implying that the model adequately represents the data even

though its performance with the test set is poor as seen by the low R^2 -coefficient as well as the overall distribution of the data-points shown in Figure 4-25.

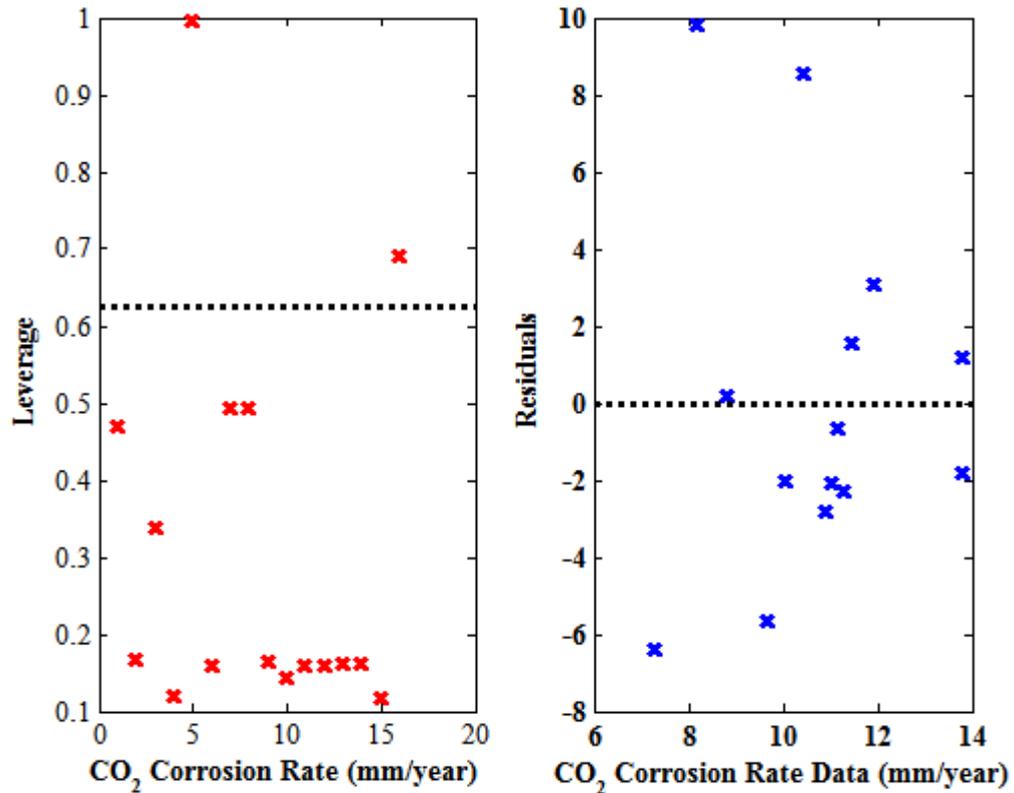


Figure 4-26. Leverage-CO₂ corrosion data plot and residuals-CO₂ corrosion data plot

4.3.5 Response Surface Plots

Response surface modelling shows the visual trends of the variables as they affect the response variable in an interactive setting. For this tool, Figure 4-27 shows the panel for each of the variables and how they affect the CO₂ corrosion rate.

The performance of the model is shown in Figure 4-28. It is seen that while this model permitted the use of all variables, the R^2 -value was extremely low. This model struggled to fit the data due to an insufficient size of the dataset. For instance, modelling involved the use of numerical techniques, and in this case a definitive solution was not reached because convergence could not be attained.

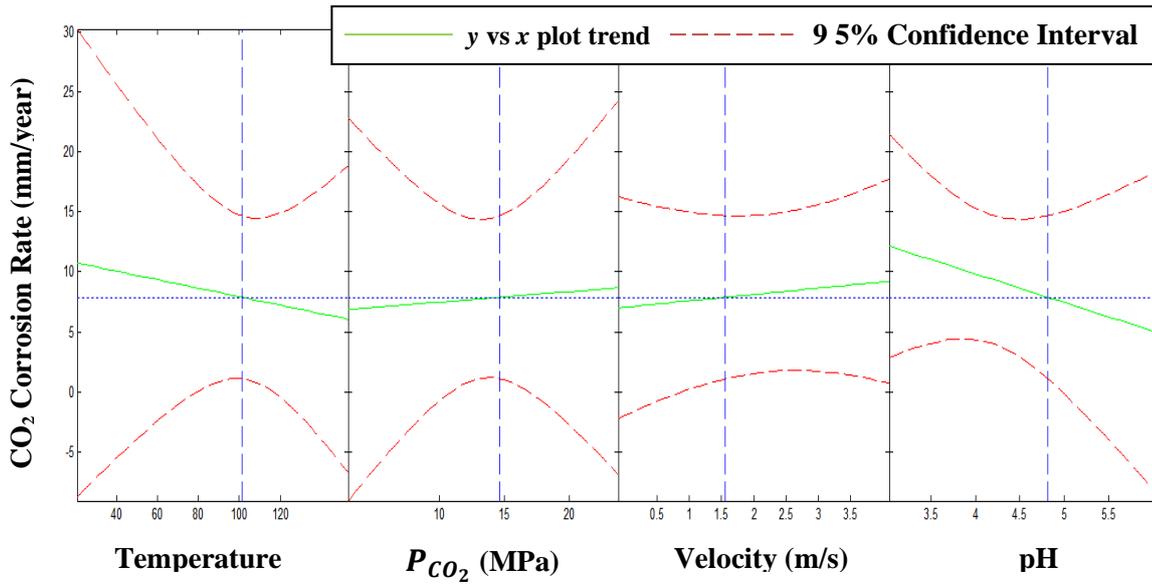


Figure 4-27. Response surface modelling plots – CO₂ corrosion rate can be investigated by changing the magnitude of each predictor variable

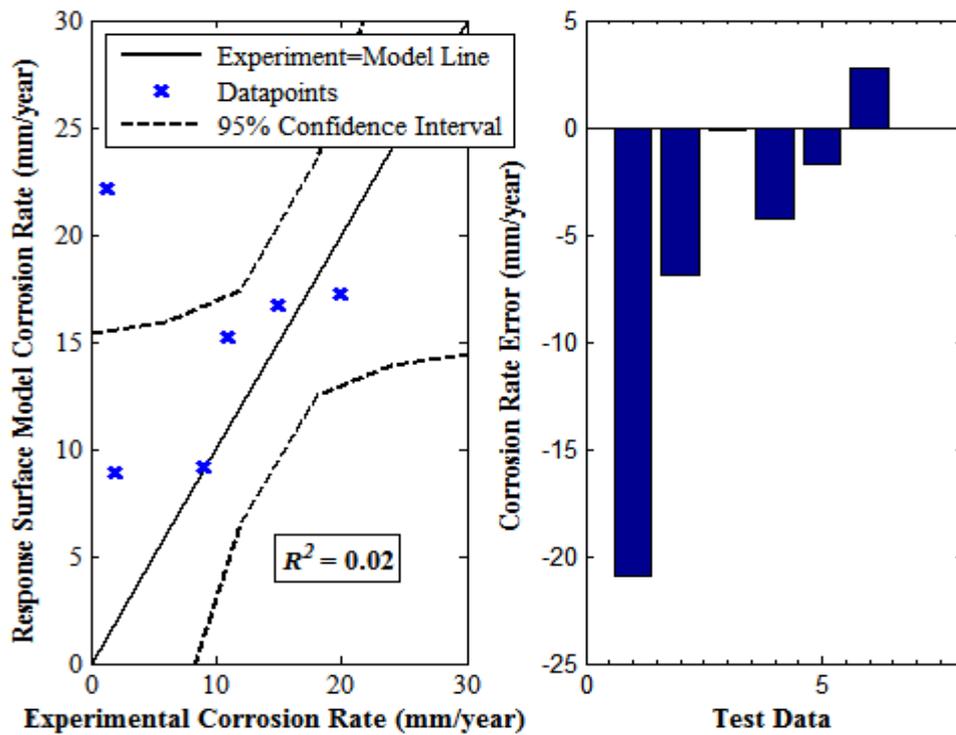


Figure 4-28. Response surface model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data

4.4 Conclusions

Descriptive statistics of the low pressure CO₂ dataset revealed that the CO₂ partial pressure and corrosion rate variables contain outliers and these variables also exhibit positive skewness, implying their means are greater than their medians. Additionally, the corrosion rate variable outliers are in the 60-70mm/year range. These outliers are data-points which lie outside of the overall pattern of the distribution. For the high pressure CO₂ dataset, the pH was the only variable distribution containing outliers. The presence of the outliers also had an effect on its skewness. Again, the pH was the only variable exhibiting notable positive skewness, with its mean only slightly greater than its median.

The results from principal component analysis on the low pressure CO₂ dataset revealed that two components were responsible for 76% of the variation within the dataset. Subsequent biplot analysis revealed that the respective variables in terms of contribution to the variation within the dataset are CO₂ partial pressure, temperature, flow velocity and pH, in that order. For the high pressure CO₂ dataset, two components accounted for 81.5% of the variation within the dataset implying that a biplot analysis in two components is sufficient to find the principal variables. Analysis from the 2-D biplot, revealed that in terms of contribution to the variance, the variables are in exactly the same order as that of the low pressure CO₂ case. The major difference between these datasets is that the pH has a more significant influence on the variance of the high pressure CO₂ dataset.

The multiple linear regression, nonlinear regression and response surface techniques were developed based on the bulk of the low pressure corrosion dataset and the accuracy of these models was carried out by assessing their prediction performances using a smaller, identical and randomly selected test dataset. This was repeated for the high pressure CO₂ corrosion dataset.

The statistical models generally have good correlations with experimental results for low pressure CO₂ corrosion. The Hougen-Watson nonlinear model has the highest R^2 -value of 0.67; test data-points have close proximity to the 95% confidence intervals. The multiple linear regression model while also achieving a similarly high R^2 -value as the nonlinear regression model, it involved the use of square-terms for the temperature variable and interaction terms comprising all variables. However, the main limitation of the Hougen-Watson nonlinear regression model is its inability to permit the use of all predictor variables.

The response surface model performed relatively poorly in comparison to the other statistical models in terms of the magnitude of the R^2 -squared coefficient. In addition, unlike the Hougen-Watson nonlinear model, it permits the use of all four predictor variables. It is also possible to interactively adjust any given predictor variable via graphical plot panes and observe the subsequent change in CO₂ corrosion rate instantaneously. Table 4-9 is a summary of the statistical model performances.

Table 4-9 Low pressure CO₂ corrosion model performance

Low Pressure CO₂ dataset	R^2-value
Multiple linear Regression	0.65
Nonlinear Regression	0.67

For high pressure CO₂ corrosion, the multiple linear regression model had a moderately high R^2 -value as seen in Table 4-10, with data-points lying in close proximity to the 95% confidence limits. There is however a slight tendency for over-prediction, particularly for low to medium magnitudes of corrosion rates. This may be due to a greater degree of variation of these points causing the model to struggle to adequately fit the data. Diagnostic plots such as the leverage and residuals revealed that there are very few outliers and that the nonlinear model satisfactorily represented the data however the model performance stated in Table 4-10 suggests otherwise. The response surface model performed very poorly in terms of the accuracy of prediction.

In general, the statistical models struggled considerably to fit the data in terms of accuracy of prediction. This can be attributed to the small dataset for high pressure CO₂ corrosion. It is worth noting that for both nonlinear regression and surface response surface models, numerical techniques are applied and for both cases definitive solutions were not reached due to a lack of convergence during numerical computation.

Table 4-10 High pressure CO₂ corrosion model performance

High Pressure CO₂ dataset	R^2-value
Multiple linear Regression	0.70
Nonlinear Regression	0.10

Chapter 5. Neural Network Models

5.1 Introduction – Historical Background

In 1943, the first artificial neuron was designed by Warren McCulloch and Walter Pitts (Haykin, 1999). These artificial neurons were capable of simple computations through the application of logical functions. The main feature of these neurons is that the weighted sum of input signals is compared to a threshold in order to determine the output of the neuron. Hence, when the sum is greater than or equal to the threshold, the output signal is 1 and when the sum is less than the threshold, the output signal is zero (Veelenturf, 1995; Ukil, 2010). This is a binary neuron since it exhibits ‘zero or one’ behaviour.

Thus,

$$\text{if } \sum w_j x_j > \text{threshold} \therefore y_j = 1 \text{ (5.1A) and if } \sum w_j x_j < \text{threshold} \therefore y_j = 0 \text{ (5.1B)}$$

Where:

w_j = weight

x_j = input signal

y_j = output signal

Figure 5-1 is an illustration of such a neuron yielding an output signal, y_i , from the various input signals, x_i , and their corresponding weights, w_i .

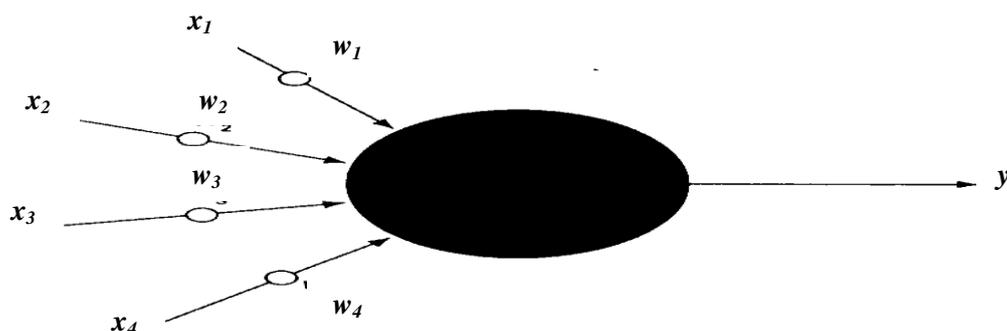


Figure 5-1. An artificial model of a neuron showing the transformation of input signals, x_i , to an output signal, y_i (Veelenturf, 1995)

However, since no training was available, all the neural network (NN) parameters had to be designed but by the late 1950's, the idea of the perceptron was conceived by Rosenblatt and others (Ukil, 2010). Perceptrons are a class of NNs that were primarily designed for pattern recognition (Veelenturf, 1995). The key difference between Rosenblatt's neurons and those

of McCulloch is the introduction of the learning rule to the former. While these perceptrons were very useful, their limitations are well-documented, one of which is their ability to solve only a limited class of problems (Haykin, 1999). Another noteworthy contribution during this period was made by von Neumann. Von Neumann made it known that there was a tendency for NNs to have redundant neuronal connections and these yielded unreliable contributions, which in turn affected overall performance (Haykin, 1999).

In the 1980's however, two key developments were made to the advancement of NNs. The first was the use of statistical mechanics to explain the operation of a recurrent network by John Hopfield, which was capable of being used as an associative memory. A recurrent network is an NN configuration with at least one feedback loop. The second key development was the formulation and use of the backpropagation algorithm for training multilayer perceptron networks, proposed by David Rumelhart and James McClelland. At present, backpropagation remains the most popular learning algorithm for the training of multilayer perceptrons (Ukil, 2010).

In chapter 3, the popular and well-established De Waard Milliam, Norsok and NPO corrosion models were discussed. In chapter 4, a derived statistical model was obtained as a means to address some of the shortcomings of the aforementioned established models. In this chapter, neural network modelling of collated low and high pressure CO₂ corrosion datasets will be carried out by dividing these sets into training and test sets.

The training set comprising variables T , P_{CO_2} , U , pH and Corr Rate, was used to develop the neural network solely and is summarised as descriptive statistics in Tables 4-1 and 4-5 (Chapter 4, Sections 4.2.1 and 4.3.1), for low and high pressure CO₂ respectively. Figures 4-1 and 4-15 are corresponding boxplots for the summary statistics in Tables 4-1 and 4-5 respectively. The test set is summarised in Tables 4-2 and 4-6 for low and high pressure CO₂ corrosion respectively. The test set is strictly used for the purpose of prediction as unseen data and for the computation of the R^2 -coefficient.

5.2 Definition of Neural Networks

A NN is a collection of information-processing units that is trained by machine learning. NNs use machine-learning algorithms to carry out non-parametric nonlinear regression of modelling data (Beale et al., 2014). By non-parametric nonlinear regression, is meant a technique in which input data is directly mapped to output data without the use of an assumed, pre-determined model equation. The stages of machine learning are shown in Figure 5-2.

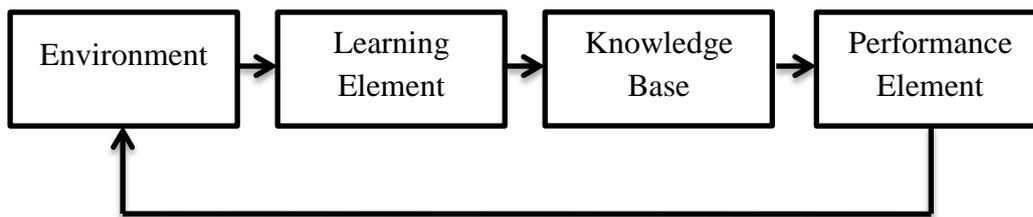


Figure 5-2. Stages in Machine Learning (Haykin, 1999)

The stages in machine learning are explained with respect to artificial intelligence (AI). The goal of artificial intelligence is the development of algorithms that require machines to perform cognitive tasks and solve problems in a manner similar to human reasoning (Haykin, 1999).

The first stage of machine learning for an AI system is to set up an environment that supplies the given input data to the learning element. The learning element then uses the data to make improvements in its existing knowledge base. The performance element makes use of the knowledge base in order for the AI system to perform its task. Input data from the environment often contains flaws; however the AI system's knowledge base is unaware initially and assesses itself by receiving feedback from the performance element. The feedback loop mechanism allows the AI system to evaluate its hypotheses and to revise them if necessary (Haykin, 1999). The error-minimisation concept in NNs applies the same logic, in which the goal is to reduce the error between a specified target and an output in a continuous loop as shown in Figure 5-3 (Demuth et al., 2009).

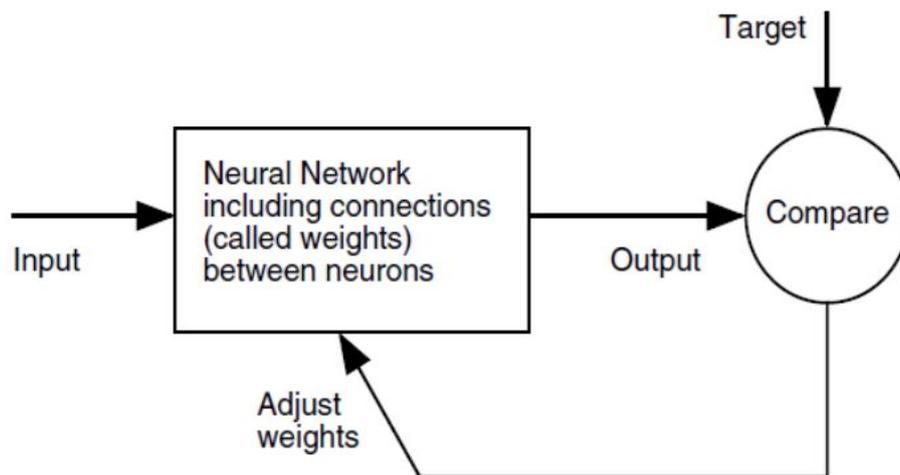


Figure 5-3. Flow loop of input to output via continuous adjustment of weights

(Demuth et al., 2009)

5.3 The Simple Neuron and Neuron with Vector Input

The most basic building unit for NNs is the single-input neuron, such as the one shown in Figure 5-4.

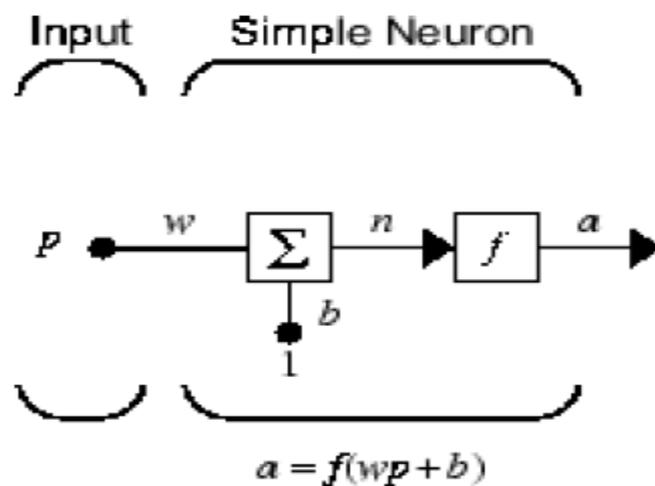


Figure 5-4. A simple neuron (Beale et al., 2014)

There are three distinct functional operations for the computation of output data from input data. Firstly, the input scalar, p , is multiplied by the weight, w . Secondly, the product of the weight and input scalar, p , is added to a bias and is the net input (n). The bias is similar to the weight except that it has a constant value of 1. Thirdly, the net input is passed through the transfer function, f . The operation of ' f ' produces the scalar output, a . For a neuron with vector input, p_1, p_2, \dots, p_j with weights w_1, w_2, \dots, w_j , the net input, n is the sum of the products of $p_1, p_2, \dots, p_\alpha$ and w_1, w_2, \dots, w_j , added to bias, b . This is shown as follows:

$$n = w_1p_1 + w_2p_2 + \dots + w_jp_j + b \quad (5.1)$$

$$\Rightarrow a = f(wp + b) \quad (5.2)$$

5.4 The Transfer Function

This is a function that calculates the output from a given NN net output. Transfer functions are of different types and are selected based on the characteristics of the input and output datasets (Vogl et al., 1988). Table 5-1 shows some of the transfer functions used for subsequent NN modelling in this thesis:

Table 5-1. Definitions of the transfer functions used (Beale et al., 2014)

Transfer Function	Algorithm	Definition
Purelin(n)	$a_1 = \text{purelin}(n) = n \quad (5.3)$	Linear transfer function
Logsig(n)	$a_2 = \text{logsig}(n) = \frac{1}{(1 + e^{-n})} \quad (5.4)$	Log-Sigmoid transfer function. Based on the logistic sigmoid function: $S(x) = \frac{1}{(1+e^{-x})} \quad (5.5)$
Tansig(n)	$a_3 = \text{tansig}(n) = \frac{2}{(1 + e^{-2n})} - 1 \quad (5.6)$	Hyperbolic tangent sigmoid transfer function. Based on the trigonometric function: $\tanh x = \frac{(1 - e^{-2x})}{(1 + e^{-2x})} \quad (5.7)$
dLogsig(n)	$\frac{d}{dn}\{a_2\} = \frac{d}{dn}\{\text{logsig}(n)\} = a_2(1 - a_2) \quad (5.8)$	Derivative function for logsig transfer function

5.5 Neural Network Architectures

The way in which neurons of a NN are structured is closely linked with the learning algorithm used in the training of the NN. It therefore follows that learning algorithms used in the design of NNs are structured. NN architectures are defined structural patterns for arranging neurons and are of three classes (Haykin, 1999).

5.5.1 Single-layer Feedforward Networks

These consist of a single layer of neurons from an input source node that projects to an output layer of neurons (computation nodes) and is acyclic. Figure 5-5 shows the arrangement of a typical single-layer feedforward network.

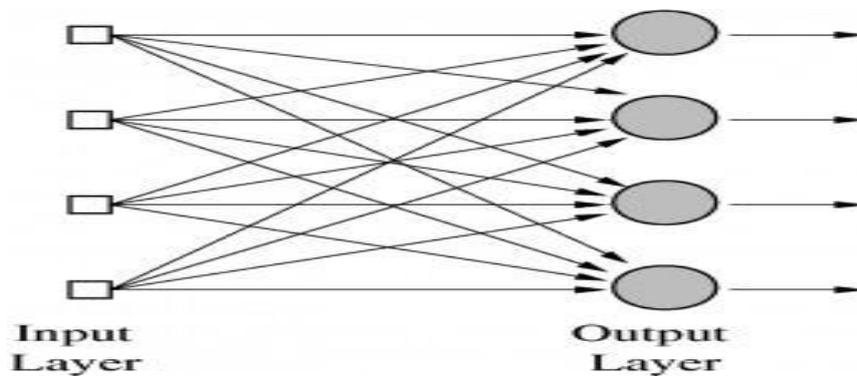


Figure 5-5. A Single-layer feedforward network

5.5.2 Multilayered Feedforward Networks

These are NN architectures with at least one hidden layer. The hidden layers thus perform the computation for this class of NN architecture (Haykin, 1999). Figure 5-6 shows the structure of a typical multilayered feedforward network.

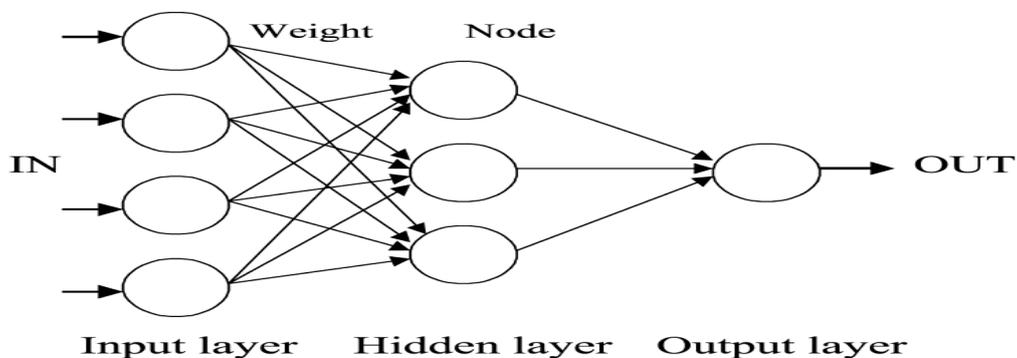


Figure 5-6. A multilayered feedforward network (Al-Fattah and Startzman, 2003)

The hidden layers in such architectures enable the NN to extract higher-order statistics and it acquires a global perspective which is particularly useful when the input layer is large (Churchland and Sejnowski, 1992).

5.5.3 Recurrent Networks

Recurrent networks are a class of NN architecture in which connections between neurons form a cyclic pattern with at least one feedback loop (Haykin, 1999). These networks are able to use their internal memory to process arbitrary sequences of inputs and makes them suitable for tasks such as unsegmented, connected handwriting recognition: (Graves et al., 2009; Sak et al., 2014). Recurrent networks can either have a self-feedback loop or not. The presence of the self-feedback loop implies that the output of a particular neuron is fed directly back into its input (Haykin, 1999). When the self-feedback loop is absent, it means the output of a given neuron is not fed into its input. The diagram of a recurrent network is shown in Figure 5-7.

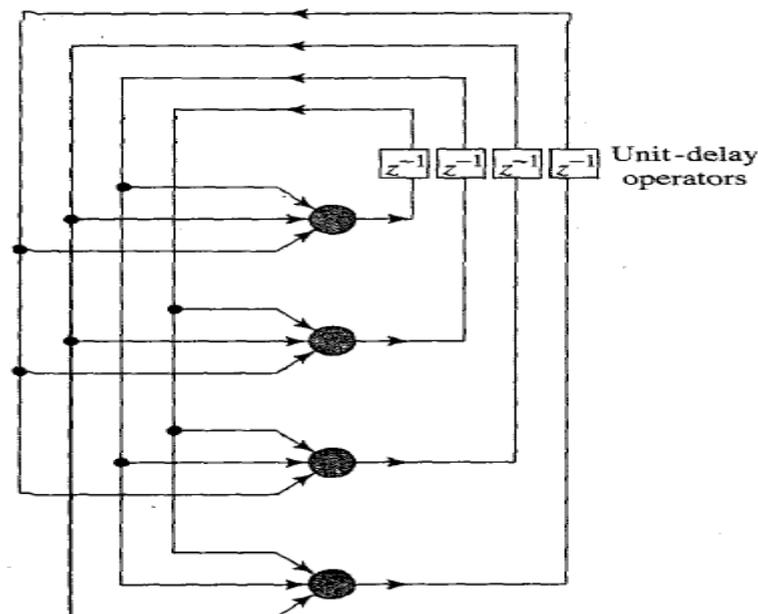


Figure 5-7. A recurrent network with no self-feedback loop (Haykin, 1999)

5.6 Design of the NN Model for Predicting CO₂ Corrosion

Figure 5-7 shows the schematic that was used to develop the procedural operation of the neural network (NN) model.

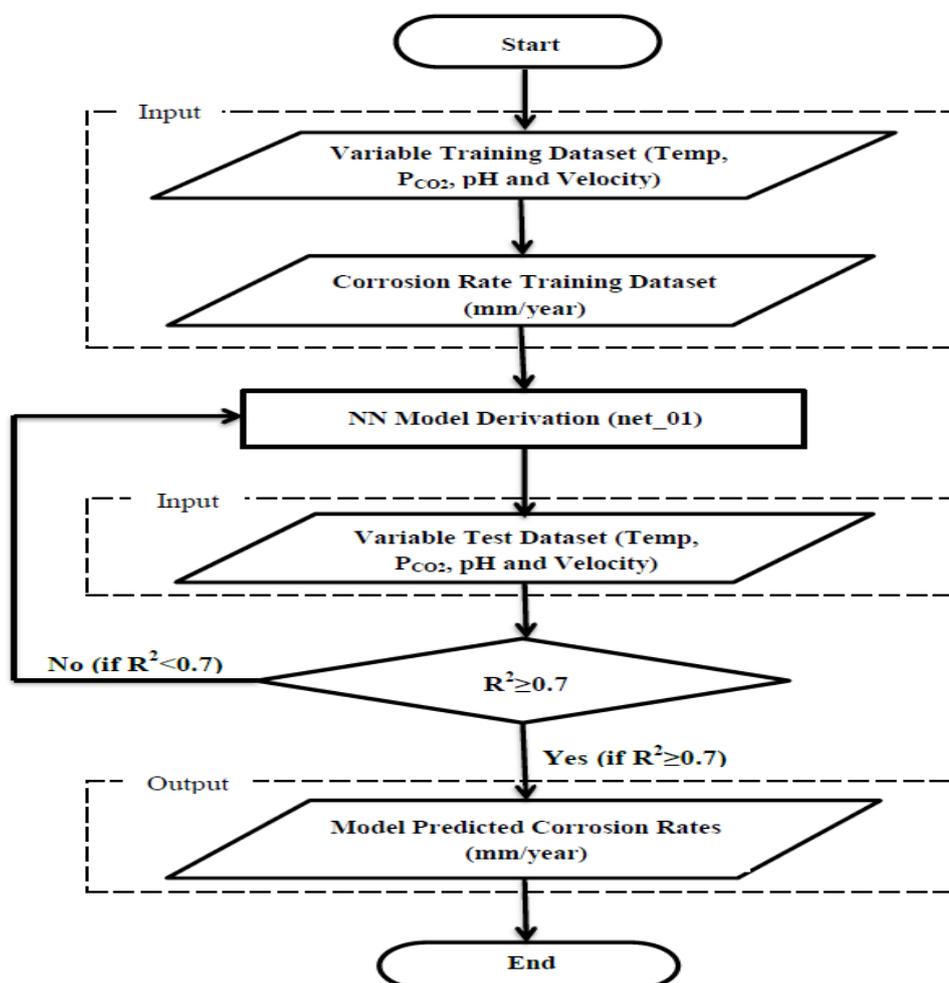


Figure 5-8. Flowchart showing the sequence of steps in the development of the neural network model

There are two sets of input to the NN model: the variable training dataset and the corrosion rate training dataset. Firstly, the NN is trained with the variable dataset which consists of the parameters which influence CO₂ corrosion rate, i.e., temperature, CO₂ partial pressure (P_{CO_2}), pH and flowrate velocity. Secondly, the NN is trained with the corresponding corrosion rate data. These corrosion rate data are direct laboratory measurements from corrosion rate experiments reported in the literature: (Dugstad et al., 1994b; Choi and Nesic, 2009; Hesjevik et al., 2003; Zhang et al., 2012a; Cui et al., 2006).

The NN model, net_01, is shown in Figure 5-8 as a process step in the flowchart sequence because the model is developed and defined by the two sets of input preceding it. The model

is then used for corrosion rate prediction. Figure 5-9 shows a block diagram view of the input and output datasets as well as the neural network, net_01.

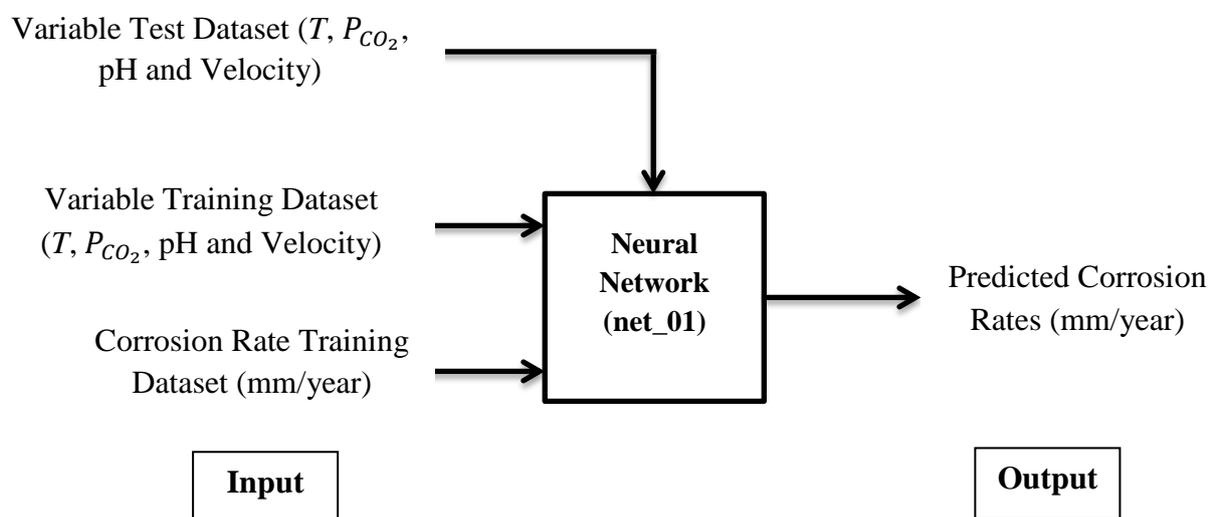


Figure 5-9. Block diagram of the neural network model

The variable test dataset is contained within the second input set and is used for testing. It contains parameters identical to those in the variable training dataset. The model reads this data and based on learned knowledge from the first input, yields output. These output results are then compared against actual corrosion rates from the original data in order to compute the correlation coefficient (R^2 -value). The method in developing the final NN model is based on the use of the R^2 -value, and is shown in Figure 5-8 as the decision step (Draper and Smith, 1998), (Abyaneh, 2014). The condition, $R^2 \geq 0.3$, was chosen in the initial testing phase in order to characterise the selected training functions to be used: a low-level screening process to rule out undesirable training functions. For fine-tuning of the final NN model, however $R^2 \geq 0.7$ was used and then later increased to $R^2 \geq 0.8$, for the final model result. R^2 -coefficients: $R^2 \geq 0.7$ and $R^2 \geq 0.8$ were selected as further fine-tuning (pruning) criteria. The actual corrosion rates are a direct consequence of the parametric conditions contained in the variable test dataset, measured from experimental corrosion rate plots. The errors are calculated by subtracting the actual (experimental) CO_2 corrosion rates from model predicted CO_2 corrosion rates. Absolute values of these errors are then evaluated and then results are summed up to obtain the ‘Sum Total of Absolute Error’. Equations for these expressions are shown as follows:

$$\text{Corrosion Rate Error} = \text{Model Predicted Value} - \text{Actual Value} = y_i - x_i \quad (5.9)$$

For any given plotted i as shown in Figure 5-10:

$$\text{Absolute Error} = |\text{Corrosion Rate Error}| = |y_i - x_i| \quad (5.10)$$

For any given plotted i as shown in Figure 5-9:

$$\text{Sum Total of Absolute Error} = \sum(\text{Absolute Error}) = \sum_{i=1}^n |y_i - x_i| \quad (5.11)$$

For all plotted points n shown in Figure 5-9:

$$R^2 = \left(\frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \right)^2 \quad (5.12)$$

Where x_i and y_i are the i^{th} observation and model predicted values respectively. \bar{x} and \bar{y} are the mean values of x_i and y_i and n is the total number of data-points: (Draper and Smith, 1998; Abyaneh, 2014). The arbitrary x - y plot of predicted against experimental corrosion rate in Figure 5-10, illustrates how the equations for the error expressions are obtained.

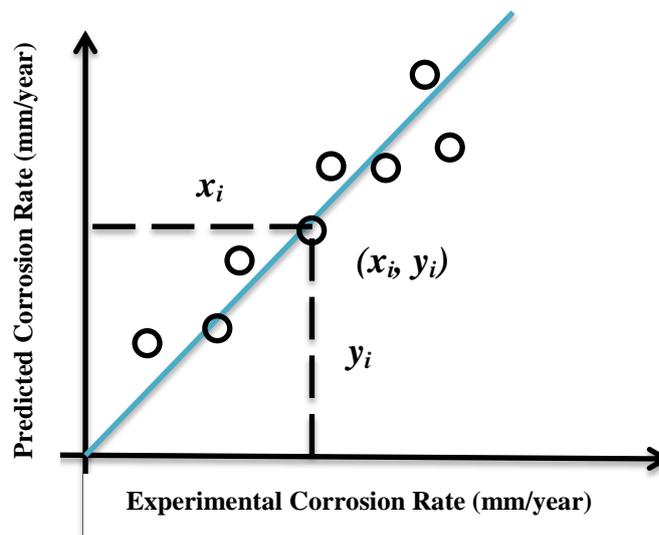


Figure 5-10. Arbitrary plot of predicted corrosion rate against experimental corrosion rate

NN training – Determination of the Final Model Parameters

In order to determine the final model parameters such as size of the network, the number of layers (network configuration) and the type of training function(s) to use, simulation runs were performed using Matlab 2012a version on a Windows 7, 64-bit system with 6 gigabytes (GB) of random access memory (RAM).

Given that in any given neural network model, the transfer function converts input data to a given output, trial runs to determine the final NN model, for the purpose of this study, are broadly based on the type of transfer function (Vogl et al., 1998). Hence simulation tests were carried out on two main groups – the logsig and tansig transfer function groups. The reason for their selection as the head of their respective groups is that their respective algorithms allow them to accept data of any magnitude (negative to positive infinity) whilst returning an output in the range 0 to 1 for the former and -1 to 1 for the latter, thus making them the most suitable to use as a starting transfer function in the network (Beale et al., 2014). Training algorithms are a set of instruction code that governs how the NN will be optimised such as the setting of weights and biases, for instance (Beale et al., 2014). There are three types of training algorithms and these are: the gradient descent, conjugate gradient and quasi-Newton algorithms (Sharma and Venugopalan, 2014).

The gradient descent algorithms evaluate function gradients at the initial guess value and take steps proportional to the negative direction of the gradient in order to find a local minimum. In conjugate gradient methods, the local minimum is located by searching along conjugate directions. This implies that a faster convergence is achieved than with the gradient-descent method. The quasi-Newton algorithms usually give better and faster optimisation than conjugate methods however a greater amount of memory is required (Sharma and Venugopalan, 2014).

Training functions are network functions that dictate a universal algorithm which sets weights and biases in any given network to optimise performance. Matlab training functions can be classified based on the type of training algorithm they use – all Matlab training functions thus fall into one of the three given training algorithms (Sharma and Venugopalan, 2014). For this study, two training functions were selected from each of the training algorithm groups to make a total of six functions. The definitions of each of these functions are shown in Table 5-2.

Table 5-2. Definitions of the training functions used (Beale et al., 2014; Sharma and Venugopalan, 2014)

Training Function	Training Algorithm Group	(Definition) Updates network weights according to:
Traingdm	Gradient Descent	Gradient descent with momentum
Trainrp	Gradient Descent	Resilient backpropagation (Rprop)
Trainscg	Conjugate Gradient	Scaled conjugate gradient method
Traincgf	Conjugate Gradient	Conjugate gradient backpropagation with Fletcher-Reeves updates
Trainbfg	Quasi-Newton	Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method
Trainlm	Quasi-Newton	Levenberg-Marquardt optimisation

The overall performance of each Matlab transfer function – logsig and tansig for this study, is an aggregate of the individual performances of the training functions listed in Table 5-2. Figure 5-11 is a pictorial representation of how the degree of suitability of the transfer functions for determining the final NN model parameters is dependent on the combined performances of the individual training functions. The testing phase of the NN model development involved the use of two transfer functions, each with several training algorithms as shown, in Figure 5-10.

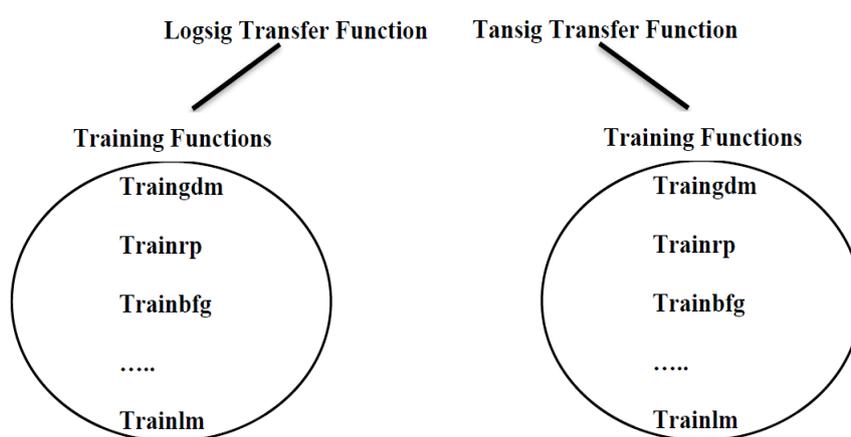


Figure 5-11. Combined performances of each training function defines the degree of suitability of the transfer function in determining the final NN model parameters

5.6.1 NN Modelling of Low Pressure CO₂ Corrosion Data

Summary of Training and Testing data for NN Modelling

Data from Dugstad et al. (1994b), Nordsveen et al. (2003) was used in developing the NN model. The information displayed in Table 4-1, Section 4.2.1 is a detailed summary of the collated data and the descriptive statistics. In total, there are 75 data-points and these were divided into training and testing sets with 60 and 15 data-points for each set respectively. A summary of the test set variables are shown in Table 4-2, Section 4.2.1.

Results

Figure 5-12 shows the plot of the sum total of the absolute error against number of neurons for the logsig transfer function.

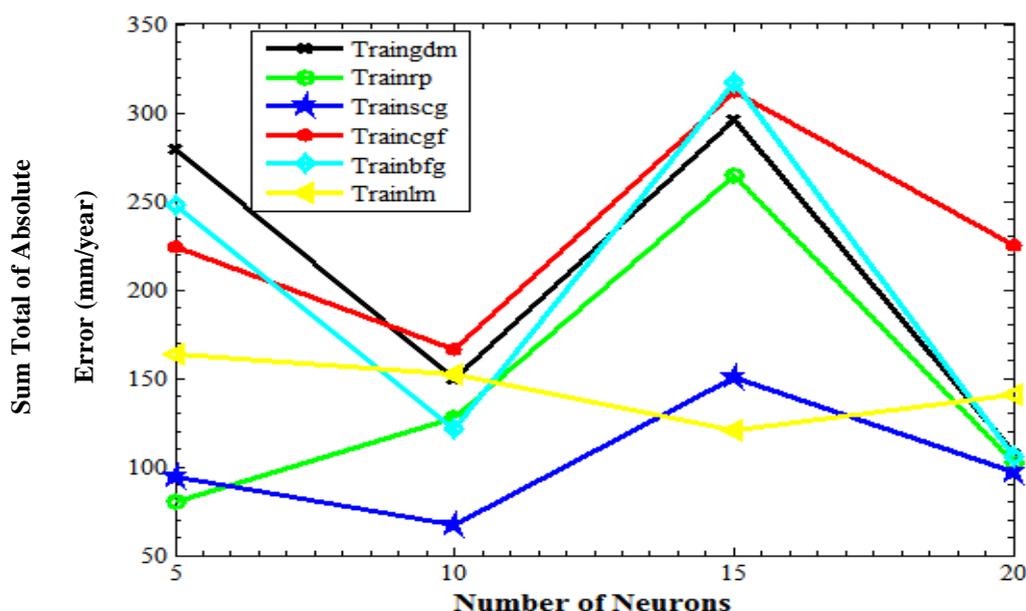


Figure 5-12. Line plots of the sum total of absolute error against number of neurons for the logsig transfer function

The sum total of absolute errors decreases with increasing number of neurons for half of the training functions tested, namely: traingdm, trainbfg and trainlm. Of all the training algorithms tested, it is observed that the trainscg and trainlm functions have the lowest overall absolute errors. This implies that predicted corrosion rates were consistently closer in magnitude to the actual corrosion rates from the original data source (Dugstad et al., 1994b), (Nordsveen et al., 2003) for the given range of number of neurons. The maximum number of

neurons tested was set at 20 to avoid overtraining. The sum total of absolute errors for trainscg is approximately 100mm/year, which is equivalent to $100/15 \approx 6.67\text{mm/year}$ (for 15 data-points).

From Figure 5-13, it is observed that for half of the training functions (traincgm, trainbfg and trainlm), the magnitude of the coefficient of determination (R^2 -coefficient) increased from 5 neurons to 20 neurons. Again, the trainlm function outperformed all others in terms of the magnitude of the R^2 -value.

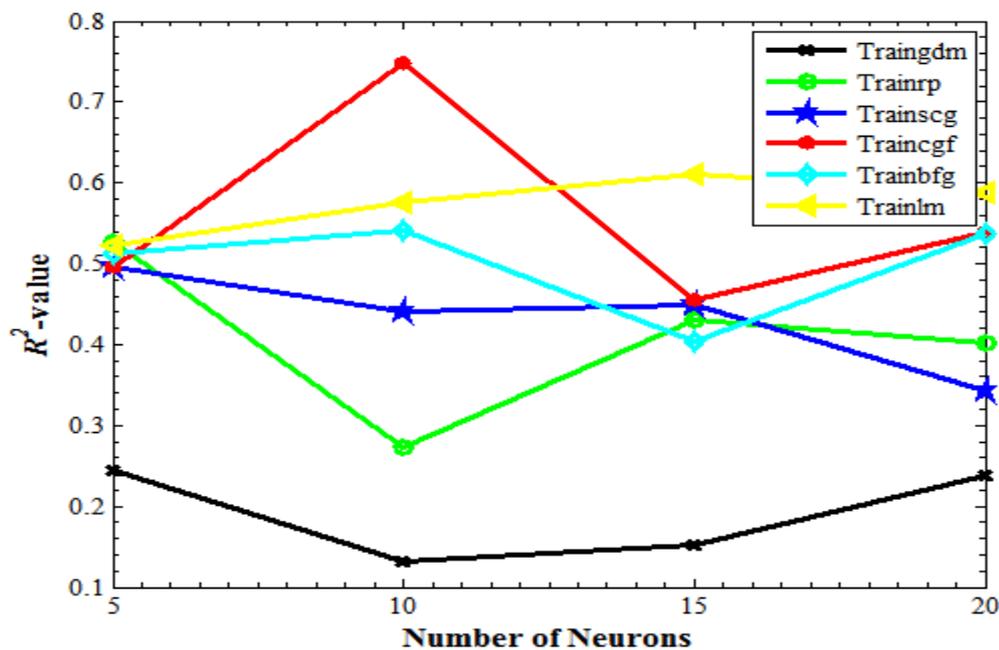


Figure 5-13. Line plots of R^2 -value against number of neurons for the logsig transfer function

The R^2 -value line plots for the tansig function are shown in Figure 5-14. The magnitudes of the R^2 -coefficients increase with increasing number of neurons in a similar trend to that seen in Figure 5-13. The trainlm function has the highest R^2 -values indicating greater precision in prediction. It is also noted that while the R^2 -coefficients for each training function fluctuates from 5 to 20 neurons for both logsig and tansig transfer functions, the only exception is that of the consistently high-levelled trainlm.

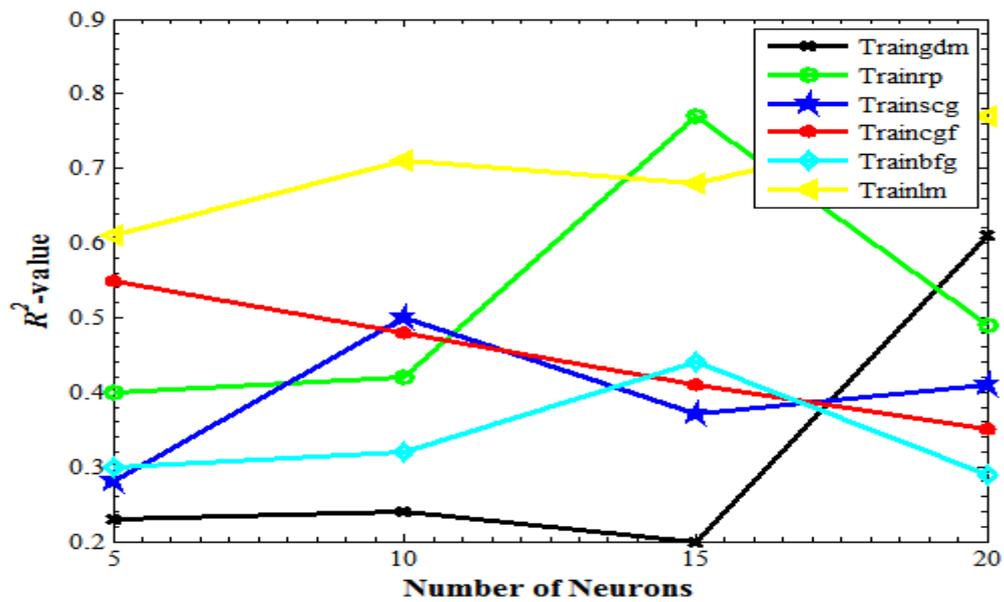


Figure 5-14. Line plots of R^2 -value against number of neurons for the tansig transfer function

The bar charts for the mean R^2 -coefficients for each training function from the logsig and tansig transfer function tests shown in Figure 5-15, confirm that the trainlm functions consistently performs better than all the others.

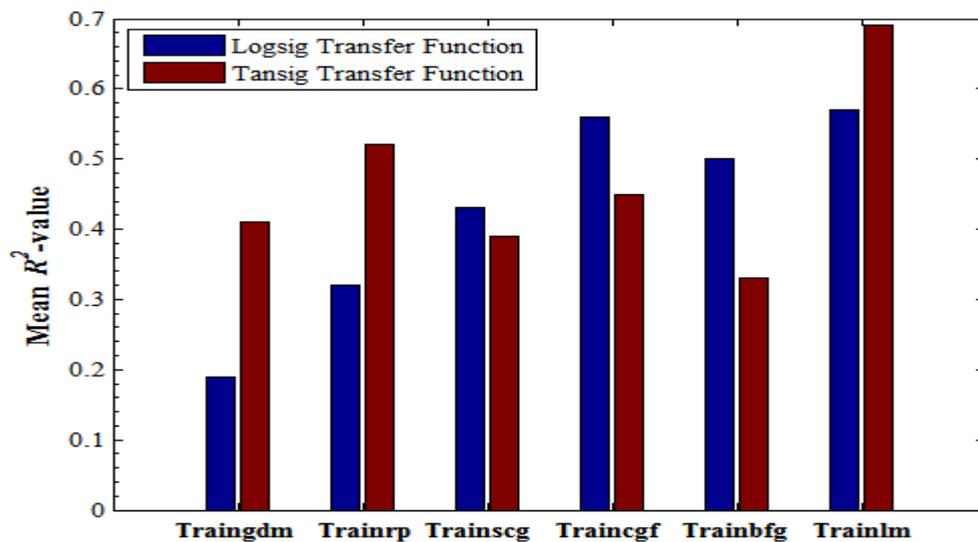


Figure 5-15. Multiple bar charts showing the mean R^2 -coefficients for each training function for both logsig and tansig transfer functions

It is also noted that the trainlm, being a quasi-Newton algorithm, is generally considered to be better than the other training functions at optimisation as they converge faster in performing nonlinear computations (Sharma and Venugopalan, 2014). Furthermore, there is additional evidence of better optimisation capability when observing the bar charts of Figure 5-16. It has the least average absolute corrosion rate error, indicating that it is the most suitable training function for the NN model due to its greater accuracy and precision.

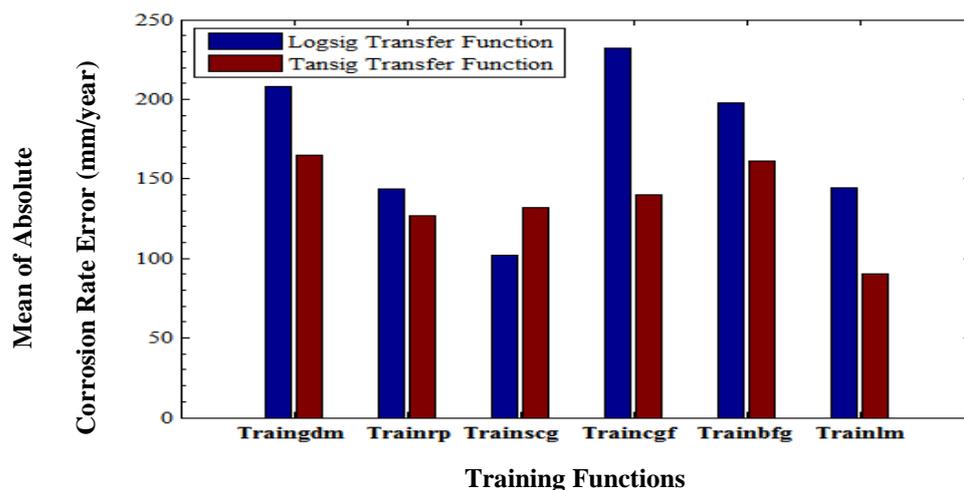


Figure 5-16. Multiple bar charts showing the absolute error means for each training function for both logsig and tansig transfer functions

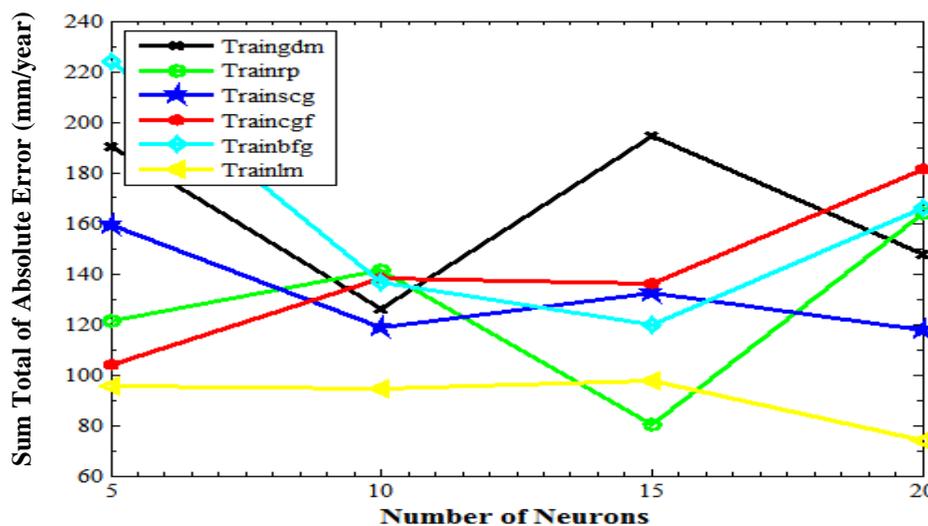


Figure 5-17. Line plots of sum total of absolute error against number of neurons for the tansig transfer function

Figure 5-17 shows the plot of the sum total of absolute errors against the number of neurons for the tansig transfer function. It is observed that for the training functions, namely, traingdm, trainscg, trainbfg and trainlm, there is a decrease in the sum total of absolute error with increasing number of neurons. The reason for this being that the NN model has a greater capacity to learn about the inherent patterns in the dataset more easily than when there are too few neurons (Alsmadi et al., 2009). Again, the sum total of absolute errors for trainlm is approximately 95mm/year, which is equivalent to $95/15 \approx 6.33$ mm/year (for 15 data-points).

Discussion of the Final Neural Network Model Specifications

Based on the discussed NN model tests, it was thus concluded that trainlm and tansig were the respective training and transfer functions to be utilised in determining the optimum neuron configuration. It was also found from these tests that 10 and 15 neurons were sufficient to obtain an accurate CO₂ corrosion prediction system as these yielded the best combination of total absolute errors and R^2 -coefficients.

The number of layers to be used was determined by testing four different neuron configurations in multiples of 5 as shown in Table 5-4. Essentially, the total number of neurons was distributed in an organised manner within layers.

Table 5-4. Summary of the results obtained for the tested neuron configurations

Neuron Configuration	Total Number of Neurons	R^2 -value
[5 5]	10	0.71
[5 5 5]	15	0.66
[5 10]	15	0.72
*[10 5]	15	0.91

* The layers are arranged with 10 neurons in the first layer and 5 neurons in the second layer

From the R^2 -coefficients of Table 5-4, it is seen that the [10 5] neuron configuration is the final choice for developing the NN model. It produces the most accurate CO₂ corrosion predictions. A summary of the NN model properties is presented in Table 5-5.

Table 5-5. Summary of neural network properties

Neural Network Property	Value/Function
Number of neurons and configuration	15 neurons, configuration= [10 5]
Number of layers	2
Training function	Levenberg-Marquardt (trainlm)
Transfer functions	tansig-tansig

The plot in Figure 5-18 shows the performance of the 15-neuron, [10 5] configuration, 2-layer neural network model. The magnitude of the R^2 -value and the closeness of all data-points to the 95% confidence intervals in Figure 5-18 indicate the model is accurate. The number of points outside of the upper boundary of the 95% confidence interval suggests some degree of over-prediction is occurring which may be due to variation within the dataset at low-mid temperatures.

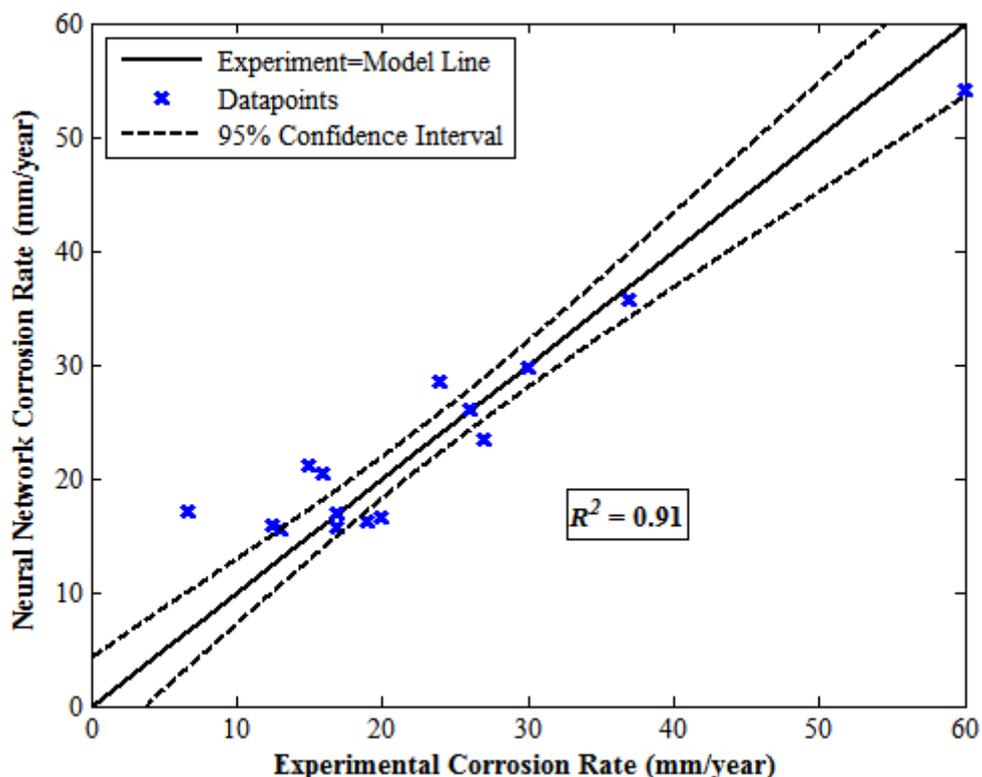


Figure 5-18. NN model corrosion rate against experimental corrosion rate

Figure 5-19 shows the error bar chart for the NN model. This plot highlights that there is over-prediction occurring for small corrosion rates.

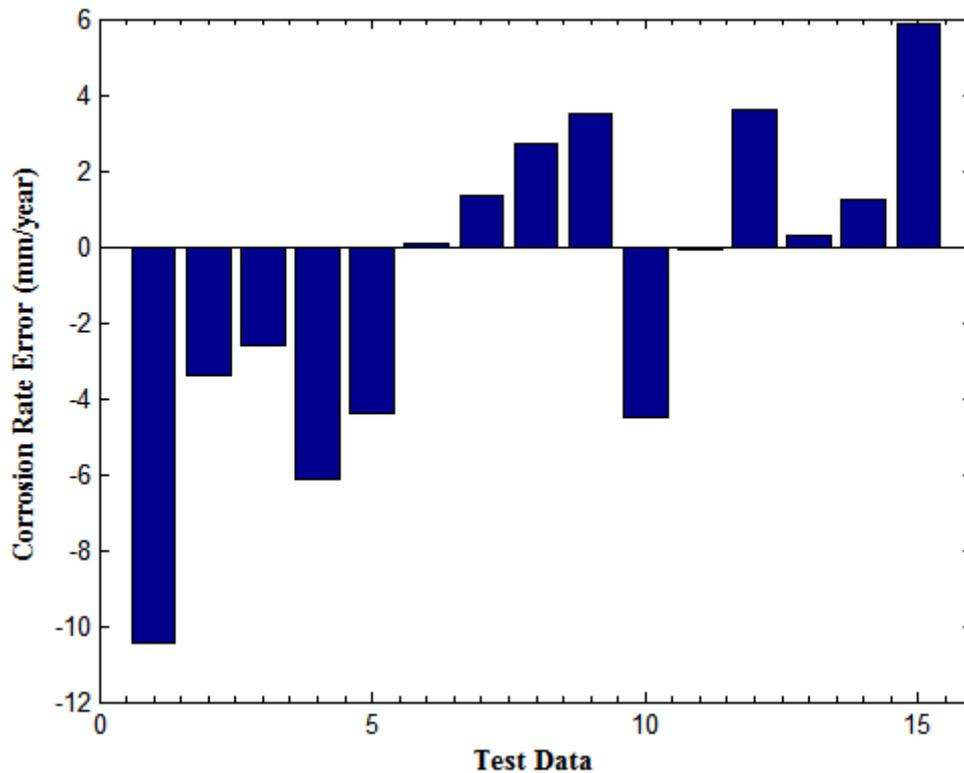


Figure 5-19. Error bar chart

5.6.2 NN Modelling of High Pressure CO₂ Corrosion Rate Data

Summary of Training and Testing data for NN Modelling

Data from multiple sources was used in developing the NN model: (Choi and Nestic, 2009; Cui et al., 2006; Hesjevik et al., 2003; Zhang et al., 2012a). The information displayed in Table 4-5, Section 4.3.1 is a detailed summary of the collated data and the descriptive statistics. In total, there are 22 data-points and these were divided into training and testing sets with 16 and 6 data-points for each set respectively. The bar chart of Figure 5-20 shows the distribution of the given data-points from each of the sources.

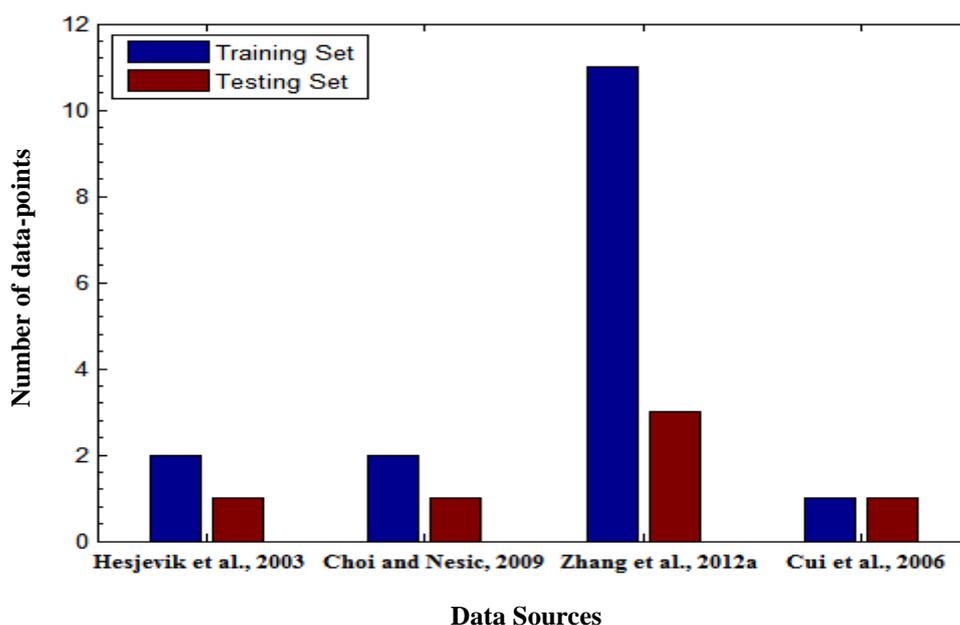


Figure 5-20. Bar chart showing the distribution of data-points from each source

Overall the number of data-points from the Zhang et al (2012a) study exceeds those of the other sources (63% share) due to the fact that the experimental corrosion rate results for this source were carried out for the widest range of temperatures (50-130°C) and pressures (9.5-23.3MPa).

For the other sources, corrosion rate tests were carried out by maintaining a constant temperature whilst varying pressures or maintaining a constant pressure while varying temperatures as is the case with the Choi and Nestic (2009) and Cui et al (2006) studies respectively. For the study by Hesjevik et al (2003), tests were focused on measurement of corrosion rates for temperatures less than 30°C. Experimental corrosion rate measurements were grouped into classes of 0-4, 4-8, ..., 16-20mm/year. Figure 5-21 shows the distribution of corrosion rates in these classes.

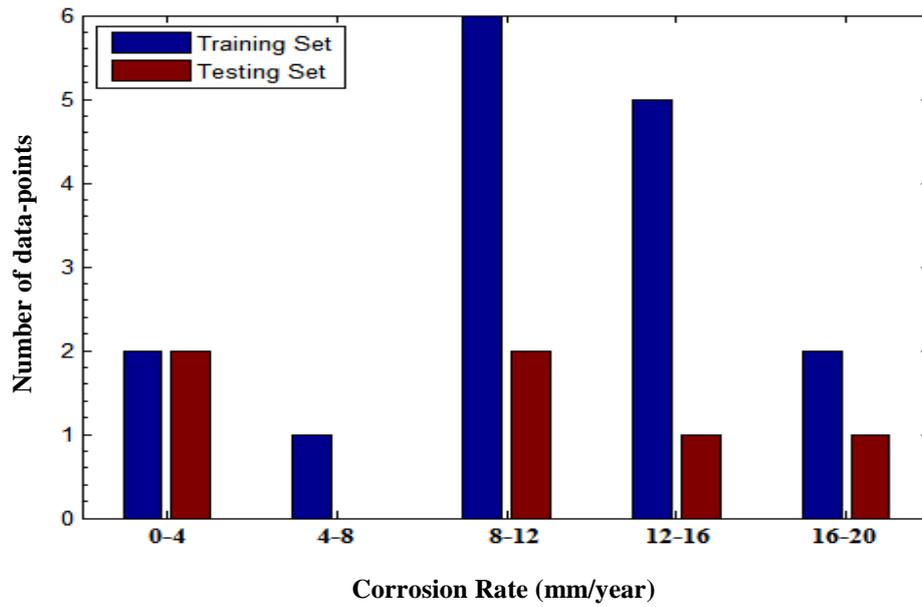


Figure 5-21. Bar chart showing the distribution of data-points for the grouped experimental corrosion rates

From Figure 5-21, the number of data-points for the mid-corrosion rate magnitude (8-12mm/year) is greater than those for end-point corrosion rate groups (0-4 and 16-20mm/year).

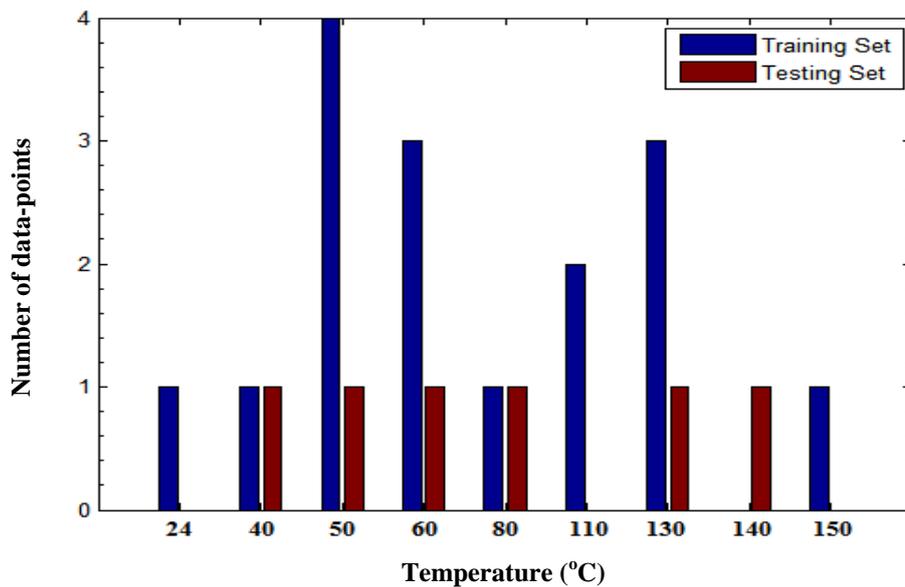


Figure 5-22. Bar chart showing the distribution of data-points for the recorded temperatures

The bar chart in Figure 5-22 shows the distribution of data-points for the recorded experimental temperatures. There are more data-points in the mid-temperatures (50 and 60°C) than for end-point temperatures (24 and 150°C).

Figure 5-23 shows the corrosion rate-temperature profile for the training dataset. A polynomial curve fit through the points depicts the classic peak observed for CO₂ corrosion rate as a function of temperature (De Waard and Lotz, 1993). It is noted that the statistical range of corrosion rates in the mid-temperatures (50-80°C) is ~11mm/year, highlighting that the greatest variation in the magnitudes of corrosion occurs in these temperatures.

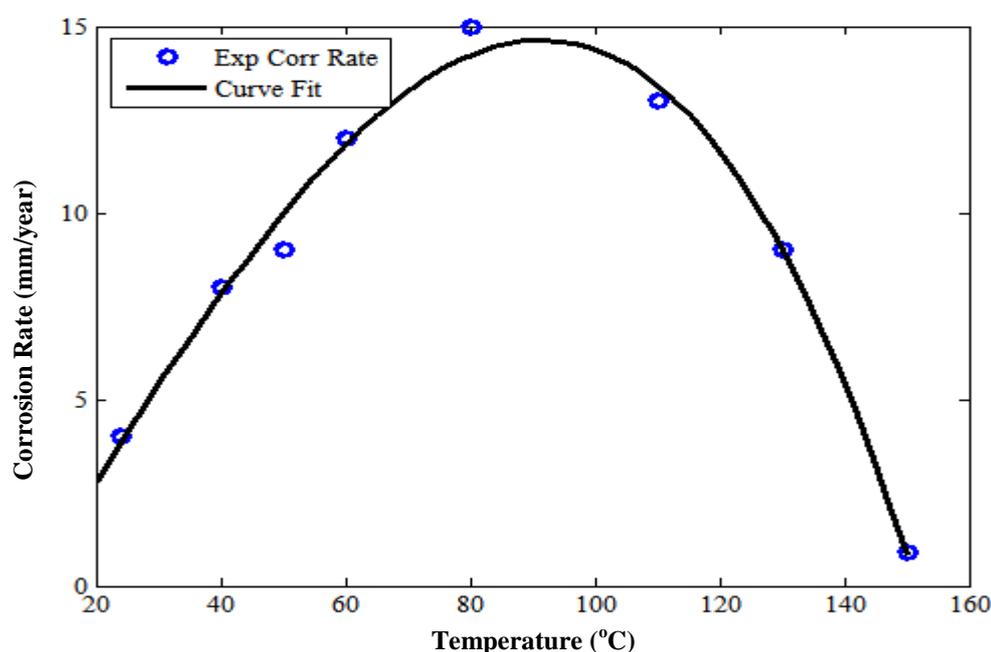


Figure 5-23. Corrosion rate against temperature plot for the training dataset

Results

The variation of the magnitudes of R^2 -coefficients of the training functions with number of neurons for the logsig transfer function is shown in Figure 5-24. There is a general decrease in the magnitude of the R^2 -coefficients with increasing number of neurons for all training functions except for the Trainscg. This fall in the magnitude of the correlation coefficient with increasing number of neurons indicates that model performance is decreasing despite the increasing network size. This may be due to the presence of a greater number of redundant nodes or synaptic weights in the NN or that training is possibly diverging, resulting in predicted values being very different from expected results (Haykin, 1999). Other reasons

may be that the training dataset is small or that the inherent characteristic of the data is such that modelling with an increasing number of neurons beyond 5, results in a fall of the R^2 -value. A similar trend was observed in an environmental research study (Abyaneh, 2014).

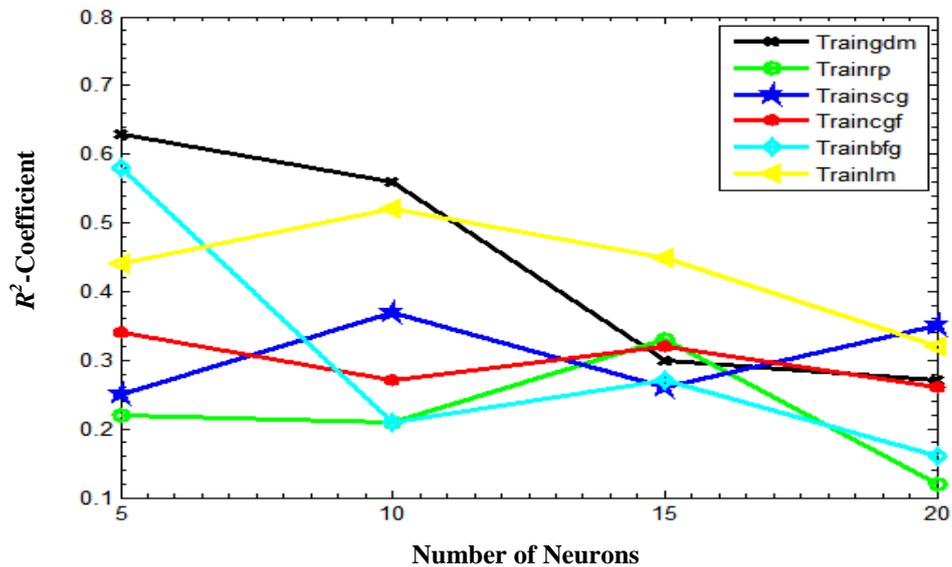


Figure 5-24. Line plots of R^2 -values of training functions against number of neurons for the logsig transfer function

For the tansig transfer function, Figure 5-25 is the variation of the magnitudes of R^2 -coefficients against number of neurons.

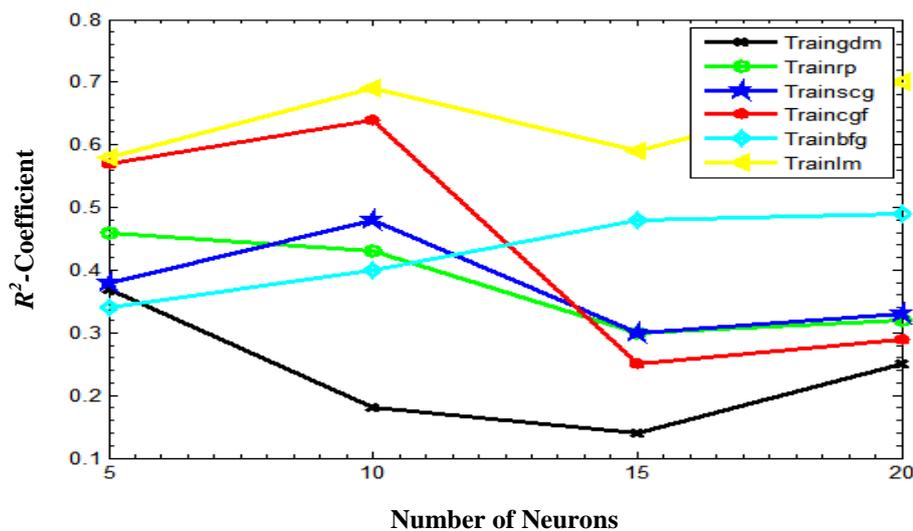


Figure 5-25. Line plots of R^2 -values of training functions against number of neurons for the tansig transfer function

There is a general dip in the magnitudes of the R^2 -coefficients as the number of neurons increases from 5 to 20. Again, given that the training dataset is small, there is greater certainty that as the number of neurons increases in the NN, the number of excess weights also increases and there is a tendency for their presence to reduce model accuracy (Haykin, 1999). The only exceptions are the quasi-Newton algorithms, `trainbfg` and `trainlm`. The quasi-Newton algorithms are generally known to give better optimisation results than the others, particularly for network sizes that are not large Beale et al., (2014), Sharma and Venugopalan, (2014).

Figure 5-26 shows the performances of each training function for both `logsig` and `tansig` transfer functions in terms of the average R^2 -value.

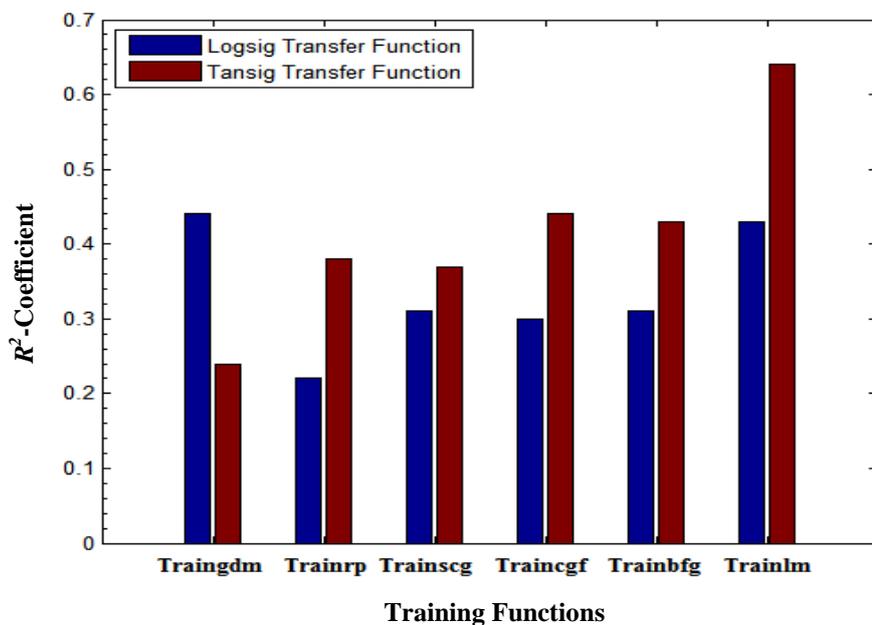


Figure 5-26. Bar charts showing the means of the R^2 -values of each training function for both `logsig` and `tansig` transfer functions

The `trainlm` function again outperforms all the others. The `traincgf` and `trainbfg` rank second and third respectively on the R^2 -value performance test. It is also worth noting that the training functions mostly perform better with the `tansig` transfer function than the alternative `logsig` transfer function, the only exception being the `traingdm` function. This may be due to the effect of the unique combination of the input data, the `logsig` transfer function and the algorithm in the `traingdm` training function culminating in a good performance level. Overall, a steady, high magnitude of R^2 -coefficient is maintained by the `trainlm` function with an

increasing number of neurons. This thus shows its suitability for use in developing the final NN model.

The training functions performances were also assessed in terms of the sum of their absolute errors. Figure 5-27 is the bar chart variation of the sum of absolute errors of the training functions against number of neurons.

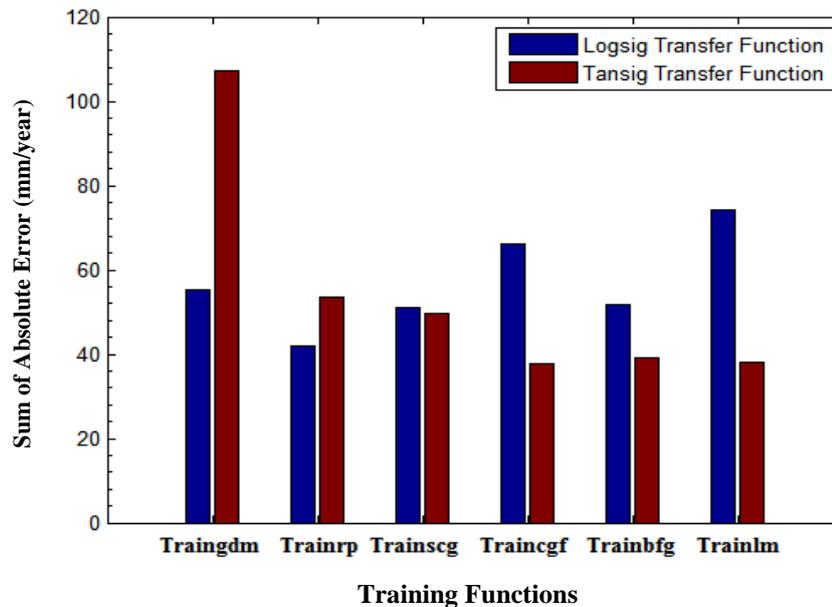


Figure 5-27. Bar charts showing the means of absolute corrosion rate error of each training function for both logsig and tansig transfer functions

It is seen that the traingdm has the least favourable performance in terms of the mean absolute corrosion rate errors obtained while the trainrp and trainbfg have the smallest errors for both logsig and tansig transfer functions as a combination. On a singular basis, the trainlm has the least error when the tansig function is in use, followed closely by the traincgf function. In summary, the best training functions appear to be the trainlm and traincgf. It must be borne in mind that for all modelling tests conducted and discussed so far, the condition applied is that $R^2 \geq 0.3$.

Discussion of the Final NN Model Specifications

The two functions used in this stage are the traincgf and trainlm functions. The best performances were obtained when the tansig transfer function was applied. In the initial testing stages, it was discovered that very few neurons were required to achieve desirable

results as shown in Figures 5-24 and 5-25. In particular, the NN performances were relatively high for neuron numbers of 5 to 15. The number of layers to be used was determined by testing four (4) different neuron configurations in multiples of 5 as shown in Table 5-6.

Table 5-6. Summary of the results obtained for the tested neuron configurations

Neural Network Property	Value/Function
Number of neurons and configuration	10 neurons, configuration= [5 5]
Number of layers	2
Training function	Levenberg-Marquardt (trainlm)
Transfer functions	tansig-tansig

Figure 5-28 shows the comparative line plots for both traincgf and trainlm functions.

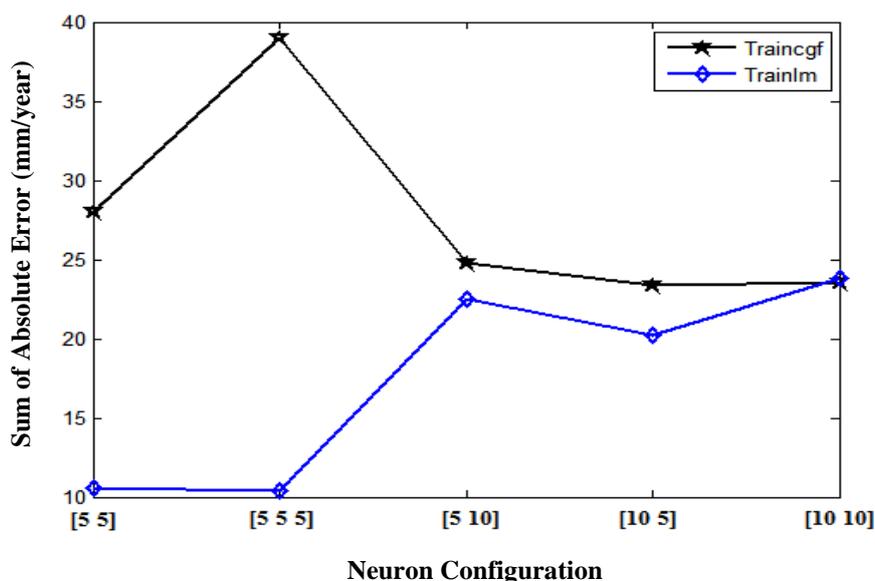


Figure 5-28. Line plots of sum of absolute corrosion rate error against neuron configuration for the traincgf and trainlm functions

From Figure 5-28, the trainlm function consistently has the least absolute corrosion rate errors implying that it is a better training function for developing the final NN model. It is also noted that NNs with one hidden layer often encounter difficulties with approximation and model-fitting due to global interaction between neurons. On the other hand, NNs with two hidden layers have no such issues, as the first layer extracts local features within the data while the second layer extracts the global features (Abyaneh, 2014).

Also, from Table 5-7, the [5 5] neuron layer configuration offers the highest R^2 -value. This value coincides with the smallest corrosion rate error, as seen in Figure 5-28. This corrosion rate error is equivalent to $10.61/6 \approx 1.77$ mm/year per plotted data-point on the NN model corrosion rate against experimental corrosion rate plot shown in Figure 5-29.

Table 5-7. Summary of neural network properties

Neuron Configuration	Total Number of Neurons	Sum of Absolute Error (mm/year)	R^2 -value
[5 5]	10	10.61	0.91
[5 5 5]	15	10.38	0.88
[5 10]	15	22.46	0.68
[10 5]	15	20.19	0.76
[10 10]	20	23.82	0.80

The plots in Figure 5-29 and Figure 5-30 show the performance of the 10-neuron, [5 5] configuration, 2-layer neural network model. The magnitude of the R^2 -value and the closeness of all data-points to the 95% confidence intervals in Figure 5-29 indicate the model is highly accurate. Figure 5-31 shows the corrosion rate-temperature profile for the test dataset with corresponding NN model predictions. The best fit line is a polynomial function with an R^2 -coefficient of 0.55. A higher R^2 -value can be attained however this results in a distortion of the overall shape and is badly conditioned. There is an increase in corrosion rates as temperatures increase from 20°C to 80°C owing to temperatures accelerating the chemical and electrochemical corrosion reactions. The precipitation rate though is said to increase as temperatures increase hence protective layers form on the metal surface leading to a reduction in corrosion rate: Johnson and Tomson, (1991), Yin et al., (2009).

NN predictions are close to the test data results shown in Figure 5-31, appearing mostly as pairs of data-points. There is a slight degree of inaccuracy in model predictions for the mid-temperature region of 50-80°C and this is attributed to greater variation in the training and test set corrosion rates.

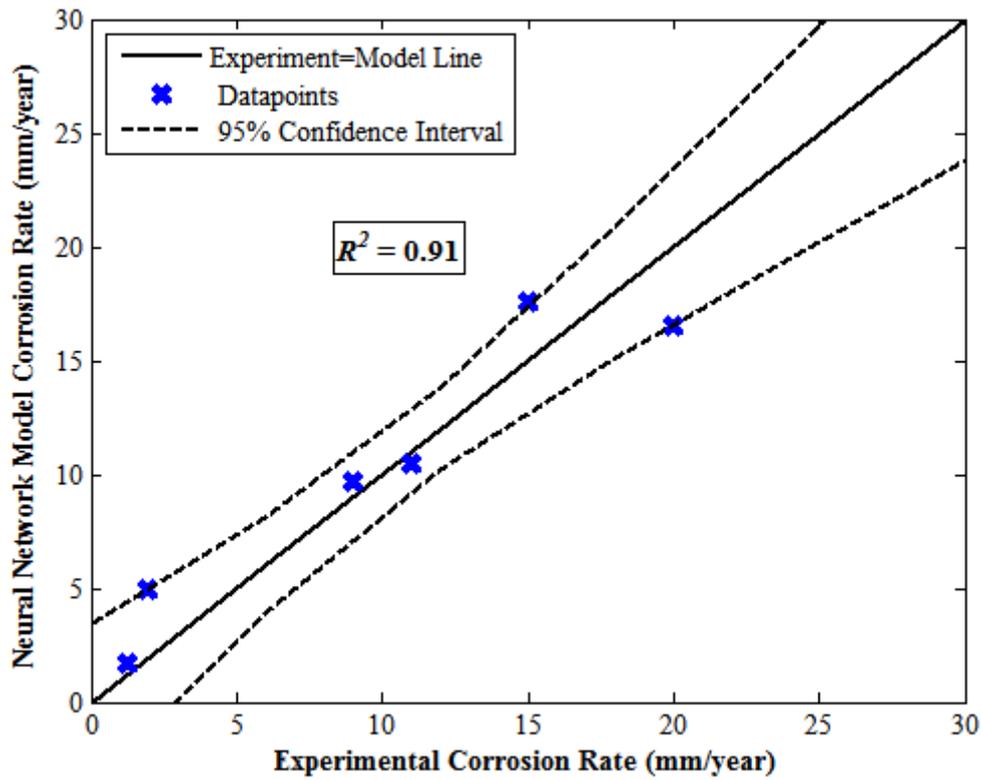


Figure 5 -29. NN Model corrosion rate against experimental corrosion rate plot

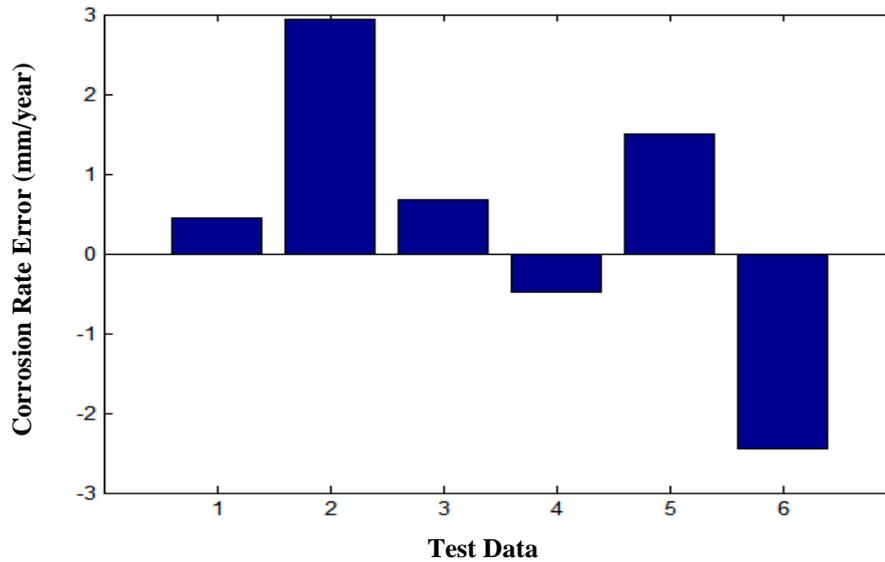


Figure 5-30. Error bar chart – Error is the difference between experimental and model corrosion rate

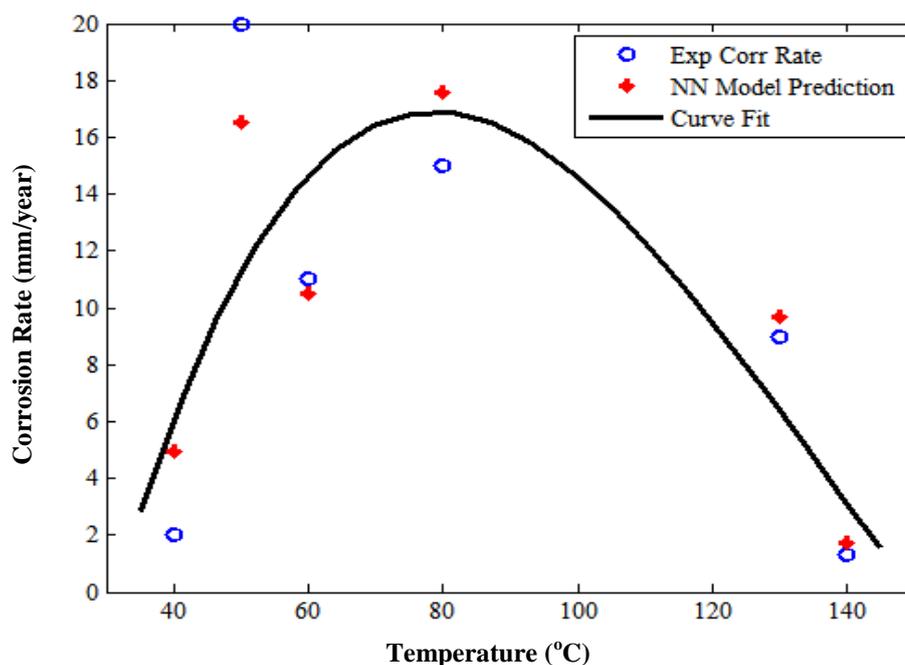


Figure 5-31. Corrosion rate against temperature plot for the testing dataset with NN model predictions

5.7 Conclusions

For both low and high pressure CO₂ corrosion data, the tansig transfer function consistently outperformed the logsig transfer function for most training functions with the exception of the traingdm and trainrp training functions, in terms of the absolute corrosion rate errors. As these two algorithms are gradient descent functions, their under-performance relative to the conjugate gradient and quasi-Newton algorithm function alternatives is to do with the technique of finding and locating local minima while carrying out nonlinear optimisation.

Also, in terms of the magnitude of the R^2 -coefficient, the tansig transfer function offered better results for all training functions except the traingdm. Again, the quasi-Newton algorithm functions, trainbfg and trainlm are known to offer better optimisation results.

The R-squared coefficient for the final neural network model is 0.91 for both low and high pressure CO₂ datasets. The models provide good fits for the corrosion test set. For the low pressure CO₂ corrosion NN model, all test points are in close vicinity to the 95% confidence bounds indicating a high degree of accuracy however for test experimental results of lower magnitude; there is a slight tendency for over-prediction. For the high pressure CO₂ corrosion

NN model, all test points are within the bounds of the 95% confidence limits indicating a high degree of accuracy. The error test bars indicate that predicted results are within ± 3 mm/year of the CO₂ corrosion test set.

As is usually the case with CO₂ corrosion, the mid-temperature experimental values of $\sim 60^\circ\text{C}$ induce the greatest corrosion rates due to the Arrhenius theory - the fact that higher temperatures speed up corrosion rates by providing greater amounts of kinetic energy to reacting species on a molecular level. However the limiting factor preventing a continuous induced elevation of CO₂ corrosion rates beyond levels of the mid-temperature values to higher levels of the very high end-point temperatures (110-150°C), is the consequent formation of iron carbonate (FeCO₃) and magnetite. Though the ferrous ion concentration ([Fe²⁺]) increases in solution with higher temperatures, solubility decreases, resulting in scale formation which coats the metal surface and brings about a decrease in corrosion rates.

This phenomenon reflects on both training and testing datasets with mid-point temperatures coinciding with very high corrosion rates. Also, in these datasets, and in particular, the high pressure CO₂ dataset, 50% of the data-points lie around the 50-80°C range, which represents a range of ~ 1 mm/year in the training dataset – a wide variation.

NNs tend to predict less accurately when trained with data of considerable variation. This is depicted in the model vs experimental plot where the high corrosion rate points are on opposite bounds of the 95% confidence limits while other data-points are not as widely-spaced apart on the plot. A summary of the NN model performances for low and high pressure is presented in Table 5-8.

Table 5-8. Model performance for each dataset

NN Model	R^2-value
Low Pressure CO ₂ dataset	0.91
High Pressure CO ₂ dataset	0.91

Chapter 6. Fuzzy Inference Systems

6.1 Introduction – Fuzzy Logic

In chapters 4 and 5, derived models were obtained by statistical and neural network modelling respectively. In this chapter, a fuzzy model will be derived for the given low and high pressure CO₂ corrosion datasets.

The model-developing data was summarised as descriptive statistics, given in Tables 4-1 and 4-5 (Chapter 4, Sections 4.2.1 and 4.3.1) for the low and high pressure CO₂ corrosion datasets respectively. Figures 4-1 and 4-15 are box plots of the data summarised in Tables 4-1 and 4-5, for low and high pressure CO₂ corrosion respectively. Also, the test set was summarised in Tables 4-2 and 4-6 for low and high pressure CO₂ data respectively, for prediction and the subsequent calculation of the R^2 -coefficient.

Classical logic is based on the assumption of bivalence; that there are two truth-values (true and false) and that the value of any given logical formula is uniquely defined by the truth-values of its components: bivalence and truth functionality respectively (Behlolvek and Klir, 2011). Various many-value logics employ the use of truth functionality while abandoning the bivalence concept (Behlolvek and Klir, 2011).

Classic logic and predicates of classic logic are linked closely with classic sets. A predicate is a system that contains formulae and quantifiable variables. A logic operation on a predicate produces a unique solution – an operation on a classical set.

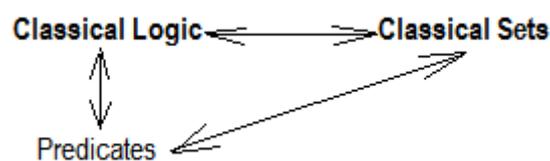


Figure 6-1. Close connection between classical logic and classical set and their relationships with predicates of logic

However with the abandoning of bivalence at the root core of many-value logics, the connection between logic and sets was severed. Historically, fuzzy logic was created by Lotfi Zadeh in the 1960s (Ukil, 2010). Lotfi Zadeh renewed the connection between many-value logic and classical sets in his 1965 publication on fuzzy sets (Zadeh, 1965). A subsequent

publication was made ten years after the first seminal paper was released, in which the term fuzzy logic, was used (Zadeh, 1975).

Fuzzy logic systems are an extension of multi-valued logic with the aim of developing approximate reasoning that is closer in spirit to human thinking and natural language than traditional logic systems: (Hajizadeh, 2006; Ukil, 2010; Fuzzy Logic Toolbox User's Guide, 2015).

6.2 Definition of Fuzzy Logic

Fuzzy logic is said to have two different meanings – narrow and broad definitions. The narrow definition is that fuzzy logic is a logical system, which is an extension of multivalued logic or probabilistic logic. It deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values), fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. The truth value for any given variable in a fuzzy set is determined by a membership function as shown in the following figure.

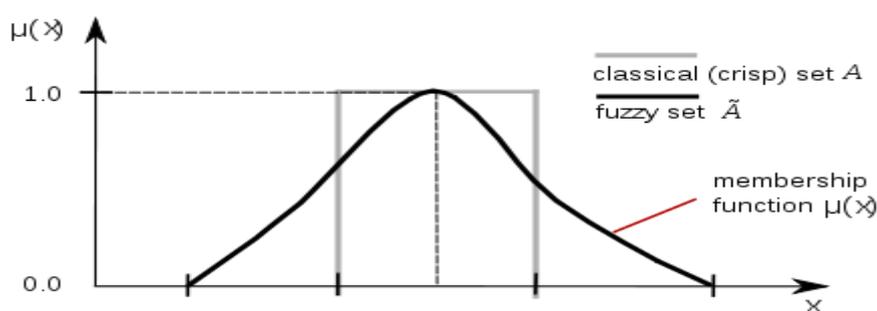


Figure 6-2. Membership function of a fuzzy set

For any set X , the membership function on X , $\mu(X)$, is a function from X to the real unit interval $[0,1]$ (Zadeh, 1965). Membership functions will be discussed further in the next section. The broad definition of fuzzy logic is almost synonymous with set theory, to which fuzzy sets belong. It is a branch of set theory that deals with the impreciseness of certain phenomena such as classes of objects with unsharp boundaries in which membership is a matter of degree. It is thus related to both logic as well as set theory (Behlolavek and Klir, 2011).

6.3 Membership Functions

A membership function is a curve that defines how each input variable is mapped into a membership value (or degree of membership) between 1 and 0 such that the membership interval range is [0,1], as stated earlier (Zadeh, 1965; Hajizadeh, 2007).

The membership degree $\mu_A(X)$ quantifies the grade of membership of the element x to the fuzzy set A . The simplest membership functions consist of straight lines and the most basic is the triangular membership function as shown in Figure 6-3A. This function is essentially a collection of three points forming a triangle. The other straight-line function, the trapezoidal membership function, which has a flat top and two sloped sides, is shown in Figure 6-3B .

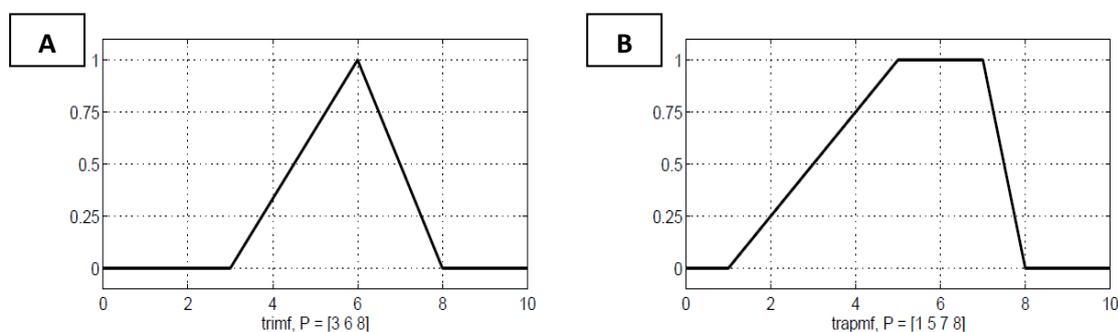


Figure 6-3. Triangular and trapezoidal membership functions

(Fuzzy Logic Toolbox User's Guide, 2015)

Piece-wise membership functions portray asymmetry in membership classification. For instance, the piece-wise membership function shown in Figure 6-4, has a point of discontinuity in the vertical region hence it groups data distinctly into two separate categories in terms of degree of membership (Fuzzy Logic Toolbox User's Guide, 2015).

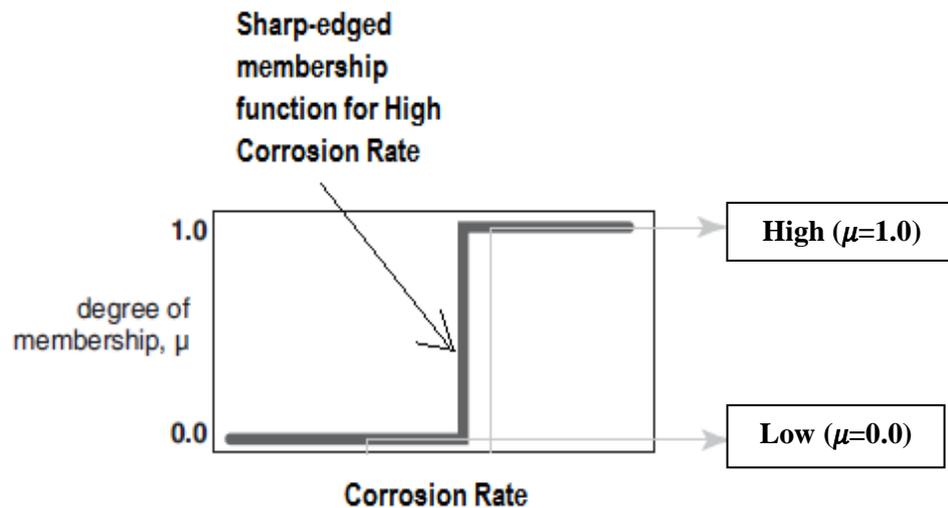


Figure 6-4. Linear piece-wise membership function (Ukil, 2010)

In general, the more complex membership functions such as the Gaussian, generalised bell, sigmoidal, polynomial and the Z, S and Pi curves exhibit a continuous form as shown in Figure 6-5. However, it must be noted that of all the continuous membership functional forms, particularly the sigmoidal and bell-shaped functions have relatively little practical use in fuzzy control (Ukil, 2010).

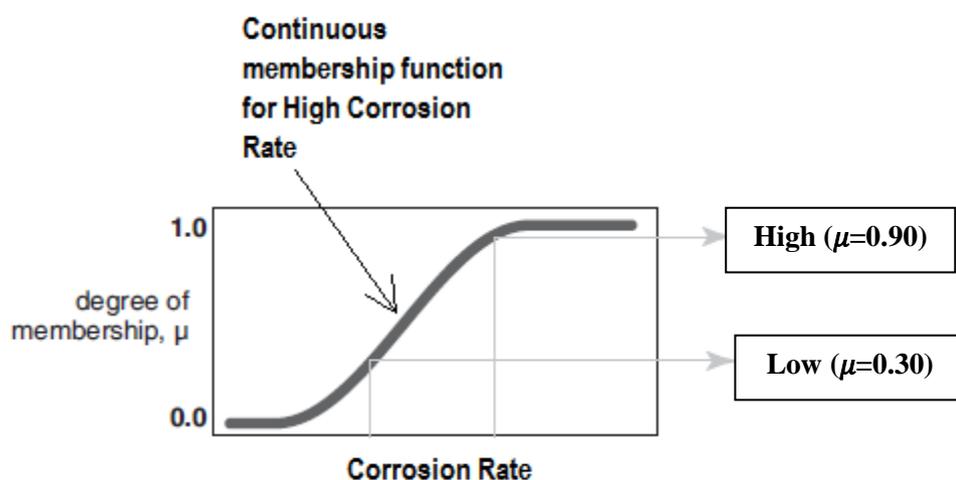


Figure 6-5. Generalised form of a continuous membership function

(Fuzzy Logic Toolbox User's Guide, 2015)

6.4 Mamdani and Sugeno Fuzzy Inference Systems

The Mamdani fuzzy inference system (FIS) is a widely accepted system for capturing expertise knowledge while allowing the ability to describe the expertise in a more intuitive and human-like manner.

The Mamdani FIS was proposed by Ebrahim Mamdani in 1975 following Lotfi Zadeh's 1973 publication, as an attempt to control a steam engine and boiler configuration by the implementation of a set of linguistic control rules obtained from experienced, well-trained human operators (Mamdani and Assilian, 1975; Mamdani, 1976; Mamdani, 1977).

In general, both Mamdani and Sugeno fuzzy inference systems involve the transformation or fuzzification of input variables from a given data set into fuzzy sets, categorising the data into degrees of membership (by membership function) and the consequent re-transformation of the fuzzy sets into desirable output (Fuzzy Logic Toolbox User's Guide, 2015). Where the main difference lies between these two methodologies, is in the re-transformation of fuzzified data into desirable output. While for the Mamdani FIS, a defuzzification technique is used, for the Sugeno FIS, a weighted-average technique is implemented to compute the output (Fuzzy Logic Toolbox User's Guide, 2015). Also, crisp functions are used as the consequences for rules in the Sugeno FIS while the Mamdani FIS is not characterised by this approach in defining rules (Ukil, 2010). Both methodologies can be used to effectively model any system however the Mamdani FIS is more versatile as it does not matter what form the desired output takes whereas the Sugeno FIS is only capable of rendering a constant or linear output.

6.5 Low Pressure CO₂ Corrosion Prediction using a Fuzzy Inference System

6.5.1 Methodology

The Mamdani fuzzy inference system (FIS) was used in the development of the Mamdani model based on the given low pressure CO₂ datasets: (Dugstad et al., 1994b; Nordsveen et al., 2003). The model's performance was then tested using the test sample dataset. The Mamdani FIS processes and its corresponding methods are listed in Table 6-1. These FIS properties produce the most desirable output for CO₂ corrosion modelling in Matlab.

Table 6-1. Summary of FIS processes and methods

FIS Process	Mamdani Method
Fuzzy Operator Application (And)	Prod
Fuzzy Operator Application (Or)	Max
Implication	Min
Aggregation	Max
Defuzzification	Centroid

In general, FIS processes comprise fuzzification, fuzzy operator application, implication, aggregation and defuzzification stages, as shown in numbered steps 1-5 in Figure 6-5. Fuzzification is the process by which a linguistic set or descriptor is defined for the initial input data (Mamdani and Assilian, 1975).

Fuzzy operators: 'And' and 'Or', as listed in Table 6-1, are then applied to the fuzzified data, thus converting the data into an antecedent. Once the fuzzy rule weights are set, implication involves taking a single number from the antecedent as input and forming a fuzzy set as output, the consequent. The consequent is represented by a membership function, which weights appropriately the linguistic features that are attributed to it. Implication is implemented for each fuzzy rule. The 'Min' or minimum implication method tends to truncate the output fuzzy set as opposed to the 'Prod' or product implication method which scales the output fuzzy set. Also, the centroid defuzzification method was chosen over the bisector method for its suitability (See Appendix A6.1 a and A6.1b).

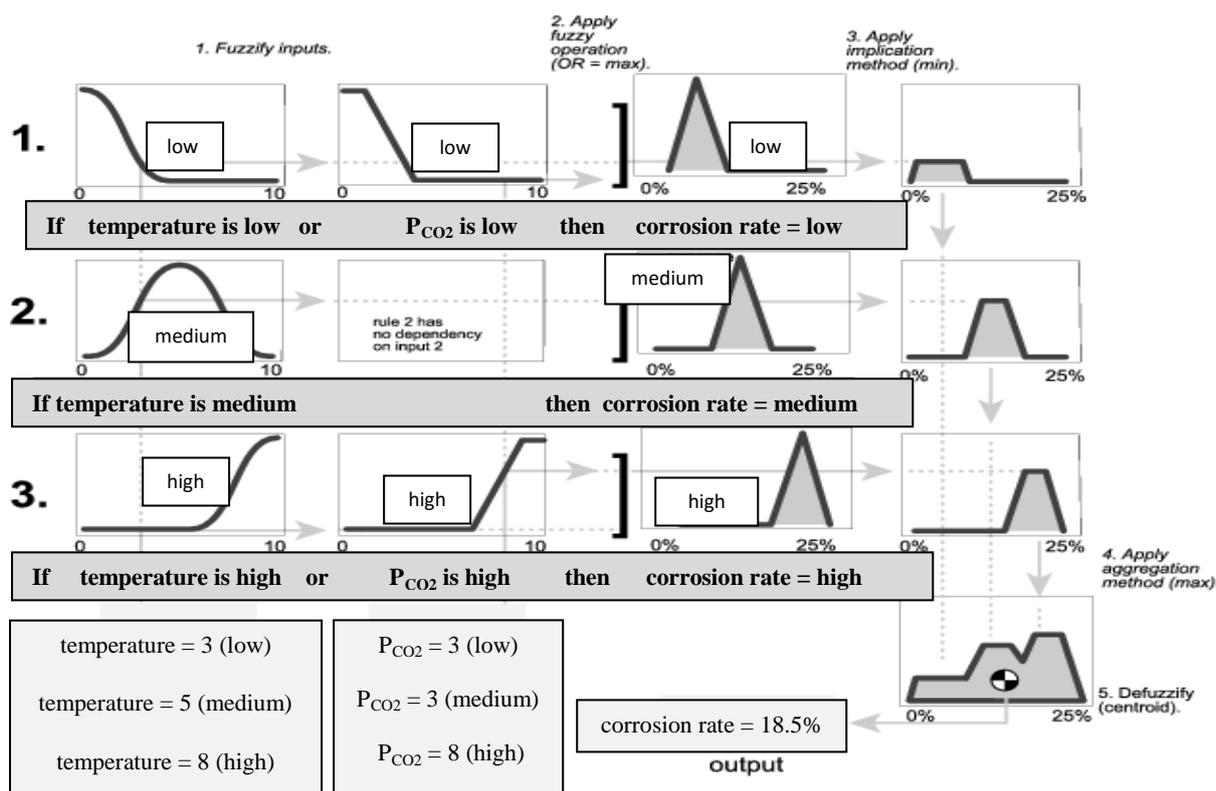


Figure 6-6. Fuzzy inference diagram showing all parts of the fuzzy inference process – from fuzzification through to defuzzification (Fuzzy Logic Toolbox User’s Guide, 2015)

Aggregation is the process by which the fuzzy sets that represent the output of each rule are combined to form an aggregate fuzzy set. The input to aggregation, is the set of fuzzy sets for each rule, defined by the implication process. The aggregate output is such that there is one fuzzy set to each input variable.

Defuzzification follows aggregation, thus receiving the input of aggregation, a single fuzzy set and then returns a single number, that best represents it. There are five different options for Mamdani FIS defuzzification – centroid, bisector, largest of maximum (lom), smallest of maximum (som) and middle of maximum (mom). The centroid was selected because it returns the centre of the area under the curve, thus producing the most desirable corrosion rate plot shape. It is important to note that while the fuzzification to defuzzification steps shown in Figure 6-6 are slightly different to the approach used by Mamdani, the overall idea is identical (Matlab Fuzzy Logic User’s Guide, 2015). Also, the Mamdani FIS is based on the original idea by Zadeh (Zadeh, 1973). Each of the input variables was assigned three

membership functions – Low, Medium, and High. For instance, the pH variable is shown in Figure 6-7.

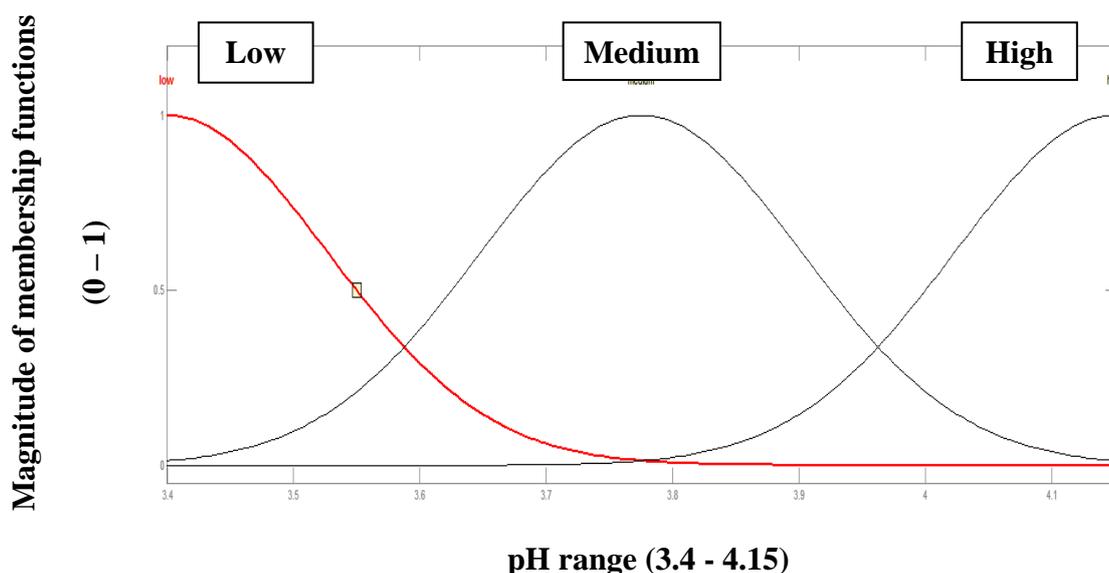


Figure 6-7. pH membership function plots

The output variable, corrosion rate is assigned seven membership functions, namely – Very low, Low, Low-Medium, Medium, Medium-High, High and Very High. Seven membership functions are used in order to define an entire range of corrosion rate magnitudes (0-70 mm/year) as seen in Table 6-2.

Table 6-2. Membership function descriptor and corresponding corrosion rate ranges

Membership Function Descriptor	Corrosion Rate Range (mm/year)
Very Low	0-10
Low	10-20
Low-Medium	20-30
Medium	30-40
Medium—High	40-50
High	50-60
Very High	60-70

In order to develop the FIS for the dataset, fourteen rules are defined. The following Table shows some of the rules of the FIS based on close observation of variable trends within the dataset.

Table 6-3. Some rules for the development of the Mamdani and Sugeno FIS

Rule Number	Temperature	P_{CO_2}	Velocity	pH	Corrosion Rate	Weight
1	Medium	High	High	Low	Very High	1
2	Medium	Low	High	Medium	Low-Medium	1
3	*Not Medium	*Not High	Low	Low	Very Low	1
4	Medium	Low	High	Low	Medium	0.1
5	High	High	High	Low	Medium-High	0.1
6	High	High	Medium	Low	High	0.01
7	Low	Low	High	Medium	Low	0.001

* Implies that these descriptors are negations

The rule viewer is set once these rules are established in the FIS. Three-dimensional (3-D) plots of each of the variables in the dataset permit observation of any two input variables with CO_2 corrosion rate. One such plot is shown in Figure 6-8.

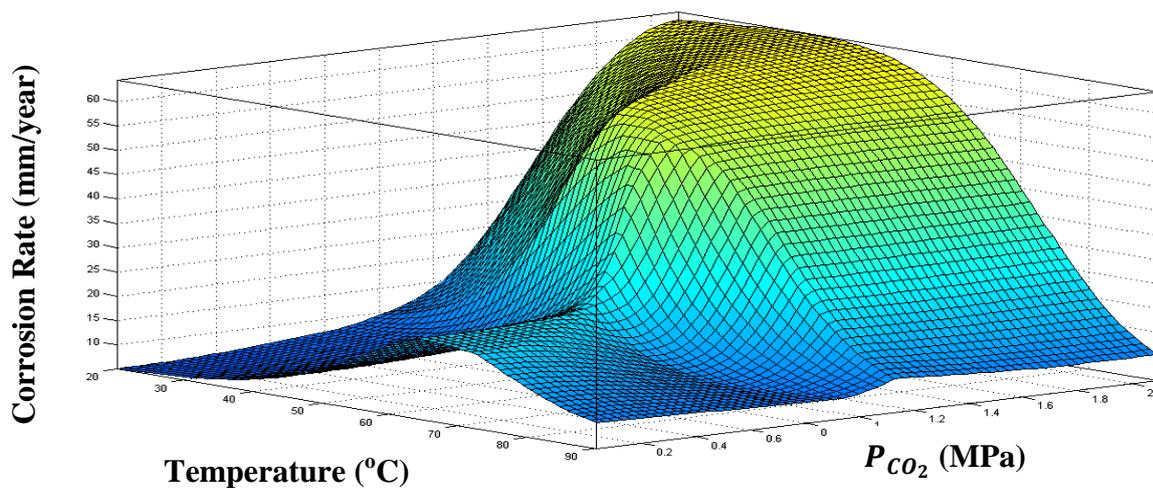


Figure 6-8. Three-dimensional plot of corrosion rate against P_{CO_2} and temperature – result of Mamdani FIS

The 3-D plot in Figure 6-8 shows a dome-like shape for the variation of CO_2 corrosion rate with temperature and CO_2 partial pressure. There is a general increase of corrosion rate with increasing temperature and P_{CO_2} , however while there is a distinct peak for the change in

corrosion rate with temperature as shown in the two-dimensional (2-D) plot in Figure 6-9, the corrosion rate appears to level off for very high values of P_{CO_2} , as shown in Figure 6-10.

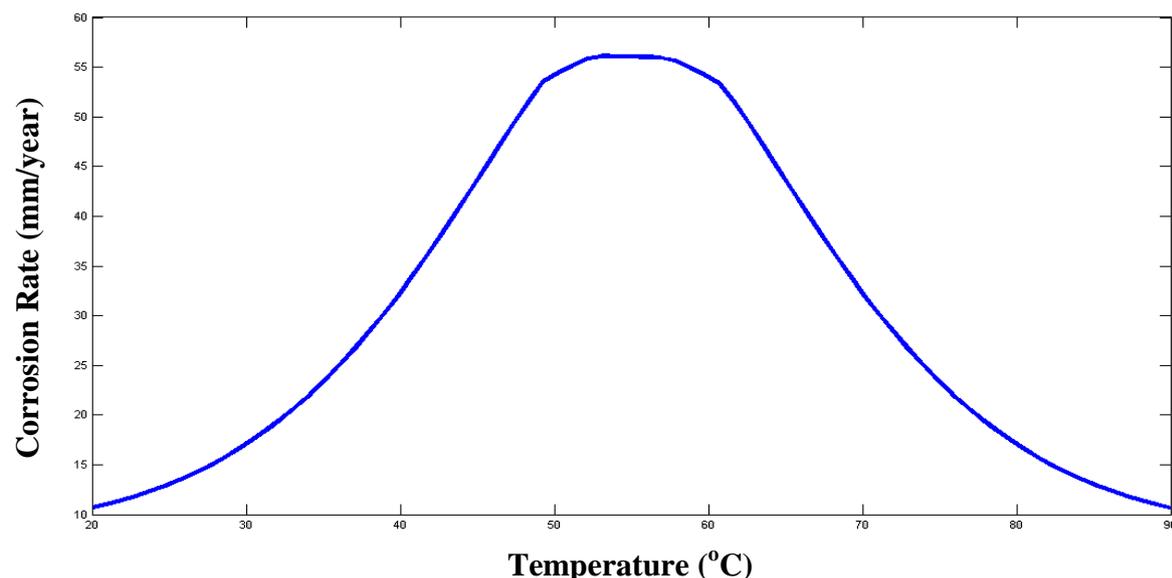


Figure 6-9. Mamdani FIS plot of corrosion rate against temperature

The corrosion rate plot against temperature shows a peak at a temperature of 55°C. The corrosion rate value at the peak is 57mm/year and as temperature increases further to 90°C, corrosion rate decreases steadily. Thus the plot in Figure 6-9 follows the typical corrosion rate against temperature plots described in research literature, for scaled corrosion (De Waard and Lotz, 1993; Zhang et al., 2012b). For this particular data, the greatest magnitudes for corrosion rate are at temperatures of 60°C, while there are lower corrosion rates at 90°C. The main shortcoming of this plot is that corrosion rates are rather conservative and it takes into account, the very high corrosion rates within the dataset. This plot does not represent the mean corrosion rates, particularly at 60°C and at 90°C.

For the variation of corrosion rate with P_{CO_2} , Figure 6-10, corrosion rates increase very steadily from 15mm/year, for CO_2 partial pressures in the range, 0-0.7MPa, to 17mm/year, after which it increases steeply to 60mm/year for P_{CO_2} values of about 1.2MPa. Beyond this point, corrosion rates plateau and this depicts the dataset accurately. Theoretically, corrosion rates are expected to increase with increasing P_{CO_2} . In fact corrosion rates increase with increasing P_{CO_2} to an exponent of 0.7, as seen in Figure 2-7A, Section 2.4.1 (Dugstad et al., 1994b). The reason corrosion rate does not carry on increasing throughout the range of P_{CO_2}

values is due to the fact that film formation on the metal surface limits CO_2 corrosion rates, by reducing the surface area of the metal exposed to the acidic medium.

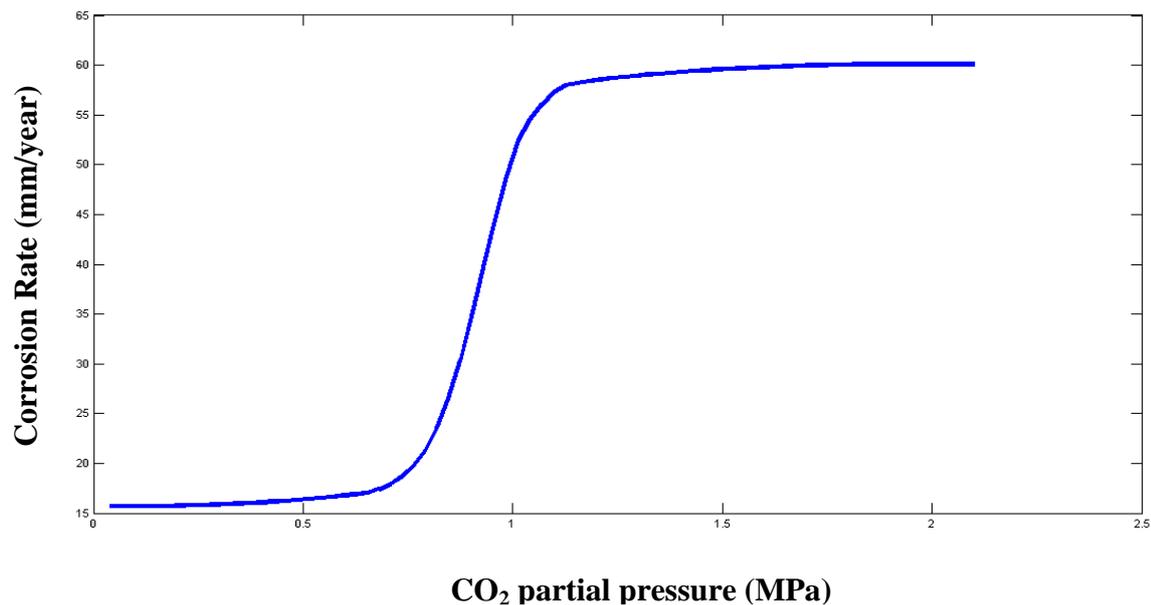


Figure 6-10. Mamdani FIS plot of corrosion rate against CO_2 partial pressure

Although, for this particular study, the water is un-buffered, implying a more corrosive medium, the protection provided by the surface film is sufficient to reduce corrosion rates at very high P_{CO_2} values. In un-buffered water, the film is said to comprise ferrous carbides and alloying elements from the steel (Dugstad et al., 1994b).

The 3-D plot of Figure 6-11, shows that there are corrosion rate peak values for both temperature and velocity due to the presence of the central crest of the dome. The 2-D plot of corrosion rate against flowrate velocity (Figure 6-12) shows that the peak occurs at (7m/s, 57mm/year). An increase in flowrate velocities results in a fairly steep elevation of corrosion rates. This is attributed to the erosional effect of fluid flow on surface film growth and thickness. The greater the velocity of fluid flow, the greater the tendency for any surface film to be washed away, exposing a greater area of the metal surface to the corrosive medium and increasing corrosion rates (Dugstad et al., 1994b).

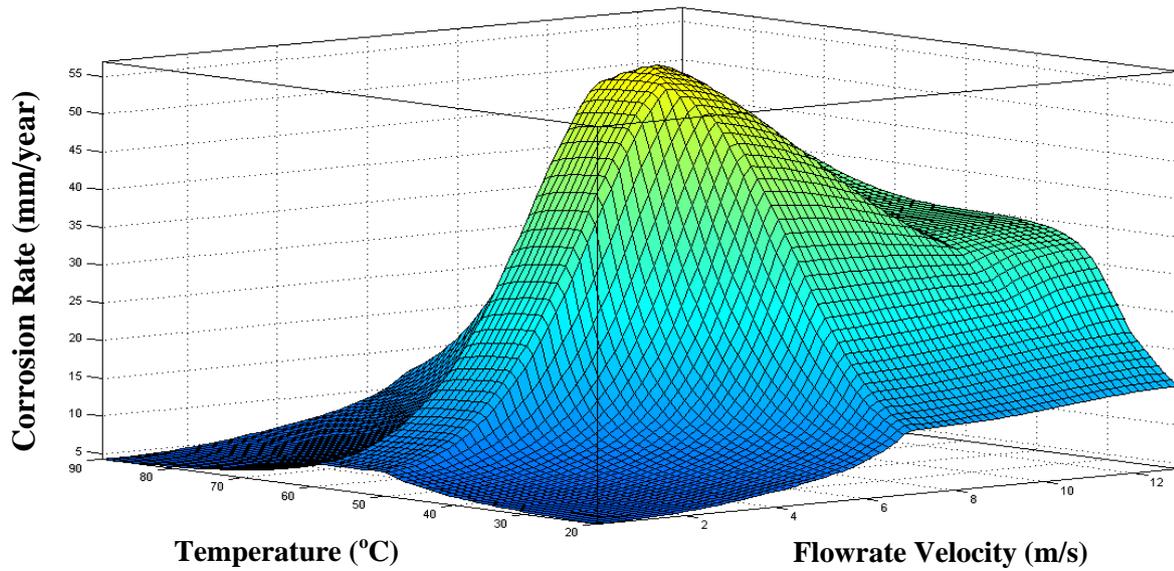


Figure 6-11. Three-dimensional plot of corrosion rate against flowrate velocity and temperature – result of Mamdani FIS

However, corrosion rates reach a peak at (7m/s, 57mm/year) and level-off at (13m/s, 35mm/year). The higher fluid velocities wash away the surface film and at the same preventing the medium from further inducing the dissolution of iron to ferrous ions in solution at a sufficiently fast rate.

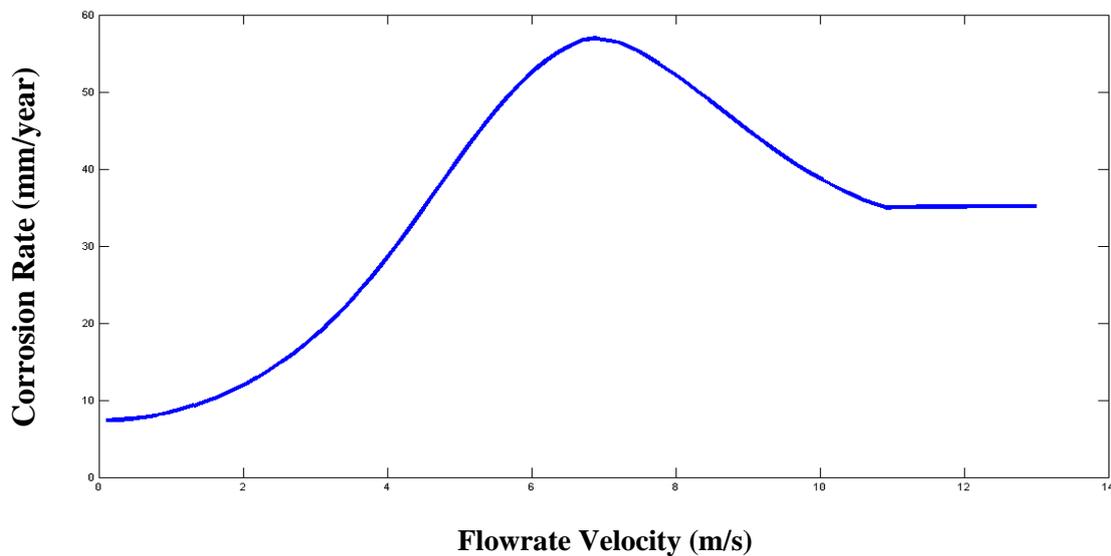


Figure 6-12. Mamdani FIS plot of corrosion rate against flowrate velocity

The overall effect is that corrosion rates slow down and then decrease as the fluid velocity reaches the 13m/s mark. In general, it is difficult to model CO₂ corrosion rate based on established fluid mechanical theories of mass transfer and shear stress for this particular study because of the effect that flow velocities have on the composition and the thickness of the surface film layer (Dugstad et al., 1994b).

Figure 6-13 shows the prediction results from the Mamdani FIS model using the test dataset. The R^2 -value is relatively low, with about half of the data-points lying outside of the 95% confidence interval. The implication is that the model is not the most accurate of the all the discussed CO₂ prediction models and in terms of reliability, a greater number of high-magnitude corrosion rates fall within the boundaries of the confidence interval than the low-magnitude corrosion rates. Hence it is more accurate and reliable in predicting high-magnitude corrosion rates and much less accurate in predicting low-magnitude corrosion rates.

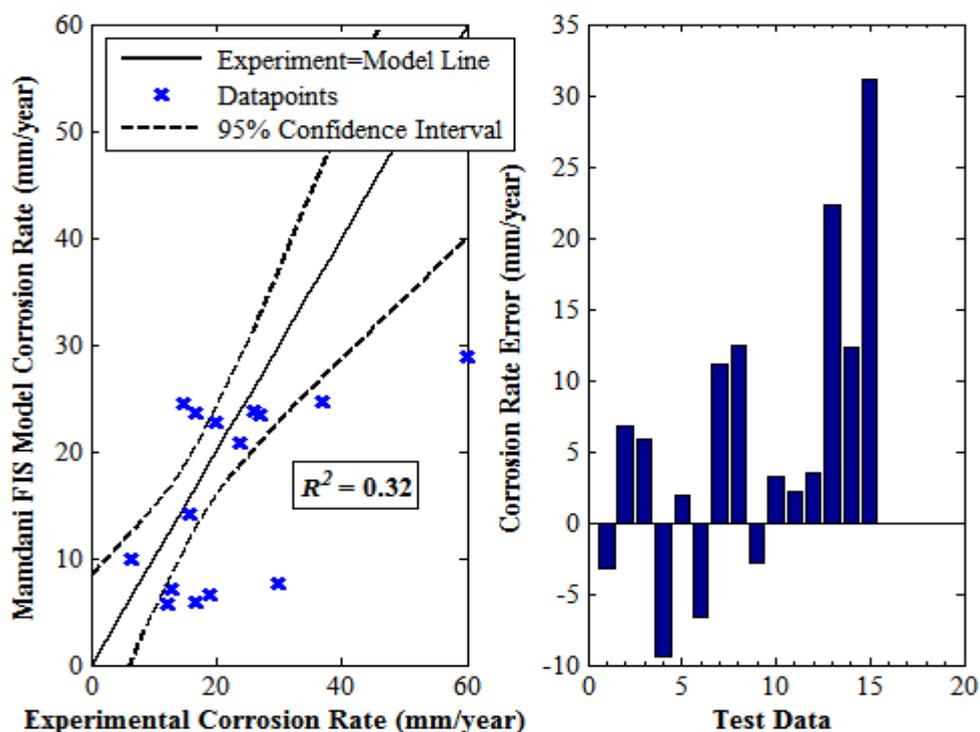


Figure 6-13. Mamdani FIS model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate

6.6 High Pressure CO₂ Corrosion Prediction using a Fuzzy Inference System

The data used for the fuzzy inference analysis in this study was obtained from the following research publications: Choi and Nesic (2009), Choi et al (2010), Hesjevik et al (2010) and Zhang et al (2012a).

6.6.1 Methodology

Mamdani and Sugeno fuzzy inference systems were implemented using the Matlab fuzzy toolbox. The variables used for the analyses are listed in Table 6-4, below:

Table 6-4. Variables used in the study and their corresponding ranges

Variable	Range
Temperature (°C)	0-150
P_{CO_2} (MPa)	3.5-23.3
pH	3.1-6
Fluid flowrate (m/s)	0-4
Corrosion Rate (mm/year)	0-15

The data used in the analyses was gathered from the aforementioned sources and sorted as listed in Table 6-4. A diagrammatic representation of the FIS is shown in Figure 6-14, below.

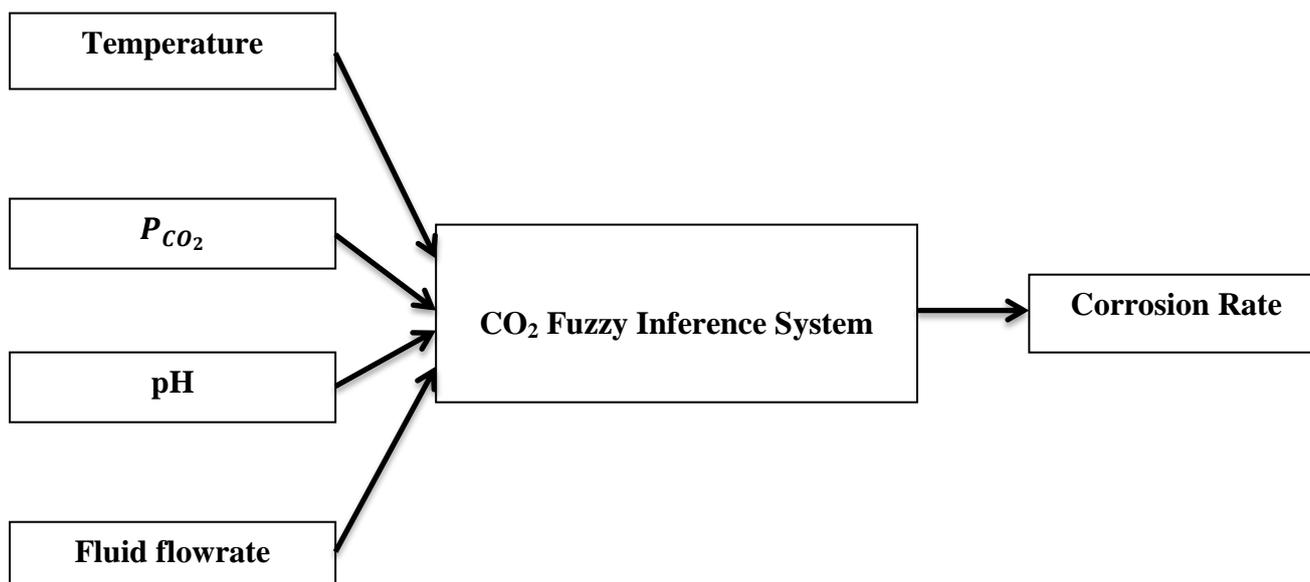


Figure 6-14. A generic fuzzy inference system showing input variables to the left (T , P_{CO_2} , pH, fluid flowrate) and the output variable to the right (Corrosion Rate)

Three membership functions are assigned to each input variable and five membership functions are assigned to the output variable. Descriptors are used in naming the membership functions. The input variable membership functions are: low, medium and high while the output variable membership functions are: very low, low, medium, high and very high as shown in Table 6-5. The membership function type used for both input and output variables is the Gaussian function. This function type was chosen for its practicality and also, for the fact in general, by the central limit theorem, the arithmetic means of samples from randomly different data sets are likely to be normally distributed (Rice, 2007).

Table 6-5. Descriptors assigned to each of the input and output variable membership functions

Input variable membership function	Output variable membership function
Low	Very low
Medium	Low
High	Medium
	High
	Very High

In order to obtain an output from an FIS, a number of rules and the respective weight contribution for each rule has to be developed. The following table shows the seven self-developed rules used by this particular FIS.

Table 6-6. Rules for the development of Mamdani and Sugeno FIS

Rule Number	Temperature	P_{CO_2}	pH	Fluid flowrate	Corrosion Rate	Weight
1	Medium	High	Low	High	Very High	0.4
2	High	Medium	High	Low	Low	0.1
3	Low	Low	High	Low	Very low	0.2
4	Medium	High	Low	High	High	0.1
5	High	Medium	Medium	Medium	Medium	0.1
6	High	*	Low	Low	Very low	0.05
7	*	High	Low	Low	Very low	0.05

* Implies that these variables are absent in the given rule

Upon creation of the rules, the rule viewer is set as shown in Figure 6-15. The rule viewer provides the facility to change the magnitude of the input variables while showing an instantaneous change in the output (high pressure CO₂ corrosion rate).

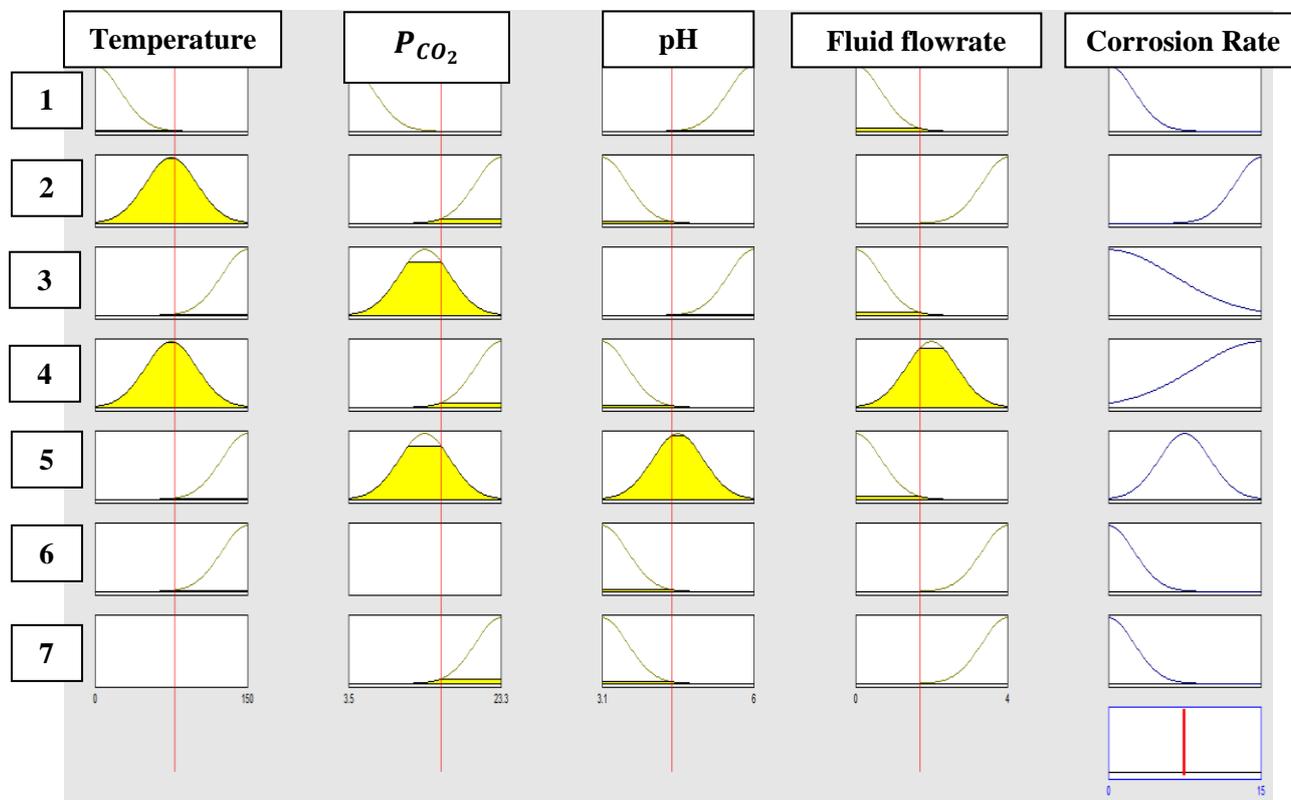


Figure 6-15. Rule viewer for all seven rules showing mini graphical representations of each of the input variables and their effect on the output variable

The following Table shows the magnitudes of the variables used in producing the Mamdani and Sugeno FIS corrosion plots shown in Figures 6-16 and 6-17, respectively.

Table 6-7. Magnitudes of input variables and respective output variable results

Input Variable	Output Variable
Temperature = 78.4°C	Mamdani FIS corrosion rate = 7.5mm/year Sugeno FIS corrosion rate = 5.43mm/year
P_{CO_2} = 15.5MPa	
pH = 4.44	
Fluid flowrate = 1.69m/s	

Three-dimensional plots of corrosion rate against P_{CO_2} and temperature are shown for both Mandani and Sugeno systems in Figures 6-16 and 6-17.

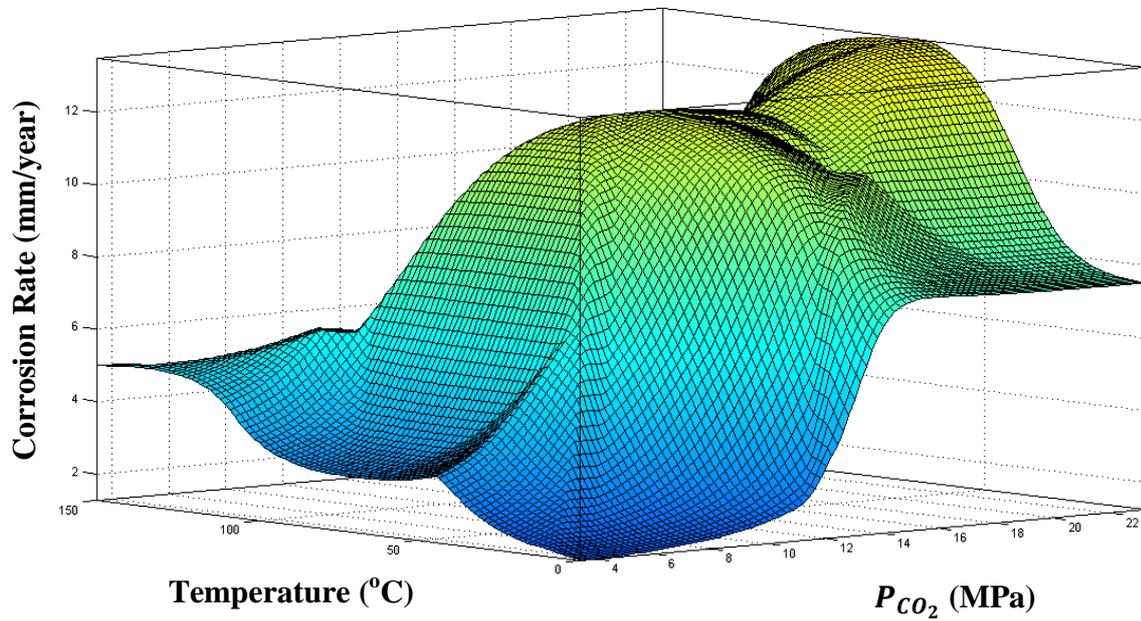


Figure 6-16. Three-dimensional plot of corrosion rate against P_{CO_2} and temperature – result of Mamdani FIS

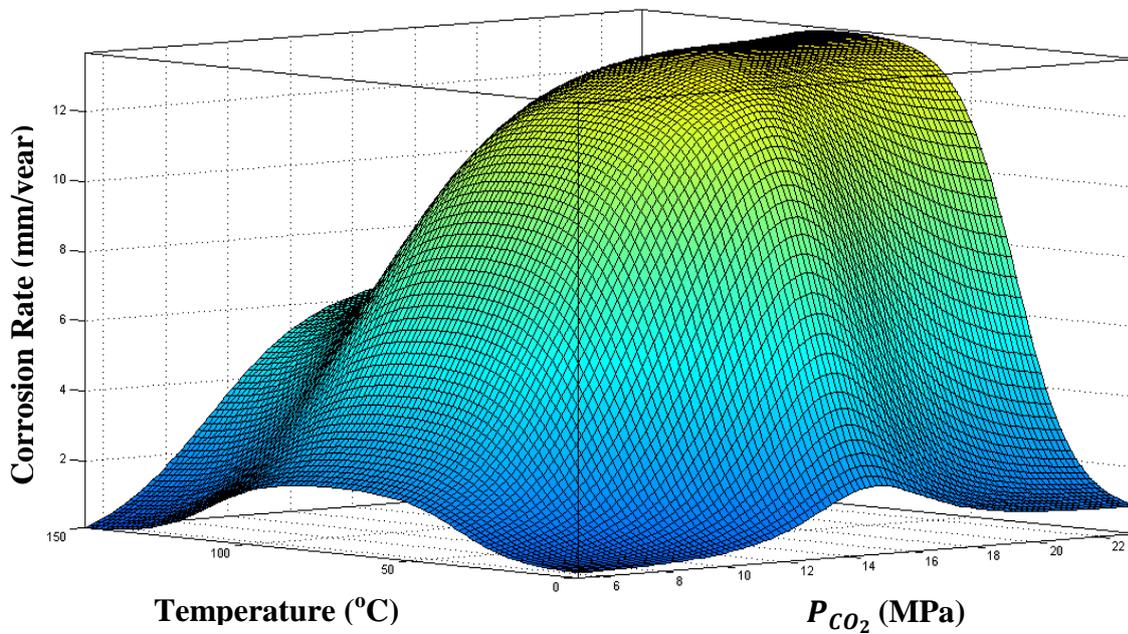


Figure 6-17. Three-dimensional plot of corrosion rate against P_{CO_2} and temperature – result of Sugeno FIS

Table 6-7 is a summary of results from the rule viewer for Figure 6-15. The Mamdani FIS plot shown in Figure 6-16, illustrates the variation of corrosion rate with P_{CO_2} and temperature in three-dimensional space. It can be seen that for the temperature variable, corrosion rates increase steadily and peaks at approximately 80°C and 12 mm/year at the centre of the dome in accordance with the Van't Hoff, Arrhenius and Kinetic theories (Tan and Chan, 2011). Corrosion rates then decrease steeply as temperatures drop (See Appendix A6.2).

For CO_2 partial pressures, the two-dimensional Mamdani plot, shown in Figure 6-18, is derived from the three-dimensional plot in Figure 6-16. It is observed that the corrosion rate increases steadily with increasing P_{CO_2} up till pressures of approximately 12MPa after which there is a slight dip before it reaches the point, (~23MPa, 13.5mm/year): this is the highest corrosion rate.

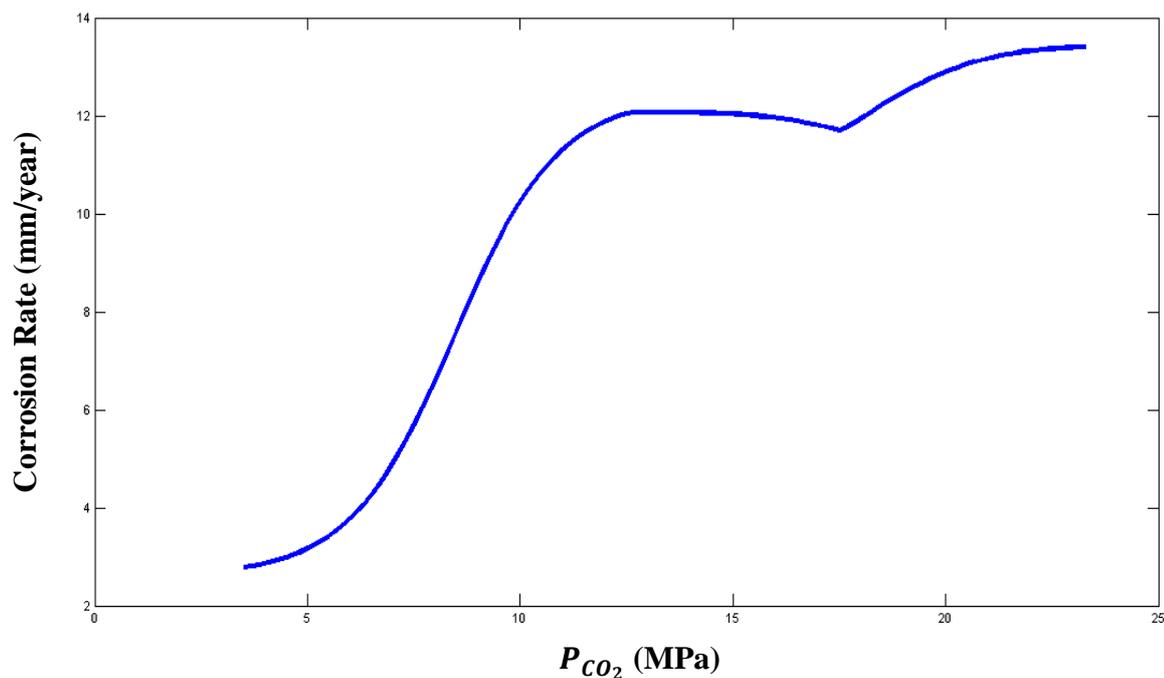


Figure 6-18. Mamdani FIS plot of corrosion rate against CO_2 partial pressure

Higher corrosion rates are possible for high pressure CO_2 corrosion than for low pressure CO_2 corrosion (Zhang et al., 2012b). However, there is a tendency for scale formation to limit corrosion rates as scales are found to be thicker. However for this plot, the velocity is high

(4m/s) and the pH is low (3.5), hence corrosion rates continue to increase with increasing CO_2 partial pressures.

The Sugeno FIS plot shown in Figure 6-17, illustrates the variation of corrosion rate with P_{CO_2} and temperature in three dimensions. The plot trend for the temperature axis of Figure 6-17, shows an initial rise and subsequent plateau in corrosion rates hence exhibiting an identical pattern to the front part of Mamdani plot dome. The variation of corrosion rates with P_{CO_2} is only slightly different to the observed trend exhibited by the Mamdani plot. Corrosion rates increase with increasing CO_2 and reaches a peak at ≈ 13.8 MPa and plateaus for increasing P_{CO_2} values as seen in Figure 6-19.

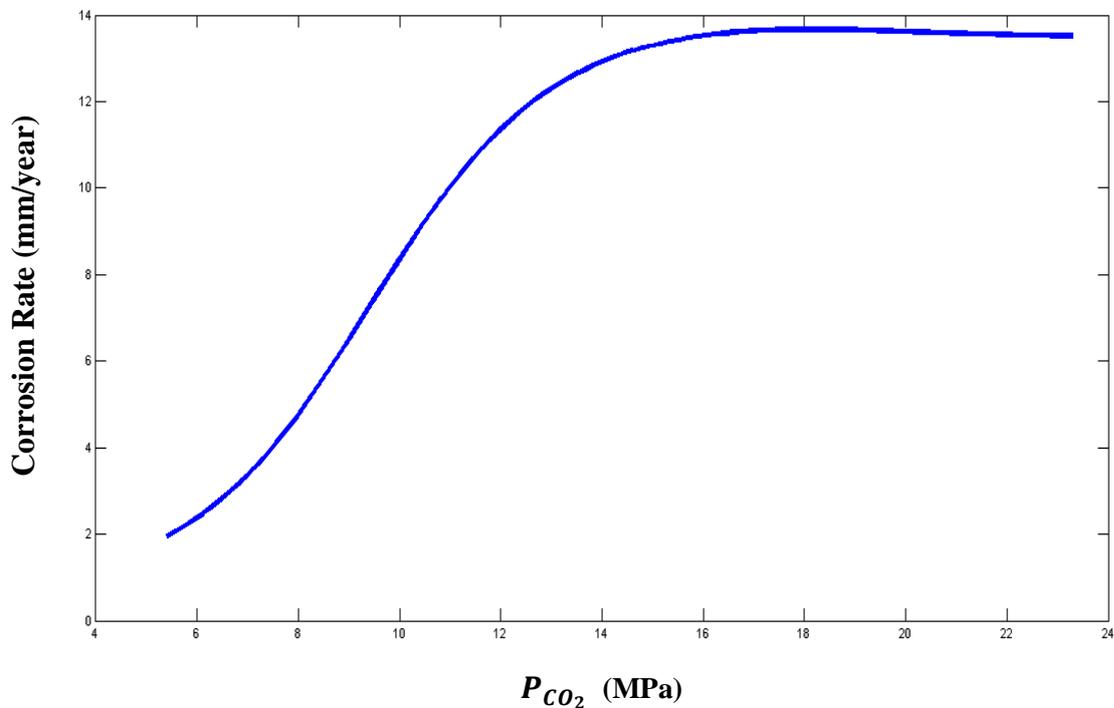


Figure 6-19. Sugeno FIS plot of corrosion rate against CO_2 partial pressure

A peak is reached in CO_2 corrosion rates because super-saturation of the aqueous medium at the metal surface would ensue. Super-saturation in this context implies that the concentration of ferrous ions (Fe^{2+}) and carbonate ions (CO_3^{2-}) have exceeded their solubility limits in the aqueous medium or fluid hence film growth occurs on the metal surface. For high pressure CO_2 corrosion, thick scale formation is favoured (Zhang et al., 2012b) and this serves to limit corrosion rates, even with increasing CO_2 partial pressures. Thick scale covers the metal

surface and reduces the rate of iron dissolution, moreover the aqueous medium is super-saturated thus cannot accommodate additional Fe^{2+} ions which ultimately results in limiting the magnitude of CO_2 corrosion rate. Figure 6-20 shows the model vs experimental plot and error bar chart for the Mamdani FIS. The model has a tendency for under-prediction. The 95% confidence bounds indicate that the reliability in prediction is not very high but this is to be expected since the size of the dataset is small.

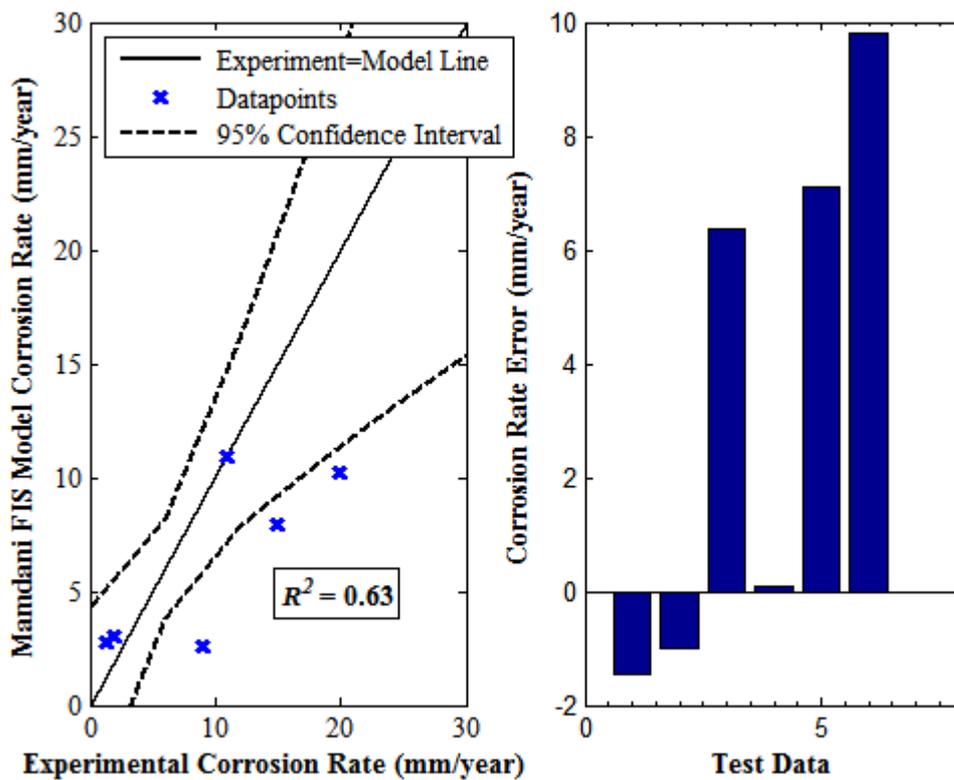


Figure 6-20. Mamdani FIS model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – Error is the difference between experimental and model corrosion rate

6.7 Conclusions

Classical logic comprises two parts: bivalence and truth functionality. Bivalence is the concept of having two truth value (true or false) outcomes while truth functionality is the use of a truth-value to represent the solution of a given logical formula. The link between classical logic and classical sets was established through the application of bivalence.

By abandoning bivalence, these two systems were disconnected from each other only to be re-connected once more by Lotfi Zadeh in the 1960's. The development of fuzzy inference systems has led to intelligent systems which use human thinking and natural language to transform input data to output data.

For modelling low and high pressure CO₂ corrosion, each of the input variables, namely: temperature CO₂ partial pressure, flow velocity, pH as well as the output variable, corrosion rate were simulated using the Gaussian membership function or bell functions. The reason that these were chosen is that they are continuous membership functions hence they depict smoother and more rounded edges during profile prediction than the straight-lined or sharp-edged results of the triangular, trapezoidal or linear piece-wise functions.

Additionally, suitable descriptors such as very low, low, medium, high and very high were used with corresponding ranges such as 0-10mm/year to categorise corrosion rates into different classes in order to aid corrosion simulation. Also, the rules were developed separately for the low and high pressure CO₂ datasets so that the inherent traits within each dataset is fully captured during simulation.

Fuzzy systems in general, map input data to output data via a procedural process: input data → fuzzification → implementation → aggregation → defuzzification → output data. For the fuzzification stage, which is the assigning of linguistic descriptors to the initial input data, input is converted to the antecedent. The 'Prod' and 'Max' options were the selected fuzzy operators for 'And' and 'Or' respectively because the alternative fuzzy operator options, the 'Min' and 'Max' produced simulated corrosion rate magnitudes that did not match-up against expected results and so were unsuitable.

For the implication stage, which involves taking a single number from the antecedent as input and forming a consequent by representing it with a membership function, the 'Min' option was selected for its suitability. The 'Prod' option tended to scale the simulated results by a small factor. For the aggregation stage, which is the combining of all fuzzy sets into a single

aggregate fuzzy set, ‘Max’ was chosen due to the apparent likeness of simulated results to the experimentally measured data of the given datasets.

For the defuzzification stage, which is the re-transformation of fuzzified data into output data, the centroid was selected out of a possible five alternatives because it returns the centre of the area under the curve thus producing the most desirable results, particularly when simulated trends for the various input variables are to be assessed against expected profiles.

These FIS processes were applied to both the low and high pressure CO₂ datasets following testing and trial runs of the various other alternatives. The Mamdani system was the FIS method deployed for both datasets. The Sugeno method was also applied to the high pressure CO₂ dataset however the R^2 -value was considerably low.

In terms of performance, the Mamdani FIS model attained a higher level of accuracy for the high pressure CO₂ dataset than for the low pressure CO₂ dataset as shown in Table 6-8.

Table 6-8. Model performance for each dataset

Mamdani FIS Model	R^2-value
Low Pressure CO ₂ dataset	0.32
High Pressure CO ₂ dataset	0.63

This may be due to the fact that though the size of the dataset for the high pressure CO₂ case is considerably smaller when compared to that of the low pressure CO₂ dataset, there is also much less variation within it. There is also a substantially less scatter of data-points in the model vs experimental plots for the high pressure CO₂ dataset when compared to that of the low pressure CO₂ one.

Overall, the FIS model is not the most accurate prediction system there is for CO₂ corrosion, but it excels at showing realistic relationships between the various variables and the corrosion rate. The simulated corrosion rate against temperature, P_{CO_2} and flow velocity plots depict the expected trend that is characteristic of experimental profiles.

Chapter 7. Monte Carlo Simulation

7.1 Introduction

Monte Carlo simulation is a computational technique for determining the sensitivity of a complex system by varying parameters within statistical constraints (Statistics and Machine Learning Toolbox User's Guide, 2015). It can also be defined as a numerical method of solving mathematical problems by random sampling (Sobol, 1994). Historically, the birth of this technique is widely accepted to be in 1949 and its main originators are: Stanislaw Ulam and John von Neumann (Sobol, 1994). Interestingly however, the theoretical foundation behind the technique which involves the use of randomness in a determinative manner can be traced back to the 18th century (Harrison, 2010).

Sections 4.2.1 and 4.2.3 covered the descriptive statistics and regression analysis for the low pressure CO₂ dataset respectively, while Sections 4.3.1 and 4.3.3 of the previous chapter covered the descriptive statistics and regression analysis for the high pressure CO₂ dataset respectively. In the previous chapter, a fuzzy model was derived with some limitations particularly in the prediction accuracy for low pressure CO₂ data.

In this chapter, the focus is a mathematical model that involves the prediction of CO₂ corrosion rate from the following input variables: temperature, CO₂ partial pressure (P_{CO_2}), pH and flow velocity. In order to carry out Monte Carlo simulations, appropriate probability distributions for the given datasets must be assumed or obtained in conjunction with a suitable best-fit regression equation, which in this thesis, was derived from the chapter on statistical analysis.

For each of the low and high pressure CO₂ corrosion datasets, the multiple linear regression equations were deemed to be most suitable for further analysis due to their moderately high R^2 -coefficient values: ~0.7 each, when these equations were assessed with separate test datasets, in Sections 4.2.3 and 4.3.3. Probability distribution plots and probability plots were used to measure the goodness of fit of each corrosion dataset from one of several probability distributions – normal, Rayleigh, exponential, lognormal and Weibull probability distributions. Tests for normality were first carried out on the datasets such as the Anderson-Darling and Kolmogorov-Smirnov tests followed by other probability distribution tests. Sensitivity analysis plots were also used to determine the relative significance of each of the input variables on CO₂ corrosion rate.

7.2 Probability Distributions

Probability distributions are theoretical distributions based on assumptions about a source population that assign a probability to the event that a random variable has a specific discrete value or a specified range of continuous values (Statistics and Machine Learning Toolbox User's Guide, 2015). In this light, an explanation of a random variable is necessary. A random variable is essentially a random number (Rice, 2007). It can also be defined as a variable whose precise value is unknown but the range of values it can assume and the probability of these values are known (Sobol, 1994).

7.3 Types of Probability Distributions

Probability distributions are broadly divided into two groups: Discrete and Continuous.

7.3.1 Discrete Probability Distributions

Discrete probability distributions are based on the discrete random variable and characterised by a probability mass function. A discrete variable is one which has a finite or countably infinite number of possible values and is often based on counts, such as the set of positive integers (Rice, 2007). Also, the probability mass function is the likelihood of occurrence of a given discrete variable to be equal to a certain fixed value (Stewart, 2009). Hence, if X is a discrete random variable, then the probability of X occurring is given by (Sobol, 1994):

$$\sum_n Pr(X = n) = 1 \quad (7.1)$$

Where:

Pr is the probability of an event occurring

n runs through all possible values of the random variable X

7.3.2 Continuous Probability Distributions

Continuous probability distributions are based on the continuous random variable and are characterised by the probability density function (pdf). A continuous variable is a variable which has an infinite number of values, whose elements can take any numeric value and can be measured at many different points. The set of real numbers which may include fractional, decimal or whole number integers are all elements of the continuous variable. Also, the probability density function is a function that describes the likelihood of the continuous random variable taking place. If X is a continuous random variable, then it has a pdf, $f(x)$.

Hence, the probability of X falling within a given range $[a, b]$ is given by (Sobol, 1994), (Harrison, 2010):

$$Pr[a \leq X \leq b] = \int_a^b f(x) dx = F(x) \quad (7.2)$$

Where:

$f(x)$ is the pdf of the random variable X

$F(x)$ is the integral of $f(x)$ and is the cumulative density function (cdf) of X over the range $[a, b]$

Given that all the variables discussed in this thesis (temperature, P_{CO_2} , flow velocity, pH, and corrosion rate) are continuous variables, Monte Carlo simulations were carried out using continuous probability distributions. The main probability distributions used as starting references were the normal and Weibull distributions. The probability density functions for these distributions are shown in Table 7-1.

Table 7-1. Probability density functions for the normal and Weibull distributions (Harrison, 2010), (Johnson et al., 1994)

Probability Distribution	Probability Density Function
Normal	$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \text{ for } x \geq 0 \quad (7.3)$ <ul style="list-style-type: none"> ▪ μ and σ are the mean and standard deviation respectively
2-Parameter Weibull	$f(x; k_W, \lambda) = \frac{k_W}{\lambda} \left(\frac{x}{\lambda}\right)^{k_W-1} e^{-\left(\frac{x}{\lambda}\right)^{k_W}} \text{ for } x \geq 0 \quad (7.4)$ <ul style="list-style-type: none"> ▪ k_W and λ are the shape and scale parameters respectively
3-Parameter Weibull	$f(x; k_W, \lambda, \theta) = \frac{k_W}{\lambda} \left(\frac{x-\theta}{\lambda}\right)^{k_W-1} e^{-\left(\frac{x-\theta}{\lambda}\right)^{k_W}} \text{ for } x \geq 0 \quad (7.5)$ <ul style="list-style-type: none"> ▪ k_W, λ and θ are the shape, scale and location parameters respectively

The 3-parameter Weibull pdf becomes the 2-parameter Weibull pdf when the location parameter, $\theta = 0$. The 2-parameter Weibull pdf is therefore a special case of the 3-parameter Weibull pdf.

7.4 Monte Carlo Simulation of Low Pressure CO₂ Corrosion Dataset

7.4.1 Experimental Dataset

The experimental CO₂ corrosion rate data was first tested to determine the best-fit distribution: (Dugstad et al., 1994b; Nordsveen et al., 2003). The distributions tested against the dataset include the normal, Rayleigh and Weibull distributions. Probability plots of the normal and Weibull distributions for the given dataset are shown in Figure 7-1.

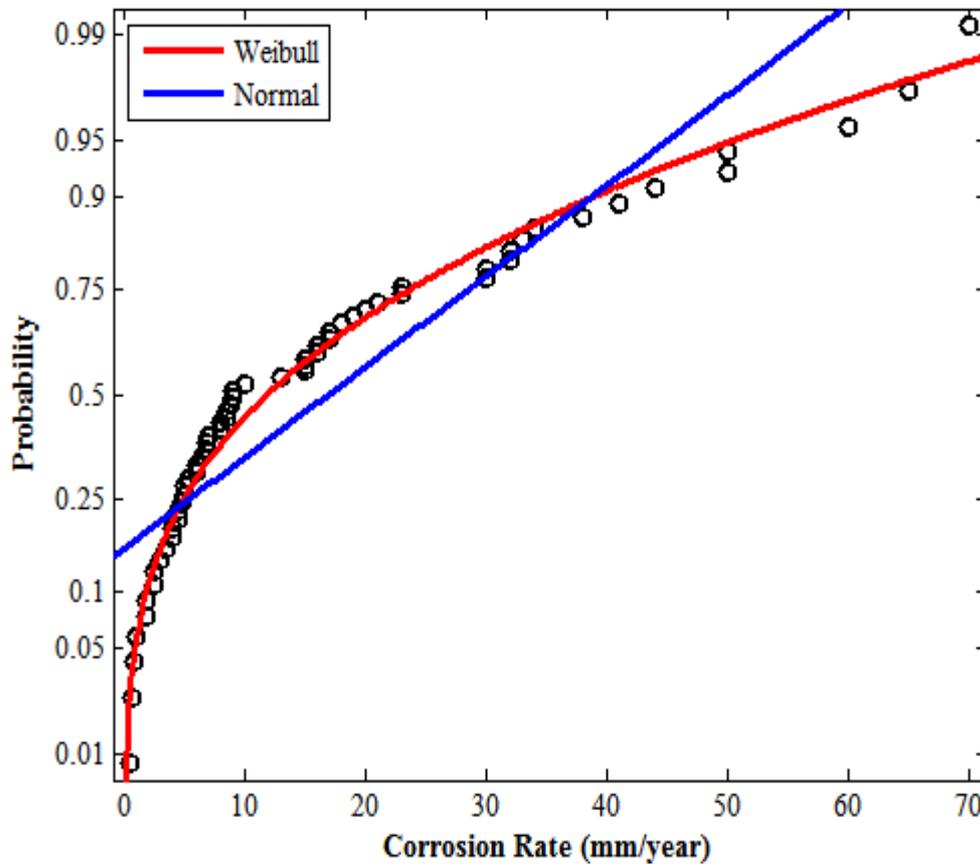


Figure 7-1. Normal probability plot and Weibull probability plot

It is observed from Figure 7-1, that the Weibull probability plot is a better fit for the dataset than the normal probability plot and as such will be used in modelling the Monte Carlo simulation. Table 7-2 shows a summary of statistical tests carried out on the experimental CO₂ corrosion rate data that confirms the Weibull distribution as a better fit than the Normal distribution.

Table 7-2. Test statistic summary for normal and Weibull Probability density functions

Test Statistic	Normal Distribution	Weibull Distribution
Anderson-Darling (AD)	3.586; p -value<0.005	0.326; p -value>0.5
Kolmogorov-Smirnov (KS)	Rejects Null hypothesis (H_0)	-

It can be concluded from the results that a distribution is a better fit provided that the magnitude of the AD-test statistic is small and the p -value is greater than 0.5 at the 5% significance level ($\alpha=0.05$). The KS-test for normality rejects the null hypothesis at the 5% confidence level. Given that the null hypothesis in this instance is the statement that there are no significant differences between the dataset and the distribution fit, the rejection of the null implies that there are indeed significant differences between the dataset and the distribution fit. This provides further proof that the Weibull pdf is better at modelling experimental CO₂ corrosion rates.

Probability density plots of the normal and Weibull distributions for the given dataset are shown in Figure 7-2. It is also observed from Figure 7-2, that the histogram fit for the Weibull probability density function, is a better representation of the dataset than the histogram fit for the normal pdf. The characteristic feature of the density plot is that the sum of the areas of all the histogram bars is equal to unity. This implies that the area of a histogram bar for a given corrosion rate range represents the probability of obtaining that given corrosion rate range.

The exponential and lognormal probability distributions are also close fit functions for the data. The exponential distribution is directly related and belongs to the Weibull family of distributions while the lognormal distribution is known to be a good fit for natural processes, chemical reactions as well as for materials undergoing failure due to reactions such as stress and corrosion (Lee and Wang, 2003; Gronh lm and Annila, 2007; Baboian et al., 2005). In general, both the Weibull and lognormal distributions are used extensively in reliability and survival analyses (Lee and Wang, 2003). Both are good fits for data that is positively skewed (Fahidy, 2005). The corrosion rate variable from the low pressure CO₂ dataset is positively skewed as discussed in Section 4.2.1 and seen from the boxplot in Figure 4-1. The lognormal and exponential pdfs are shown in Table 7-3.

**Table 7-3. Probability density functions for the lognormal and exponential distributions
(Harrison, 2010; Johnson et al., 1994)**

Probability Distribution	Probability Density Function
2-parameter Lognormal	$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\left\{-\frac{(\ln x - \mu)^2}{2\sigma^2}\right\}} \text{ for } x \geq 0 \text{ (7.6)}$ <ul style="list-style-type: none"> ▪ μ and σ are scale parameters respectively
3-parameter Lognormal	$f(x; \mu, \sigma, \gamma) = \frac{1}{(x - \gamma)\sigma\sqrt{2\pi}} e^{\left\{-\frac{[\ln(x - \gamma) - \mu]^2}{2\sigma^2}\right\}} \text{ for } x \geq 0 \text{ (7.7)}$ <ul style="list-style-type: none"> ▪ μ, σ are scale parameters and γ is the threshold
Exponential	$f(x; \lambda) = \lambda e^{-\lambda x} \text{ for } x \geq 0 \text{ (7.8)}$ <ul style="list-style-type: none"> • λ adjusts the exponential decay

The 3-parameter lognormal pdf reduces to the 2-parameter lognormal pdf, when the threshold parameter: $\gamma = 0$.

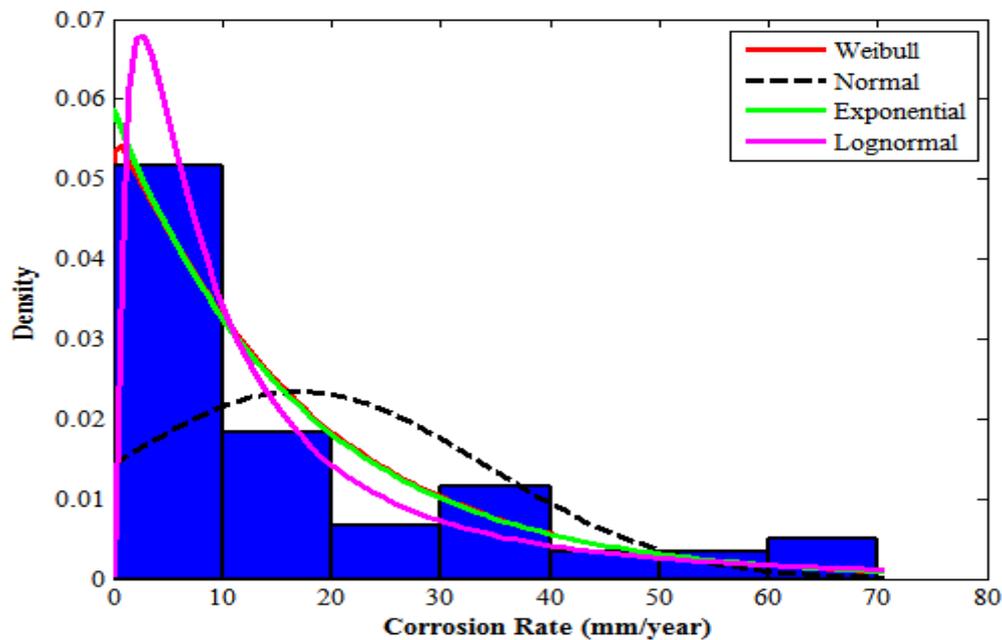


Figure 7-2. Normal probability density and Weibull probability density plots for experimental CO₂ corrosion rate dataset

The following Table summarises the statistical parameters of each distribution.

Table 7-4. Statistical parameters for each distribution

Statistical Parameter	Weibull	Normal	Exponential	Lognormal
Mean (mm/year)	17.03	17.03	17.03	19.38
Standard Deviation (mm/year)	16.78	17.02	17.03	32.74
Shape factor	17.14	-	-	μ (log location) = 2.29
Scale factor	1.02	-	μ (scale) = 17.03	σ (log scale) = 1.14

7.4.2 Monte Carlo Dataset

As previously stated, the Monte Carlo method involves the generation of random numbers for simulation purposes. The aim of this simulation is to compare the percentage errors in means and standard deviations of the simulated data against the experimental data.

Simulations were carried out for two groups. Case Study I involved the modelling of each input parameter as a uniform probability distribution function in conjunction with a previously-derived regression equation. Case study II involved the determination of the best-fit probability distribution function for each parameter and the subsequent application of these distribution functions in a pre-derived regression equation.

For every test simulation, 100,000 (1×10^5) Monte Carlo simulation runs were executed, resulting in a randomly generated dataset of 100,000 CO₂ corrosion rates. In general, increasing the simulation runs decreases the size of the class intervals (bins) of the histograms, leading to smoother representative probability density function plots however this comes as a cost to computation. Also, there is a risk of the appearance of very thin comb-like lines in the pdf plot if the simulation runs are increased indefinitely. The number of runs chosen is shown in Figure 7-3. As simulation runs increased by an order of magnitude, likewise time elapsed increased. This however meant that at one million (1×10^6) runs, the time elapsed for this section of code increased to 0.1 seconds and subsequently, a whole second for ten million (1×10^7) runs. Therefore, the point before the angled deviation on the plot was chosen as the ideal for simulation runs. The difference between 0.1s and 1s is small

however in a context where optimisation is required (discussed in Section 7.6), a bit more time may be required to run through all the lines of code. A Monte Carlo simulation was performed based on the statistical parameters of the experimental dataset. Figure 7-4 shows the result of this process. The simulated data was modelled specifically on the shape and scale factors of the Weibull distribution, shown in Table 7-4. This was done because the probability plot and the two statistical tests, whose results are shown in Table 7-2, revealed the Weibull to be the better fit.

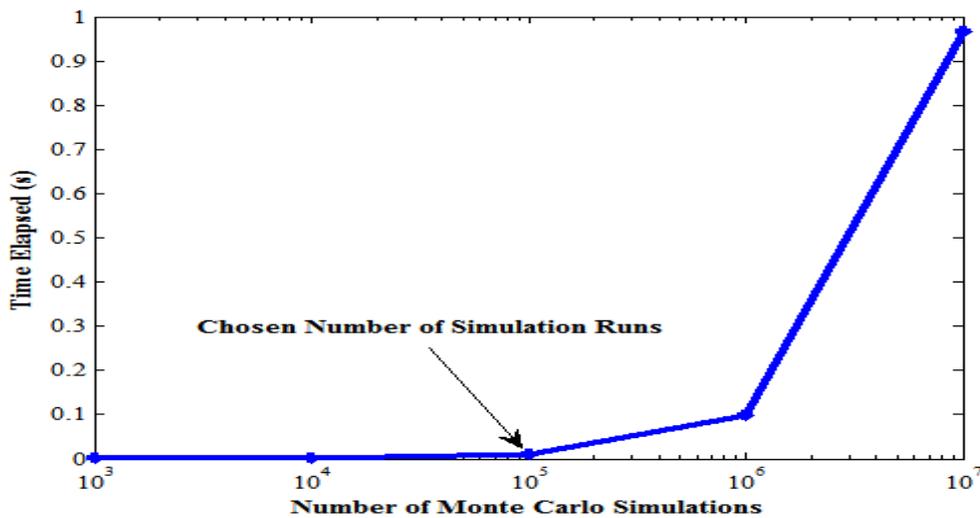


Figure 7-3. Number of runs used in Monte Carlo simulations

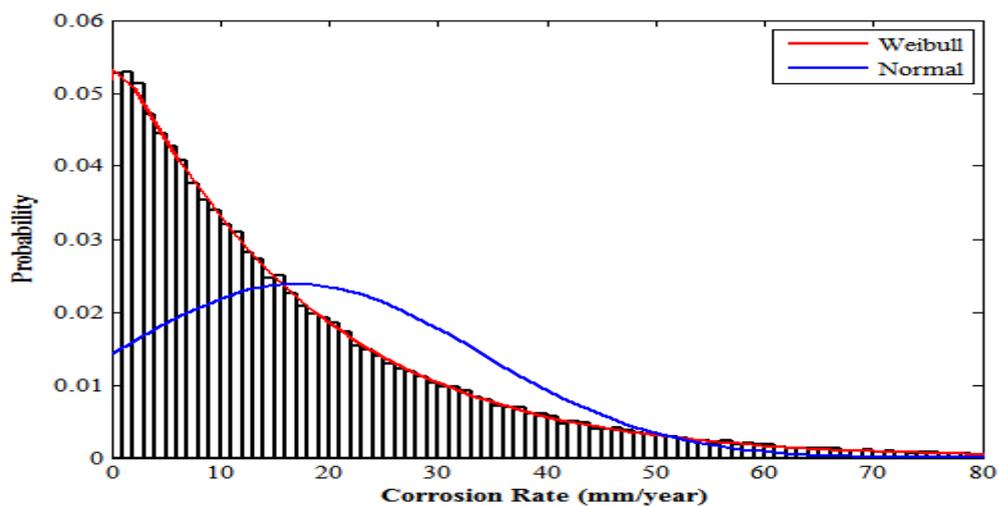


Figure 7-4. Monte Carlo simulation using statistical parameters of the experimental dataset

The Weibull model fit statistics were then compared against those for the experimental data. Table 7-5 is a summary of the results.

Table 7-5. Summary statistics for the use of the Weibull distribution as a direct fit for the low pressure CO₂ corrosion rate

Statistical Parameter	Experimental Corrosion Rate (mm/year)	Weibull Monte Carlo Model (mm/year)	Percentage Error (%)
Mean	17.03	17.02	0.00*
Standard Deviation	17.02	16.78	1.41*

* Percentage Errors are calculated as follows: $Percentage\ Error\ (\%) = \frac{|Monte\ Carlo\ Corr\ Rate - Exp.\ Corr\ Rate|}{Exp.\ Corr\ Rate} \times 100\%$

Two other theoretical distributions had also been found to be good fits to the corrosion rate experimental data as seen in Figure 7-2. However, where the Monte Carlo simulation runs of the exponential model improved on the statistics of the Weibull, the variance for the lognormal model was quite high.

Table 7-6 Summary statistics for the exponential and lognormal probability distributions

Statistical Parameter	Experimental Corrosion Rate (mm/year)	Exponential Monte Carlo (mm/year)	Lognormal Monte Carlo (mm/year)
Mean	17.03	17.03 (0.00)*	19.39 (13.86)*
Standard Deviation	17.02	17.03 (0.06)*	32.76 (92.50)*

* Figures in brackets are percentage errors with respect to experimental data

Percentage Errors are calculated as follows: $Percentage\ Error\ (\%) = \frac{|Monte\ Carlo\ Corr\ Rate - Exp.\ Corr\ Rate|}{Exp.\ Corr\ Rate} \times 100\%$

From the results of Table 7-5 and 7-6, it is concluded that the exponential distribution is a better distribution than the Weibull for representing low pressure CO₂ corrosion data.

Case Study I - Based on the Derived Regression Equation

The regression equation. (7.3) derived in Section 7.2.3 for the low pressure CO₂ dataset was used as a basis for carrying out further Monte Carlo simulations. Uniform probability distributions were used to model each parameter. The simulated temperature is evaluated using the following expression:

$$T_{sim} = T_0 + \{(T_1 - T_0) \times N\} \quad (7.1)$$

Where:

T_{sim} is the simulated temperature (°C)

T_0 is the lowest temperature (°C)

T_1 is the highest temperature (°C)

N is the randomly generated number

The form of the equation shown in (7.1) was used for each of the other variables, replacing T with P_{CO_2} , pH and velocity accordingly. Given that the number of simulated corrosion rates is very large ($N=100,000$), a test sample subset ($N=100$) was randomly selected for the purpose of carrying out the AD and KS statistical tests. The results of these tests are shown in Table 7-7. The Mersenne twister was chosen as the algorithm for the generation of pseudo-random numbers. It was implemented in Matlab with a seed of zero due to its high number generation speed in comparison to other modern generators, its long period length and has also passed several stringent statistical tests for randomness (Matsumoto and Nishimura, 1998).

Table 7-7. Test statistic summary for normal and Weibull probability density functions

Test Statistic	Normal Distribution	Weibull Distribution
Anderson-Darling (AD)	0.427; p -value=0.307	0.232; p -value>0.5
Kolmogorov-Smirnov (KS)	Rejects Null hypothesis (H_0)	-

The AD-statistic of the test sample for both the normal and Weibull distributions is low in magnitude and as such the dataset can be adequately modelled by both distributions. The KS-test for normality rejects the null hypothesis. Given that the null hypothesis is the assumption that no significant differences exist between the sample test and a typical normal distribution, rejection of the null therefore implies significant differences exist.

Figure 7-5 shows the probability density function plot for the simulated corrosion rates. It is observed that both normal and Weibull probability curves fit the data adequately particularly for very low and very high corrosion rates (tailends of the dataset) however both curves seem slightly off for corrosion rates in the 20-35mm/year range. This imprecise prediction can also be observed on the cumulative density function (cdf) plot for the 20-35mm/year range, shown in Figure 7-6. In general, the Weibull distribution is seemingly a closer fit to the corrosion rate data than the normal hence is used to model this dataset.

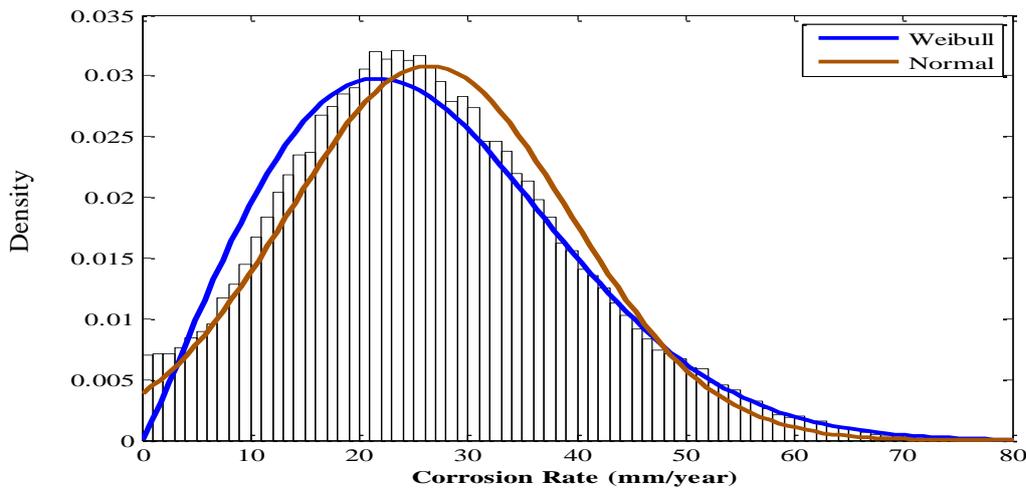


Figure 7-5. Probability density function plot for Monte Carlo simulation of low pressure CO₂ corrosion rate

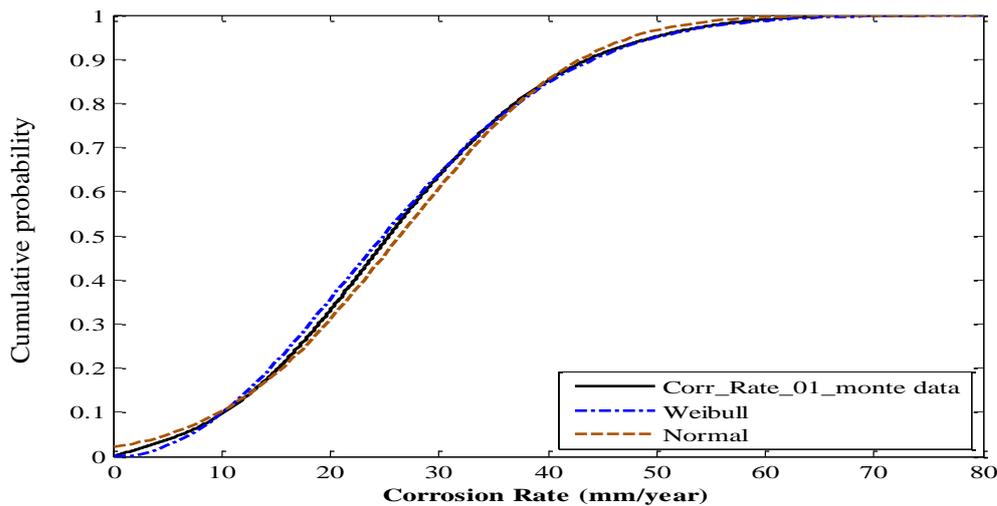


Figure 7-6. Cumulative density function plot for Monte Carlo simulation of low pressure CO₂ corrosion rate

Table 7-8 shows the summary statistics for Case study I and how they compare against those of the experimental corrosion dataset. It is observed that while there is a sizeable error in the estimated mean, the standard deviation estimate is closer to the true value. A second case study is carried out while assuming best-fit distributions for each predictor variable and the results of the study will be compared against that of Case Study I. This is described in the next Section.

Table 7-8. Summary statistics for case study I

Statistical Parameter	Experimental Corrosion Rate (mm/year)	Monte Carlo Model Corrosion Rate (mm/year)	Percentage Error (%)
Mean	17.03	26.40	55.03*
Standard Deviation	17.02	12.95	23.88*

* Percentage Errors are calculated as follows: $\text{Percentage Error (\%)} = \frac{|\text{Monte Carlo Corr Rate} - \text{Exp. Corr Rate}|}{\text{Exp. Corr Rate}} \times 100\%$

Case Study II - Based on the Derived Regression Equation

Datasets for each variable were plotted on a normal probability plot in order to test for normality. Once it was established by inspection that a given dataset did it follow the normal distribution, it was then tested against the Rayleigh, Uniform and Weibull distributions by plotting the probability density functions for the dataset. The best-fit distribution for the variable dataset usually came from one of the aforementioned theoretical distributions. The best-fit distribution for each variable is summarised in Table 7-9.

Table 7-9. Best-fit distributions for each input parameter

Predictor Variable	Probability Distribution
Temperature	Uniform
CO ₂ Partial Pressure (P_{CO_2})	Weibull
pH	Normal
Velocity	Uniform

The best-fit distributions listed in Table 7-9 were used as equivalent representations for the corresponding predictor variables in the derived expression obtained from multiple linear

regression on the low pressure CO₂ corrosion dataset (Equation. 7.3), and is used to perform Monte Carlo simulation. Table 7-10 is a summary of the test statistics.

Table 7-10. Test statistic summary for normal and Weibull probability density functions

Test Statistic	Normal Distribution	Weibull Distribution
Anderson-Darling (AD)	0.605; p -value=0.113	0.579; p -value=0.114
Kolmogorov-Smirnov (KS)	0.0766; p -value=0.5881	0.0749; p -value=0.6209

The AD-test value for the Weibull distribution is smaller in magnitude than the corresponding test-value for the normal distribution, $0.579 < 0.605$. Also, the p -value of the AD-test for the Weibull distribution is greater than the corresponding value for the normality test at the 5% significance level. This proves the Weibull distribution is marginally better suited to fit the data. The values from the KS-test also support the results of the AD-test.

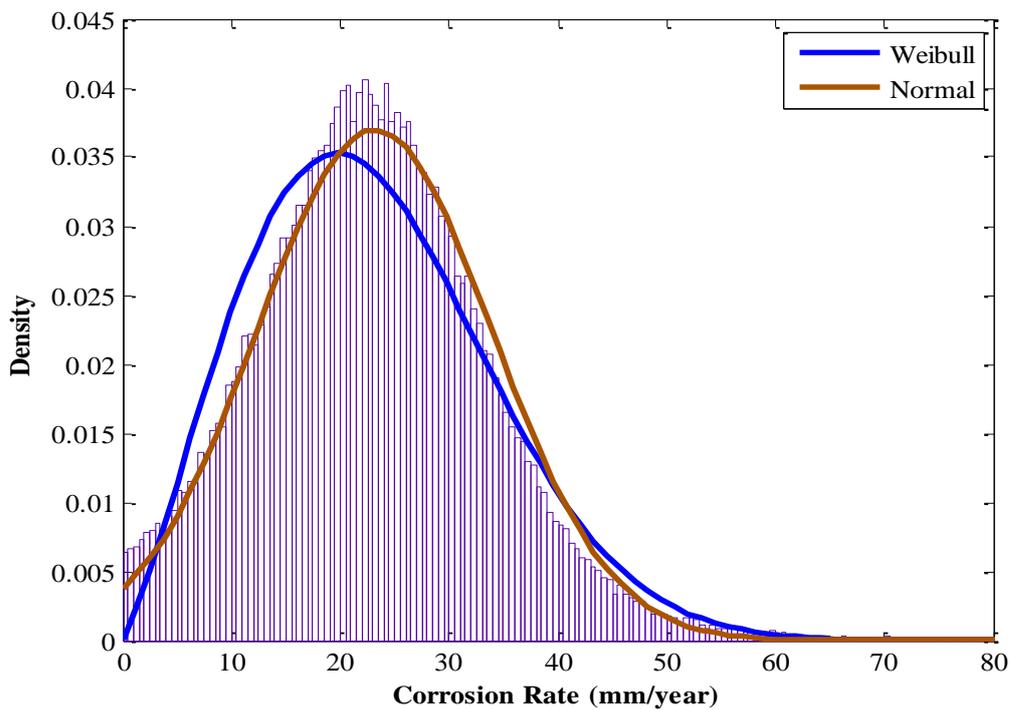


Figure 7-7. Probability density function plot for Monte Carlo simulation of low pressure CO₂ corrosion rate

The plot in Figure 7-7 is the result of the Monte Carlo simulation using the best-fit Weibull statistical parameters for shape and scale. This plot is approximately normal in appearance despite the pH being the only predictor variable to be modelled as normally-distributed. This observation is explained by the Central limit theorem, which states that with sufficiently large or a near-infinite number of draws, the arithmetic mean of independent random variables, each with their distinct averages and variances converges to the normal distribution (Rice, 2007).

Figure 7-8 is the cumulative density function plot of the distribution in Figure 7-7. It is seen that both Weibull and normal distributions closely match the simulated data, such that it is difficult to tell which distribution is more suitable by mere inspection. This emphasises the significance of the statistical test carried out in Table 7-10.

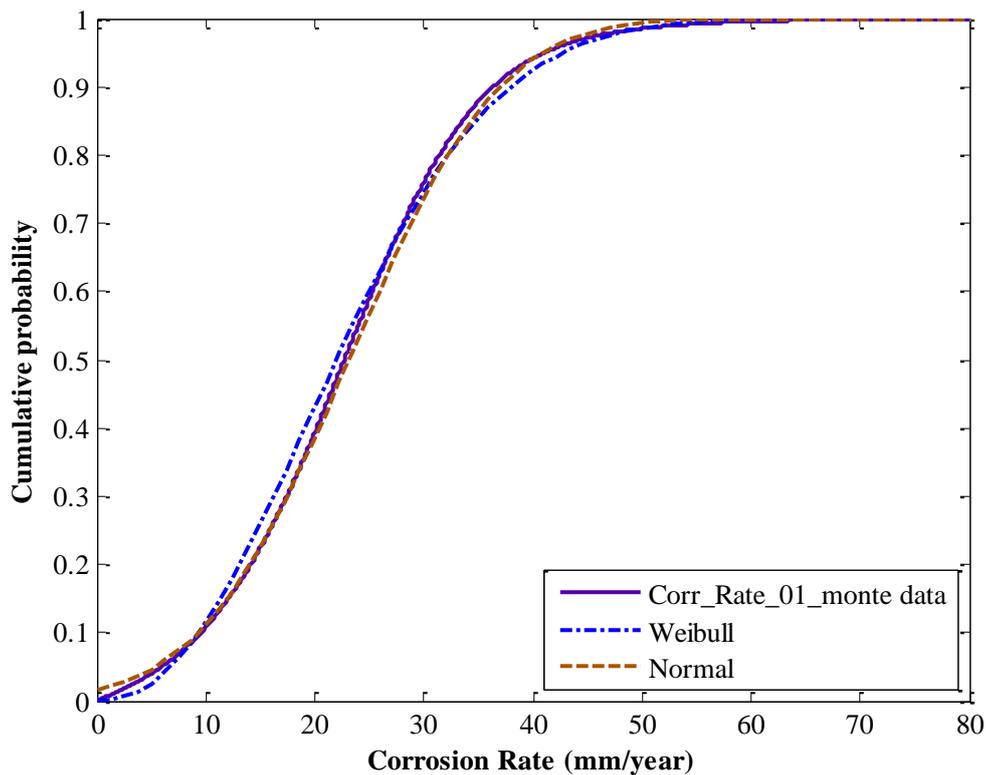


Figure 7-8. Cumulative density function plot for Monte Carlo simulation of low pressure CO₂ corrosion rate

Thus, comparing the results from Case Study I, shown in Table 7-8, with the results of Case Study II, shown in Table 7-11, the drop of approximately 20% in the mean percentage error

suggests that the results of Case Study II are marginally better than Case Study I, even though the standard deviation increases by ~13%.

Table 7-11. Summary statistics for case study II

Statistical Parameter	Experimental Corrosion Rate (mm/year)	Monte Carlo Model Corrosion Rate (mm/year)	Percentage Error (%)
Mean	17.03	23.13	35.58*
Standard Deviation	17.02	10.79	36.57*

* Percentage Errors are calculated as follows: $Percentage\ Error\ (\%) = \frac{|Monte\ Carlo\ Corr\ Rate - Exp.\ Corr\ Rate|}{Exp.\ Corr\ Rate} \times 100\%$

Sensitivity Analysis

Case Study II was used as the basis for the Monte Carlo sensitivity analysis. Figure 7-9 shows the sensitivity plot for corrosion to each of the predictor variables for the low pressure CO₂ dataset.

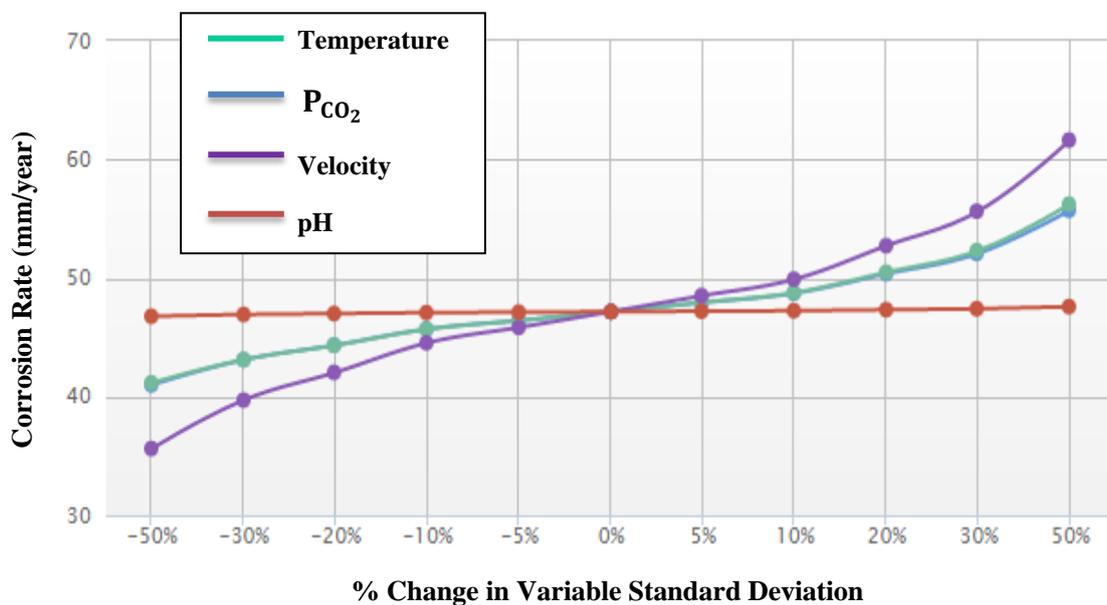


Figure 7-9. Sensitivity analysis – effect of each parameter on corrosion rate

It is observed that the corrosion rate is most sensitive to changes in the flow velocity and least sensitive to changes in pH. A physicochemical explanation for this fact is that for low pressure CO₂ corrosion, the flow velocities need to be closely monitored due to the washing

away of protective film formation on the metal surface. Higher velocities are likely to trigger greater corrosion rates in carbon steel pipes by more readily exposing the metal surface. This is so because scales are thinner, less dense and less compact than for high CO₂ pressure conditions (Zhang et al., 2012b). The effect of temperature and CO₂ partial pressure can be seen as the second most-influential predictor variables. Their effect here can be explained from the principal component analysis (PCA) results from Section 7.2.2, as they are the greatest contributors to the variation within this dataset. The pH is the least contributor to the variation within the dataset from PCA analysis, which also can be seen to have little impact on corrosion rates even with a $\pm 50\%$ change in the magnitude of its standard deviation. The velocity variable is the only variable that does not follow the PCA order in terms of influence and statistically, this may be due to the imperfect mean and standard deviation percentage errors for Case Study II with respect to the experimental dataset. The best-fit distributions are the best possible matches to the predictor variable datasets but cannot completely describe them due to the presence of random errors in measurements or noise (Cottis et al., 1999).

7.5 Monte Carlo Simulation of High Pressure CO₂ Corrosion Dataset

7.5.1 Experimental Dataset

Figure 7-10 shows that though both the Weibull and normal probability distributions represent the data adequately, the normal line plot is a slightly better fit. Results from the summary statistics Table 7-12, confirm the normal distribution as the slightly better fit. Firstly, the magnitude of the AD-test statistic is smaller than that of the Weibull, given that the smaller test result is the better statistic. Also, the corresponding *p*-values are greater for the former than for the latter at the 5% significance level (0.05).

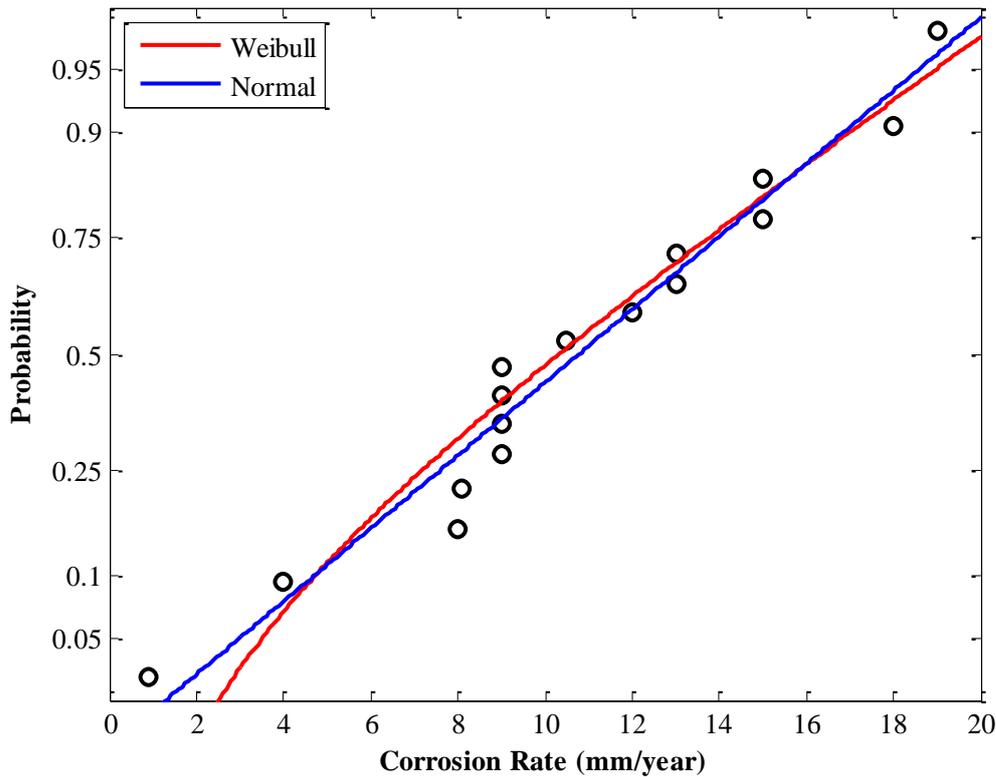


Figure 7-10. Normal probability plot and Weibull probability plot

For the KS-test statistic, again, the smaller the magnitude of the test result, the better, while the p -value is greater for the normal than for the Weibull distribution suggesting that the probability of obtaining a good fit is slightly better for the Normal than the Weibull.

Table 7-12. Test statistic summary for normal and Weibull probability density functions

Test Statistic	Normal Distribution	Weibull Distribution
Anderson-Darling (AD)	0.314; p -value=0.513	0.492; p -value=0.211
Kolmogorov-Smirnov (KS)	0.164; p -value=0.815	0.168; p -value=0.776

Figure 7-11 shows the probability density plot for the high pressure CO₂ corrosion dataset. By inspection, both normal and Weibull curves appear to fit the data accurately. However, the AD and KS tests show the data more likely comes from a normal distribution.

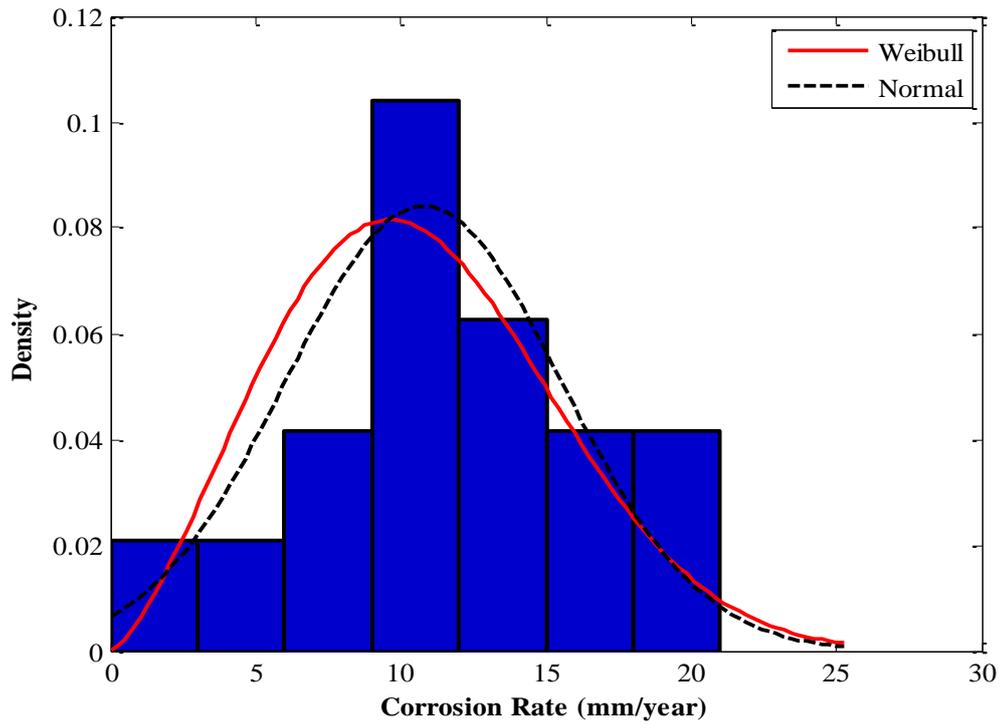


Figure 7-11. Normal probability density and Weibull probability density plots for experimental CO₂ corrosion rate dataset

Table 7-13 shows comparative statistics for the Weibull and normal distributions.

Table 7-13. Statistical parameters for each distribution

Statistical Parameter	Weibull	Normal
Mean (mm/year)	10.66	10.78
Standard Deviation (mm/year)	4.72	4.75
Shape factor	12.03	-
Scale factor	2.41	-

7.5.2 Monte Carlo Dataset

The Monte Carlo datasets were modelled using 100,000 simulations. Figure 7-12 is a probability density function plot for the Monte Carlo simulated corrosion rates. The Weibull distribution does not fit the data hence is not shown.

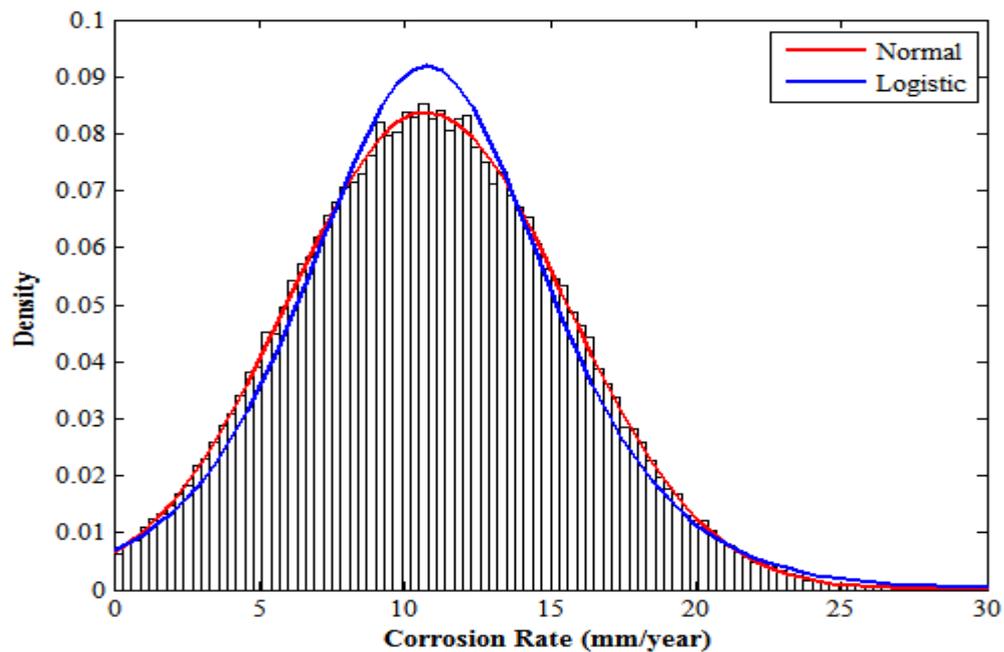


Figure 7-12. Monte Carlo simulated probability density function plot showing normal and logistic curve fits

Both the normal and logistic curves are good fits for the data. However, the logistic curve hangs slightly over the centrally-located histograms at the peak of the pdf. Since the data comes from a normal distribution, the normal curve is a good fit. The following Table shows the summary statistics for both Normal and Logistic distributions. The normal Monte Carlo outperforms the alternative Logistic distribution in terms of the calculated statistical parameters.

Table 7-14 Summary statistics for the normal and logistic probability distributions

Statistical Parameter	Experimental Corrosion Rate (mm/year)	Normal Monte Carlo (mm/year)	Logistic Monte Carlo (mm/year)
Mean	10.78	10.79 (0.09)*	10.79 (0.09)*
Standard Deviation	4.75	4.76 (0.21)*	4.94 (4.00)*

* Figures in brackets are percentage errors with respect to experimental data

Percentage Errors are calculated as follows: $\text{Percentage Error (\%)} = \frac{|\text{Monte Carlo Corr Rate} - \text{Exp. Corr Rate}|}{\text{Exp. Corr Rate}} \times 100\%$

Case Study I - Based on the Derived Regression Equation

The regression Equation. (7.6) derived in Section 7.3.3 for the high pressure CO₂ dataset was used as a basis for carrying out Monte Carlo simulations for this Case Study. Again, uniform probability distributions were used to model each variable. Equation (7.1) was used in the formulation of the uniform distributions for each of the variables in the dataset: temperature, CO₂ partial pressure, flow velocity and pH. Table 7-15 is a summary of the test statistic results for simulations of the data using the Normal and Weibull distributions.

Table 7-15. Test statistic summary for normal and Weibull probability density functions

Test Statistic	Normal Distribution	Weibull Distribution
Anderson-Darling (AD)	0.624; <i>p</i> -value=0.101	0.618; <i>p</i> -value=0.075
Kolmogorov-Smirnov (KS)	0.064; <i>p</i> -value=0.813	0.064; <i>p</i> -value=0.814

The results of the given test statistics imply that both distributions closely match the data. The magnitude of the AD-test result is slightly smaller for the Weibull than for the normal distribution however the *p*-value at the 5% significance level is higher for the Weibull than for the normal distribution. Hence both distributions are as good as each other. The result of the KS-test statistic implies given that both distributions have the same KS-test value, the Weibull is slightly better because of the higher *p*-value.

For the probability density function plot of Figure 7-13, both distributions also show how closely they match-up against the given data. The cdf plot of Figure 7-14 also shows that by inspection, the better fit might be difficult to deduce. The Weibull curve fits the data less accurately for corrosion rates in the range: 8 to 16mm/year as well as for corrosion rates greater than 20mm/year. The Normal distribution matches up quite closely with the data except for corrosion rates in the range: 0-8mm/year. Despite the close fit of the normal curve, by visual inspection, the AD-test statistic is sufficiently conclusive.

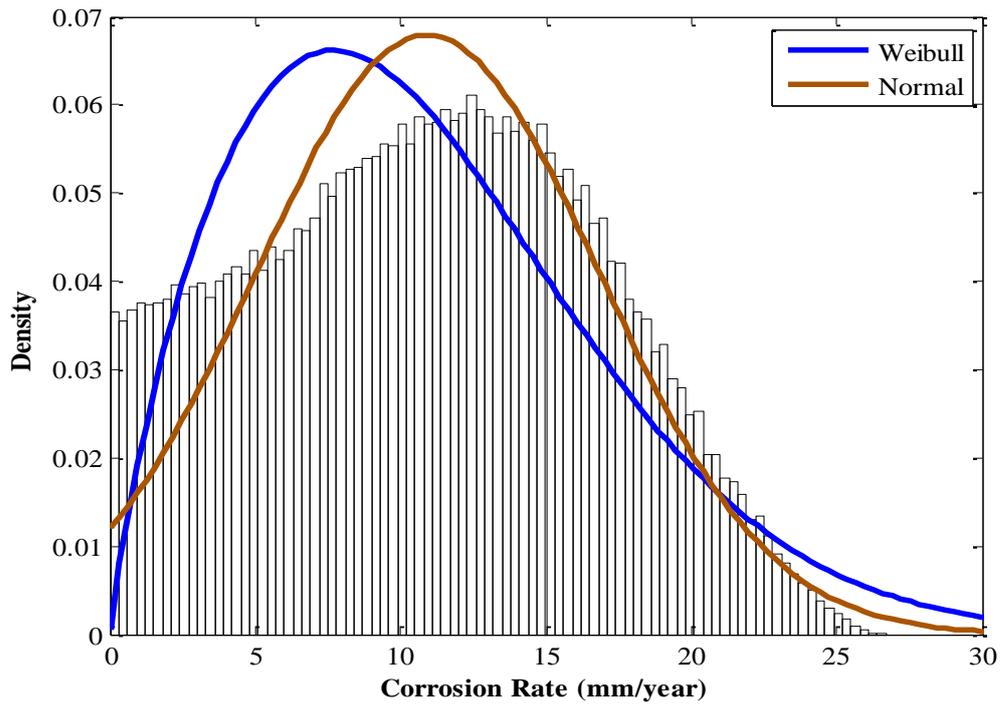


Figure 7-13. Probability density function plot for Monte Carlo simulation of high pressure CO₂ corrosion rate

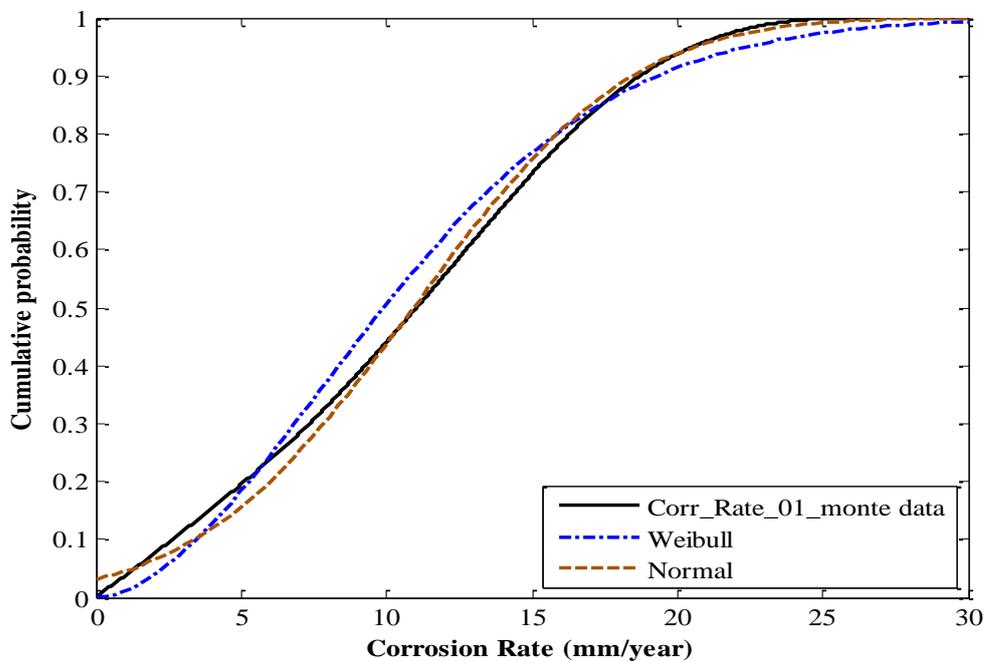


Figure 7-14. Cumulative density function plot for Monte Carlo simulation of high pressure CO₂ corrosion rate

Table 7-16 shows summary statistics for Case Study I. The mean of the Monte Carlo dataset is accurate with a percentage error of 0.56% however the standard deviation is not as accurate. Optimisation of the Monte Carlo dataset would be beneficial. This is discussed in Section 7.6.

Table 7-16. Summary statistics for case study I

Statistical Parameter	Experimental Corrosion Rate (mm/year)	Monte Carlo Model Corrosion Rate (mm/year)	Percentage Error (%)
Mean	10.78	10.84	0.56*
Standard Deviation	4.75	5.90	24.21*

* Percentage Errors are calculated as follows: $\text{Percentage Error (\%)} = \frac{|\text{Monte Carlo Corr Rate} - \text{Exp. Corr Rate}|}{\text{Exp. Corr Rate}} \times 100\%$

Case Study II - Based on the Derived Regression Equation

Best-fit distributions were determined for each of the variables and Table 7-17 summarises the results.

Table 7-17. Best-fit distributions for each input parameter

Input Parameter	Probability Distribution
Temperature	Uniform
CO ₂ Partial Pressure (P_{CO_2})	Weibull
pH	Normal
Velocity	Uniform

Table 7-18 shows the summary statistics for the Normal and Weibull probability density functions.

Table 7-18. Test statistic summary for normal and Weibull probability density functions

Test Statistic	Normal Distribution	Weibull Distribution
Anderson-Darling (AD)	0.694; p -value=0.068	0.801; p -value>0.5
Kolmogorov-Smirnov (KS)	0.066; p -value=0.784	0.067; p -value=0.773

From the Table, the AD and KS test statistic values have smaller magnitudes for the normal distribution than for the Weibull distribution. This implies that the data comes from a normal distribution. However, the p -value for the AD test is greater for the normal distribution than for the Weibull distribution. For the KS test, the p -value at the 5% significance level is 0.784, which is greater than the corresponding value for the Weibull distribution. This leads to the conclusion that the data is normally-distributed. Figure 7-13 shows the Monte Carlo probability density function plots for both normal and Weibull distributions.

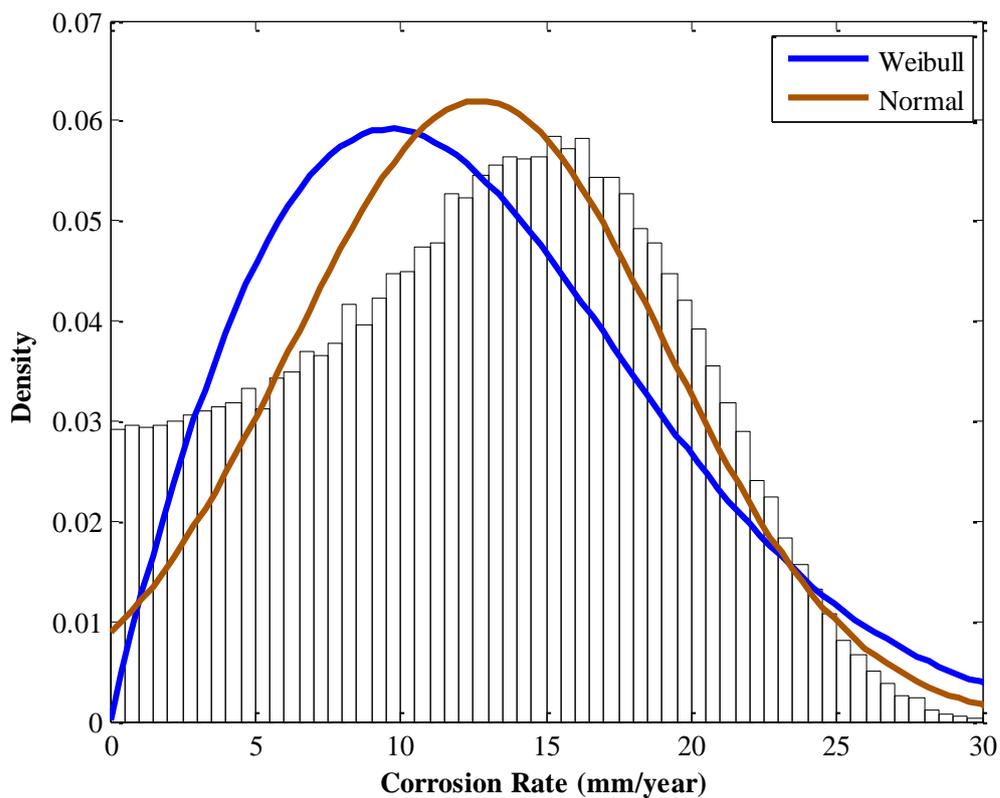


Figure 7-15. Probability density function plot for Monte Carlo simulation of high pressure CO₂ corrosion rate

It is observed from the pdf plot, that while the Weibull curve originates at point (0, 0), it does not match-up closely against the data for very low and very high-magnitude corrosion rates. The normal curve on the other hand, performs better at fitting the data, particularly for the high corrosion rates of 15-30mm/year. Figure 7-16 shows the Monte Carlo cdf plot for the data with normal and Weibull cumulative distributions.

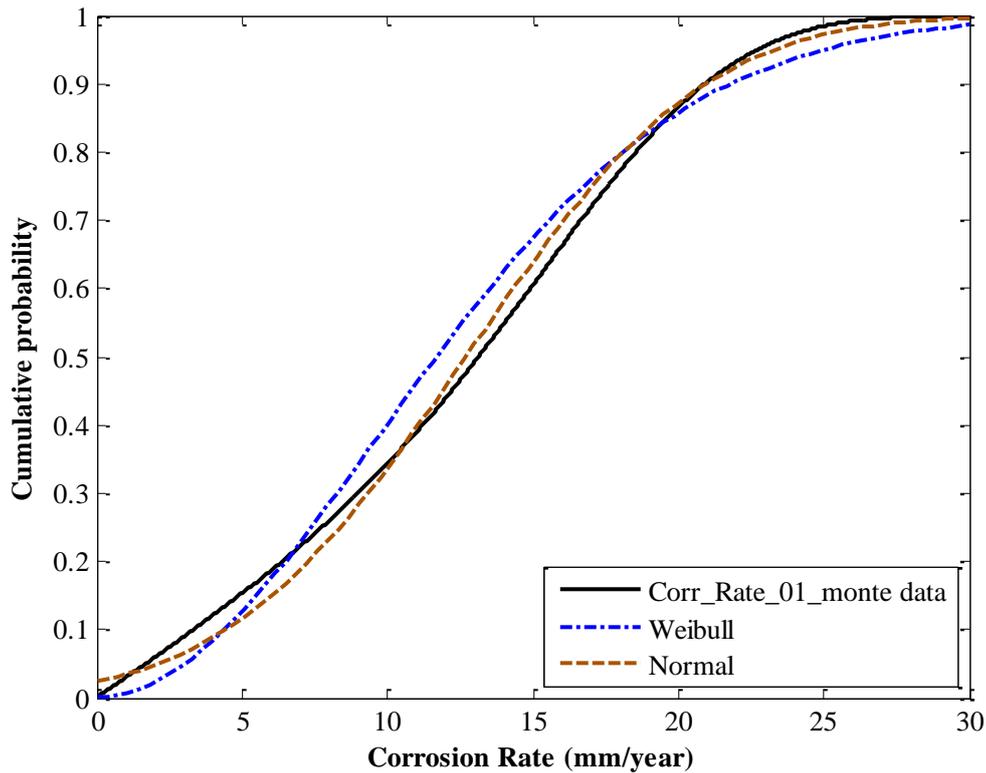


Figure 7-16. Cumulative density function plot for Monte Carlo simulation of high pressure CO₂ corrosion rate

While the shapes for both curves roughly match-up against the data, the Weibull curves deviates from the data for corrosion rates of the range: 7-17mm/year and for the tail-points of the data. The normal curve generally approximates the data better than the Weibull curve with evidently smaller deviations. Table 7-19 is the summary statistics for Case Study II.

Table 7-19. Summary statistics for case study II

Statistical Parameter	Experimental Corrosion Rate (mm/year)	Monte Carlo Model Corrosion Rate (mm/year)	Percentage Error (%)
Mean	10.78	12.65	17.35*
Standard Deviation	4.75	6.17	29.89*

* Percentage Errors are calculated as follows: $Percentage\ Error\ (\%) = \frac{|Monte\ Carlo\ Corr\ Rate - Exp.\ Corr\ Rate|}{Exp.\ Corr\ Rate} \times 100\%$

The results of this Case Study were compared with those of Case Study I, presented in Table 7-16. It is apparent that for both statistical parameters, Case Study I outperforms Case Study II. The predictor variables in Case Study I were modelled using Uniform distributions and not their best-fit distributions, the latter was applied for Case Study II, yet the Monte Carlo simulation results are better for the first Case Study than the second. This may be because the experimental dataset is quite small in size therefore what may appear to be the best-fit distribution may not necessarily be the case. The small size of the dataset makes it somewhat flexible to alternatives when Monte Carlo simulations are carried out.

Sensitivity Analysis

Case Study I was used to evaluate the sensitivity of corrosion rates to the various predictor variables of the high pressure CO₂ corrosion dataset. The Monte Carlo sensitivity plot is shown in Figure 7-17.

From the plot, it is observed that the corrosion rate was most sensitive to changes in temperature, CO₂ partial pressure, pH and flow velocity, in that order. The reason this is so is partly explained by the results from PCA of the high pressure CO₂ dataset in Section 4.3.2. From a statistical perspective, the temperature and CO₂ partial pressure induce the greatest influence on the variance for this dataset. In addition, the flow velocity and pH are less significant in terms of their contribution to the variance.

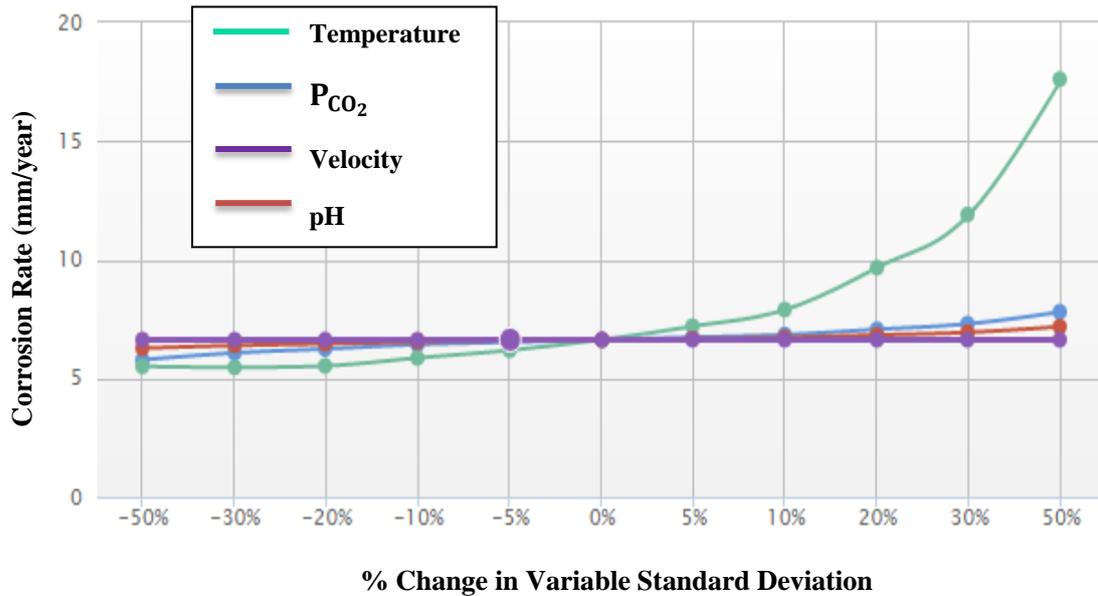


Figure 7-17. Sensitivity Analysis – effect of each parameter on corrosion rate

From a corrosion mechanism standpoint, the impact of the temperature is significant as it speeds up the rate of chemical reactions, which is explained by the Arrhenius theory (Tan and Chan, 2011). The corrosion rate is least sensitive to the flowrate velocity, which is contrary to the low pressure CO_2 dataset. The reason is that under high pressure CO_2 conditions, thicker, denser and more compact scales are formed and its presence on the metal surface provides a protective barrier against corrosion (Zhang et al., 2012b). These scales have a greater tendency to resist the effects of fluid flow such that its effect on the corrosion rate is limited.

7.6 Optimisation of Results

For both low pressure and high pressure CO_2 corrosion datasets, Case Study II and Case Study I produced the better Monte Carlo solutions for each group respectively, in terms of accuracy in predicting the means and standard deviations of their respective experimental datasets. Notwithstanding, in order to model the experimental data more accurately, optimisation is required.

For the low pressure CO_2 corrosion dataset, sensitivity analysis revealed corrosion rates to be most sensitive to changes in the velocity. For this reason, optimisation was performed on the velocity variable. The other three variables (temperature, P_{CO_2} and pH) were modelled using

their best-fit distributions while carrying out Monte Carlo simulations. The best-fit distributions were used given that the summary statistics (Case Study II) produced marginally better results than when modelling was carried out assuming uniformly distributed variables (Case Study I). The mean and standard deviation of the Monte Carlo simulated dataset were then equated with the corresponding statistics for the experimental dataset. An initial estimate for the velocity was chosen after which an iterative scheme was implemented in Matlab 2012a to determine the actual flow velocity for the optimisation process. During optimisation, the solutions for the flow velocity were: 5.47m/s. and 6.96m/s. Even though there is more than one solution for the optimised velocity, a conclusion can be drawn, which is the velocity needs to be greater than 5.47m/s for the simulated data to more accurately model the experimental dataset. The solution is more likely to be closer to 6.96m/s, given that the recorded mean for the velocity at low pressure stated in Section 4.2.1, Table 4-1 is: 7.20m/s. Though, the best-fit distributions were used in optimisation for each of the variables, these distributions were only the best-match to the variable datasets and not their ideal representations. It is highly unlikely that a given theoretical distribution will completely represent every given data-point in any given experimental dataset due to the presence of random errors or noise (Cottis et al., 1999). For instance, Figure 7-18 shows the CO₂ partial pressure dataset and its best-fit distribution. It can be seen that though the Weibull is a good-fit, it does not accurately model the tail-points of the CO₂ partial pressure data, i.e., the very low and the very high data-points.

Similarly for the high pressure CO₂ dataset, optimisation is carried out with respect to the variable that the corrosion rate is most sensitive to, which is temperature. Again, for high pressures, the percentage errors in Case Study I suggests adequate precision in modelling, hence the other variables are modelled using the uniform distribution. There is one solution result when the means and standard deviations of the Monte Carlo and experimental corrosion rates are identical. This solution is at 60.36°C. Though this value is well below the mean of temperatures for the high pressure corrosion dataset stated in Section 4.3.1, Table 4-5: (80.25°C), it must be noted that the assumption during optimisation is that all other variables are uniformly distributed.

In general, optimising the Monte Carlo data proved to be unusual because for the high pressure CO₂ dataset in particular, the size of the experimental dataset is quite small hence there is the possibility of having several fitting solutions or no solutions at all.

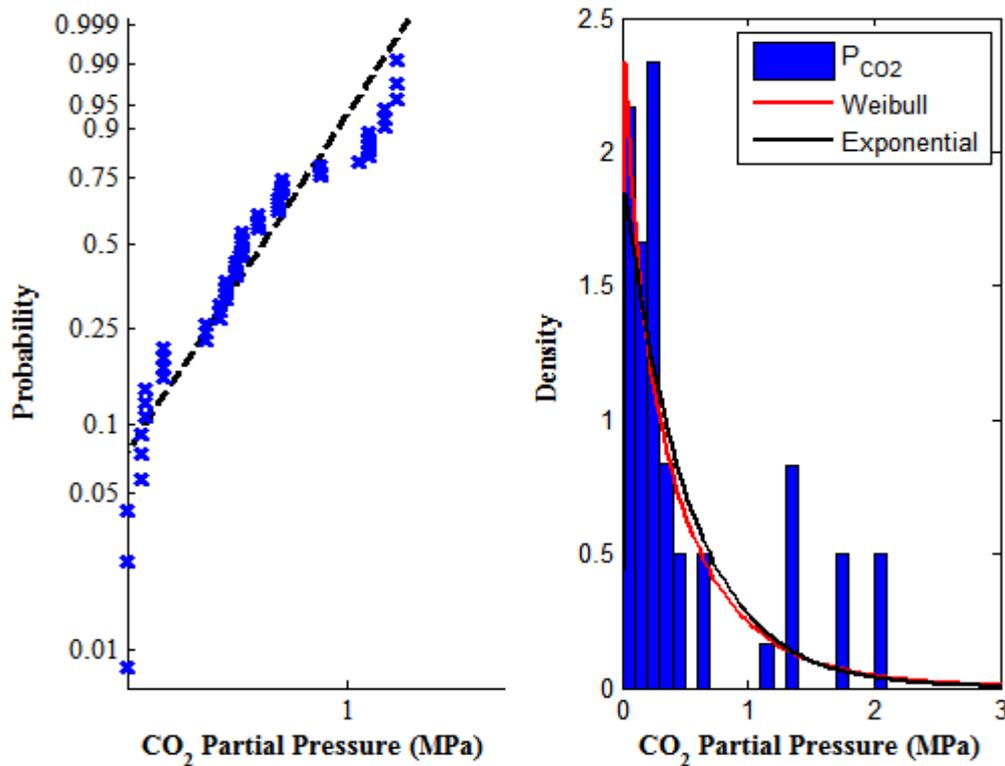


Figure 7-18. Probability plot showing the Weibull distribution as the best-fit for CO₂ partial pressure variable

7.7 Conclusions

Monte Carlo simulation is a technique involving the use of repeated sampling of a random discrete or continuous variable. The experimental datasets for low and high pressure CO₂ corrosion are of a continuous nature, in that the respective variables can be measured at different points and contain decimal numbers and not simply whole number integers as is applicable to discretised variables. An attempt was made to find a suitable range for the number of simulations to be used when carrying out Monte Carlo simulations. A range of between a hundred thousand to one million ($1 \times 10^5 - 1 \times 10^6$) simulations seemed sufficient for modelling.

The experimental datasets were first tested for normality using probability plots and then were subsequently fitted against several other theoretical distributions. The low CO₂ pressure corrosion dataset was found to come from the Weibull family of distributions. The Weibull, lognormal and exponential distributions were all good fits judging by the low percentage errors of the estimated distributions statistical parameters with respect to those of the

experimental dataset. The statistical reason for this is due to the positive skewness of the corrosion rate predictor variable, which was shown as a boxplot in the descriptive statistics of Section 4.2.1. The importance of these theoretical distributions to reliability and survival analysis is well-known. The lognormal distribution, for instance, is a good fit for materials undergoing failure due to stress or corrosion.

For the high pressure CO₂ dataset, the normal distribution was found to be a closer fit than the Weibull distribution. This may partly be due to its small size. Unlike the low pressure CO₂ dataset, the corrosion rate response variable is approximately symmetrical and not positively skewed as shown in the boxplot in Section 4.3.1. It therefore does not lend itself to being modelled accurately by the Weibull family of distributions. The regression equations derived from Sections 4.2.3 and 4.3.3 were used in the Case Studies for the evaluation of the corrosion rate sensitivity plots, for the low and high pressure CO₂ corrosion datasets respectively.

For the low pressure CO₂ corrosion dataset, Case Study II was used to derive the corrosion rate sensitivity plot. This Case Study involved the use of best-fit distributions, which produced better statistics than the alternative Case Study I, for which Uniform distributions were assumed for the predictor variables: temperature CO₂ partial pressure, flow velocity and pH. Table 7-19 shows these results. Corrosion rates were proven to be most sensitive to the flow velocity, temperature, CO₂ partial pressure and pH, in that order. Flow velocities tend to have a great impact on low CO₂ corrosion because according to experimental findings, the ferrous carbonate (FeCO₃) scale formed is less-dense, less-compact and comparatively thinner than the scale formed under high pressure CO₂ conditions. The flow velocities under low pressure CO₂ conditions will have to be closely monitored to avoid a scenario where moderately high to very high flowrates sweep away protective scale on the metal surface, thus exposing the metal to further corrosivity.

For the high pressure CO₂ corrosion dataset, Case Study I, in which all variables were assumed to follow the Uniform distribution, outperformed Case Study II, in which the best-fit distributions were used to represent the predictor variables. This may be due to the small experimental dataset used as the basis for carrying out Monte Carlo simulations. Table 7-20 shows summary statistics for low and high pressure CO₂ corrosion datasets.

Table 7-20. Monte Carlo simulation results based on the derived regression equations

	Statistical Parameter	Experimental Corrosion Rate (mm/year)	Monte Carlo Corrosion Rate (mm/year)	Percentage Error (%)
Low Pressure CO₂ dataset	Mean	17.03	23.13	35.58
	Standard Deviation	17.02	10.79	36.57
High Pressure CO₂ dataset	Mean	10.78	10.84	0.56
	Standard Deviation	4.75	5.90	24.21

Corrosion rates for the high pressure CO₂ dataset were found to be most sensitive to temperature, followed by the CO₂ partial pressure, pH and flow velocity, in that order. The temperature and CO₂ partial pressures are influential variables regardless of the dataset. This is partly explained statistically by the PCA results of Sections 4.2.3 and 4.3.3. These variables have the greatest contributions to the variances of both datasets.

In terms of the corrosion mechanism, temperature affects corrosion rates because it speeds up the underlying transport and electrochemical processes. It is generally known that the rate constants of most reactions increase with an increase in temperature and corrosion is not an exception. This implies that where possible, temperatures will have to be kept low, to control corrosivity. A 5 or 10% increase in the temperature is roughly equivalent to a 2.5mm/year increment in corrosion rates. This is obtained from the sensitivity analysis plot in Section 7.5.2, Figure 7-17. The flow velocity has the least effect on corrosion due to the fact that the thicker scales under high pressure conditions diminish its effect. Table 7- 21 shows the least and most influential variables to corrosion rates for both low and high pressure CO₂ corrosion datasets.

Table 7-21. The least and most influential predictor variables on CO₂ corrosion

	Low Pressure CO₂ dataset	High Pressure CO₂ dataset
Least Influential Variable	pH	Flow velocity
Most Influential Variable	Flow velocity	Temperature

As with all attempts at modelling, there are usually limitations. The limitations of the sensitivity analysis results are that they are based on derived regression equations whose R^2 -values are moderately high, with adequate reliability but are not perfectly accurate models. The use of best-fit distributions as equivalent representations in Monte Carlo simulations can be somewhat simplistic, given that these theoretical distributions are not always representative of the experimental datasets and are capable of adding noise into simulations.

Also, the high pressure dataset is small in size therefore simulations based on the use of this dataset may not fully capture the intrinsic patterns as thoroughly as possible, such as when a sufficiently large dataset is used.

Chapter 8. Extended Discussion

8.1 Introduction

In Chapter 3, various CO₂ corrosion models were discussed including the empirical 1991, 1995 De Waard and Norsok correlations (Nyborg, 2002; Nyborg, 2010) as well as the mechanistic Freecorp model. Also, in Chapters 4 to 7, statistical, neural network, fuzzy, and Monte Carlo models were derived respectively.

In this Chapter, the models discussed in Chapter 3 are assessed using an identical experimental dataset used in determining the performances of the derived models of Chapters 4 to 7. Model performances were evaluated using the R^2 -statistic in conjunction with 95% confidence intervals. In addition, the applicability and limitations of the derived models from Chapters 4 to 7 are discussed.

8.2 Model Performances of the 1991, 1995 De Waard, Norsok and Freecorp

The test dataset was used to evaluate the performance of the models. The predicted outputs were plotted against their equivalent experimental corrosion results in the same manner as described for the previously discussed models.

8.2.1 Low Pressure CO₂ Corrosion Prediction Results

The 1991 De Waard Model

Figure 8-1 shows the correlation of the 1991 De Waard model results with experimental results. It is seen that the 1991 De Waard model has an average correlation with experimental results, as indicated by the R^2 -value. There is evidence of under-prediction – the data-points are consistently below the ‘experiment=model line’. The error bar chart of Figure 8-1 also indicates that under-prediction is highly prevalent, since all of the error bars are positive.

Also, all of the data-points lie outside of the 95% confidence interval. This is due to the fact that for this model, only temperature and P_{CO_2} variables are required with the other two variables, velocity and pH not represented. The model is therefore very limited hence the model-experiment plot indicates that the parameters are inadequately describing the data.

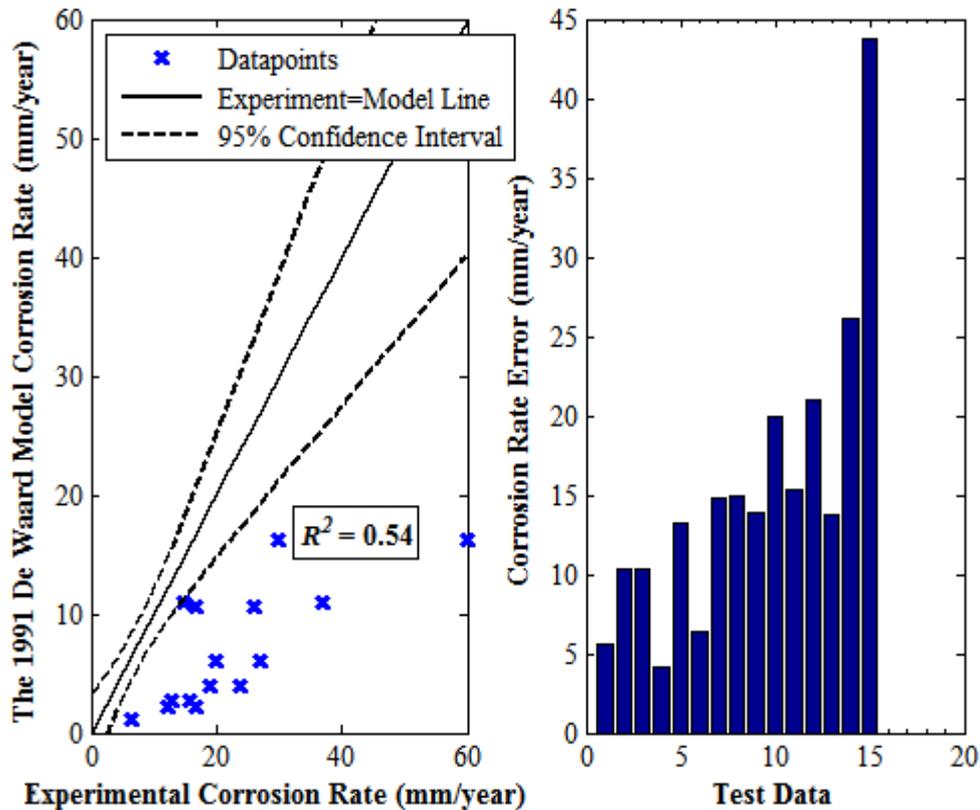


Figure 8-1. The 1991 De Waard model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate

The 1995 De Waard Model

Figure 8-2 shows the correlation of the 1995 De Waard model results with experimental results. It is seen that the 1995 De Waard model has a very strong correlation with experimental results, as indicated by the very high R^2 -value. Also, majority of the data-points lie within the 95% confidence interval, validating the reliability of the model. The interval range is not as large as the ranges associated with the other models, which implies the 1995 De Waard model has a greater degree of precision.

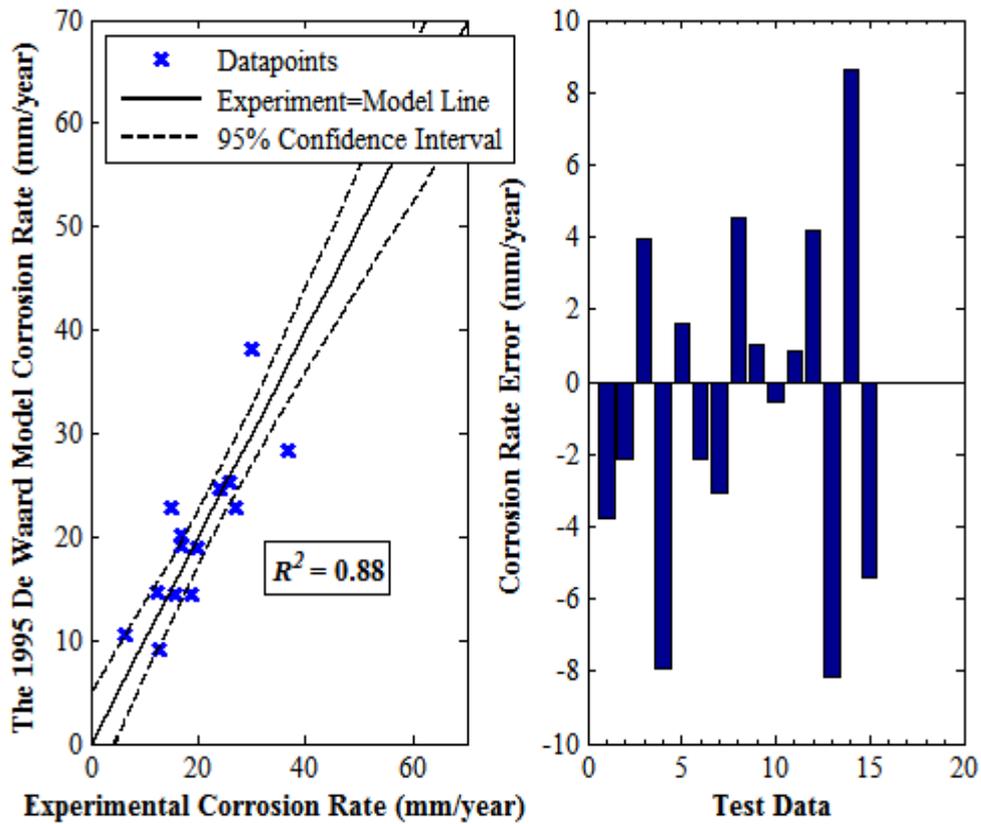


Figure 8-2. The 1995 De Waard model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data

Norsok Model

Figure 8-3 shows the correlation of the Norsok model results with experimental results. It is seen that the Norsok model has a very strong correlation with experimental results, as indicated by the very high R^2 -value even though there is evidence of over-prediction – the data-points are consistently above the ‘experiment=model line’.

Additionally, a significant number of data-points lie outside of the narrow-range 95% confidence interval indicating that though the model may have a strong correlation with experimental results, its reliability is satisfactory.

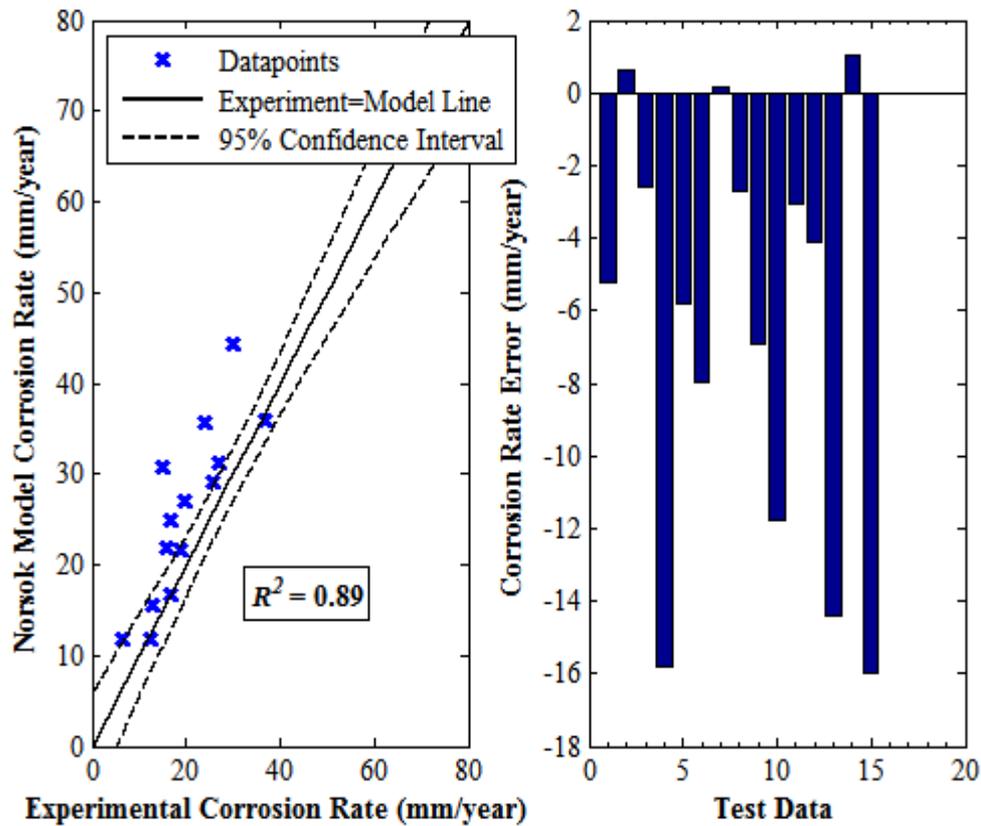


Figure 8-3. Norsok model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data

Freecorp Model

Figure 8-4 shows the correlation of the Freecorp model results with experimental results. It is seen that the Freecorp model has a rather weak correlation with experimental results, as indicated by the very low R^2 -value. The data-points appear to be divided into two sub-groups, where the low-magnitude corrosion rates are close together, under-predicted and the high-magnitude corrosion rates that are more scattered, yet evenly dispersed around the ‘experiment=model line’.

Also, a significant number of data-points lie outside of the wide-range 95% confidence interval. This information coupled with the fact that the R^2 -value is low implies that the model is not sufficiently accurate and reliable. The wide confidence bounds also indicate there is a greater degree of uncertainty in prediction results in comparison to other models.

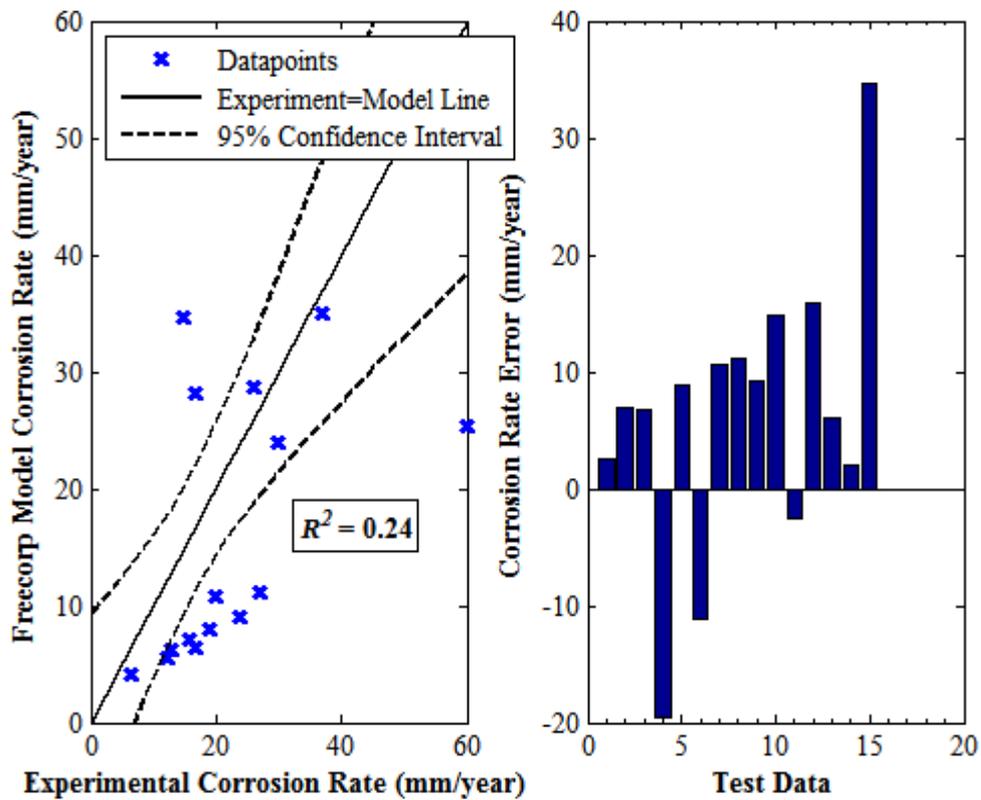


Figure 8-4. Freecorp model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model corrosion rate

8.2.2 High Pressure CO₂ Corrosion Prediction Results

The 1991 De Waard Model

The R^2 -coefficient for this model was very low when tested with the high pressure CO₂ corrosion test set. However this result is expected given that the model was originally formulated for predicting corrosion rates under low pressure. Figure 8-5 shows the correlation of the 1991 De Waard model results with the test dataset.

Also, the model over-predicted all test data-points indicating a high degree of conservatism. While the predictions are mostly within the 95% confidence bounds, the boundary limits are very wide indicating a high degree of uncertainty in predictions.

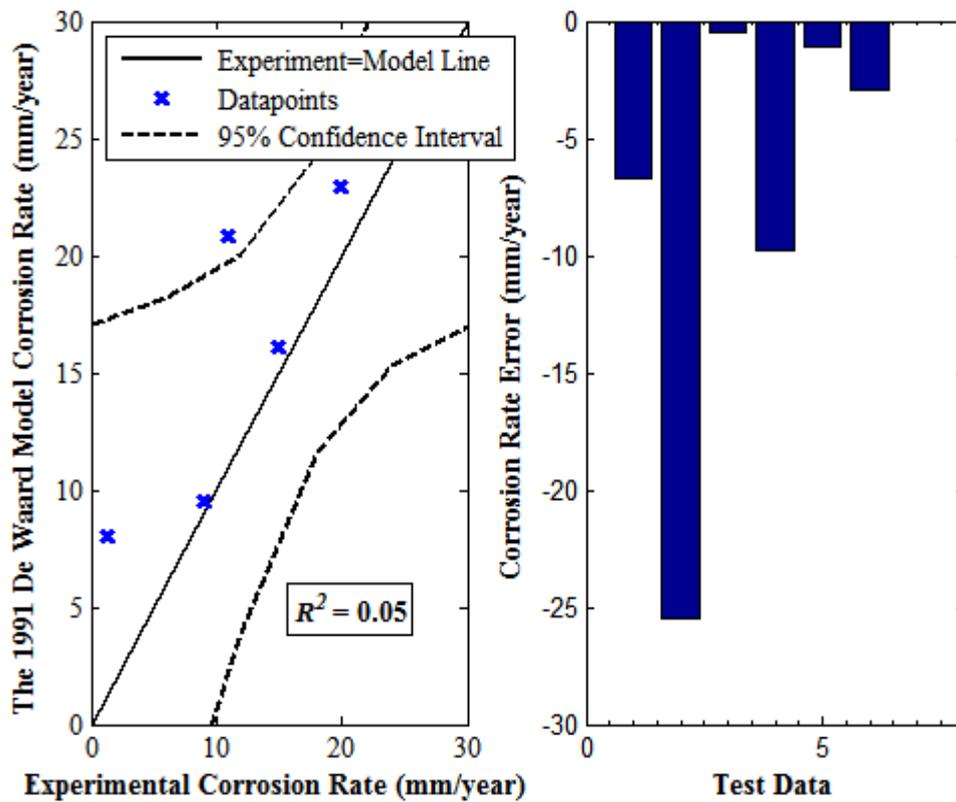


Figure 8-5. The 1991 De Waard model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data – error is the difference between experimental and model Corrosion rate

The 1995 De Waard Model

The R^2 -value for the 1995 DeWaard model is very low indicating a poor performance in prediction. Nevertheless, as with the earlier-published 1991 De Waard model, it was developed for low pressure CO₂ corrosion data.

The model vs experimental results, shown in Figure 8-6, is characterised by a very wide confidence interval. This implies a high degree of imprecision in prediction, even though most data-points lie within the confidence bounds. In addition, all test points with the exception of one is under-predicted.

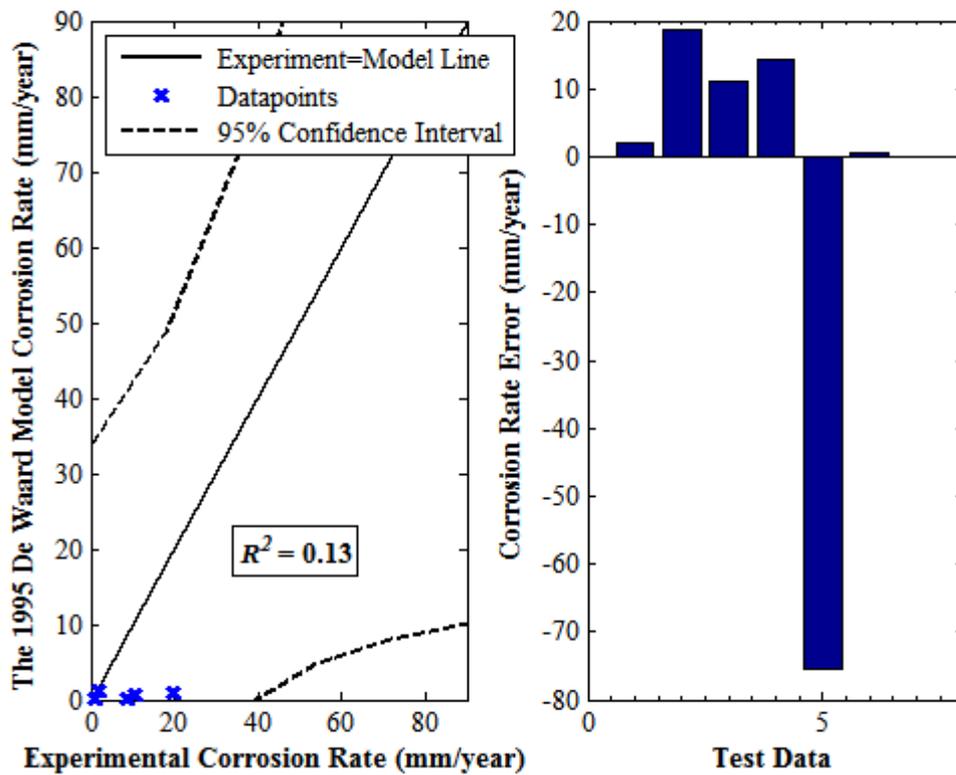


Figure 8-6. The 1995 De Waard model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data

Norsok Model

Figure 8-7 shows the correlation of Norsok model results with the test dataset. The R^2 -coefficient for the model predictions of high pressure CO₂ corrosion is low. The model was developed for predicting low pressure CO₂ corrosion thus struggles for this test dataset. The model under-predicts test set data hence it is not a conservative model for high pressure CO₂ data unlike the very conservative De Waard models. Though all predicted points lie within the 95% confidence bounds, it is not a reliable model given the wide range for the confidence intervals.

The Freecorp model was not discussed because for the high pressure dataset, it was very conservative leading to considerable over-predictions.

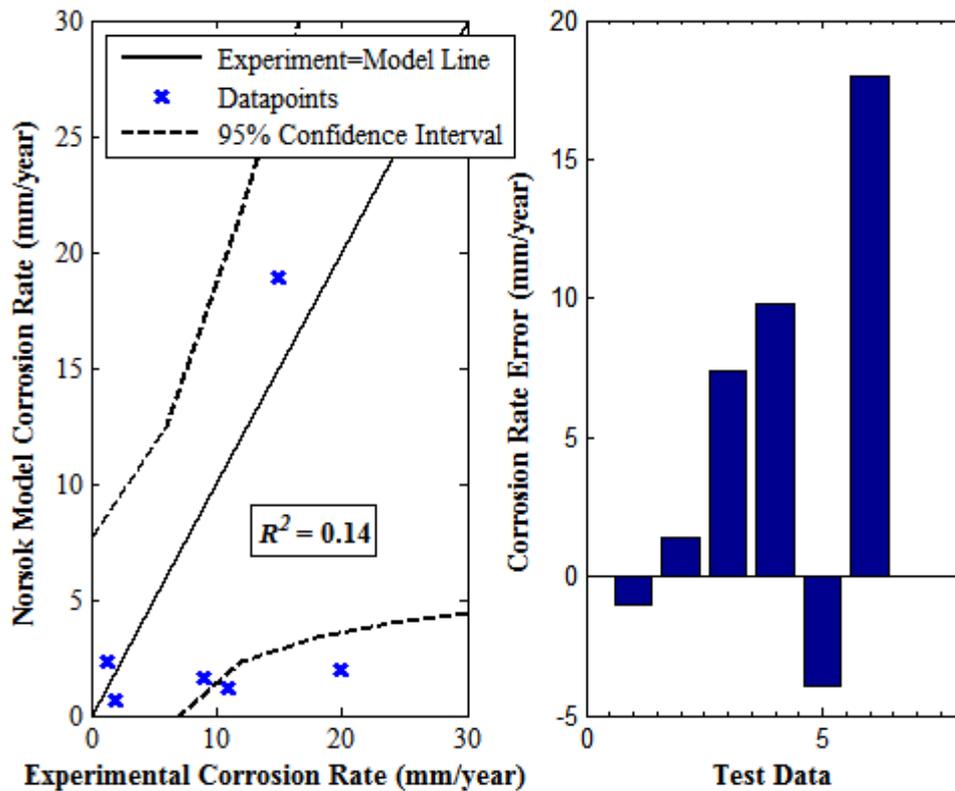


Figure 8-7. Norsok model corrosion rate against experimental corrosion rate plot and error bar chart for selected test data

8.3 Applicability and Limitations of the Various Derived Models

The derived models from Chapters 5 to 8 are the neural network, fuzzy, statistical and Monte Carlo simulations, respectively. These are discussed in the order in which they are listed in Section.

The derived NN models for both low and high pressure CO₂ datasets were highly accurate, in terms of their computed performances using the R^2 -statistic. In addition, the use of the tansig transfer function in conjunction with the trainlm training function produced the best results. This is because the trainlm being a quasi-Newton algorithm is known for offering excellent optimisation results. The size of the networks obtained were also a reflection of the size of the datasets being used for modelling. The low pressure dataset was bigger in size than the high pressure dataset and a network configuration of [10 5] was sufficient for obtaining accurate predictions. For the high pressure dataset, a network configuration of [5 5] produced desirable results. In general, NN models are very useful; however their performances are dependent on

the selection of the training data used. It is imperative that the training data adequately captures the trends within the overall data or is at least representative of the entire dataset being modelled.

The advantage of the NN models is that despite the small size of the high pressure CO₂ corrosion dataset, with the appropriate data division for training and testing coupled with pruning, high model accuracy was attained. It is capable of predicting corrosion rates to within ± 3 mm/year. They are somewhat robust and are adaptable to handle datasets of different sizes.

For the derived Mamdani FIS model, the obtained R^2 -coefficients varied markedly for the low and high pressure CO₂ corrosion datasets, as seen in the Conclusion Section of Chapter 6, Table 6-7. The reason for this is because the low pressure dataset is bigger and contains a greater degree of variation within it while the high pressure dataset is small and has much less variation within it. In general, both derived models are not accurate. There is a more even spread of test data-points about the experiment= model line for the low pressure CO₂ corrosion dataset, as seen in Section 6.5.1, Figure 6-12, than for the high pressure CO₂ corrosion dataset, as seen in Section 6.6.1, Figure 6-19.

Fuzzy models have been described as useful models that excel at filling banks or gaps of data which is either unknown or otherwise unavailable. They also excel at producing accurate profiles for the given predictor variables in a two or three-dimensional format. The plots of simulated corrosion rate against each of the predictor variables (temperature, CO₂ partial pressure, flow velocity and pH) showed realistic depictions of what is expected in an experimental setting. They have also been described as models that improve substantially in terms of performance when combined with NN models, forming neuro-fuzzy models (Hajizadeh, 2007). This model aspect is beyond the scope of this thesis.

For the described statistical models, the multiple linear regression equations were moderately accurate for both low and high pressure CO₂ corrosion datasets. They also produced good scatters for test data-points around the experiment-model line on the Model vs experimental plots, though with a slight degree of over-prediction for both low (Section 4.2.3, Figure 4-7) and high pressure CO₂ corrosion datasets (Section 4.3.3, Figure 4-21) respectively.

For the nonlinear regression equations, the performances were markedly different for the datasets. While the R^2 -coefficient obtained for the low pressure dataset was moderately high

($R^2=0.67$), that of the high pressure dataset was very low ($R^2=0.10$). The reason for this is because nonlinear regression modelling involves the application of numerical techniques and while the dataset is sufficiently large for the low pressure dataset, the high pressure dataset is quite small. Therefore, there was a lack of convergence during numerical computation. This manifests itself in the model vs experiment line, shown as wide 95% confidence bounds, as seen in Section 4.3.4, Figure 4-25. This implies that the results are imprecise and unreliable. The multiple linear regression models are versatile and lend themselves for further statistical analysis such as Monte Carlo simulations.

The statistical models have a drawback, which is the size of the dataset being modelled. They generally require sizeable amounts of data in comparison to NN models in order to attain high performance levels in terms of accuracy. The advantage they have over the NN models is that relationships between predictor variables are inferred. A given response variable can be expressed explicitly in terms of its predictor variables, when parametric modelling is carried out, as was done in this thesis, unlike the ‘black-box’ approach of the NN models.

Monte Carlo simulations use random sampling and statistical modelling to estimate statistical parameters or mathematical functions of interest. This technique is therefore an extension of statistical analysis. The advantage of this technique is that it offers a solution for modelling data-points as a ‘collective’. For instance, low pressure corrosion rates were found to come from a Weibull family of distributions which includes the exponential distribution as a special case. Though, the lognormal distribution was also found to fit the data, after a 100,000 Monte Carlo simulations, the best-fit distribution was the exponential; with mean and standard deviation percentage errors given as: 0.00 and 0.06%.

The Weibull, exponential and lognormal distributions are extensively applied in reliability and survival analysis as they accurately model naturally-occurring phenomena such as corrosion and wear accurately. For the high pressure CO₂ dataset, it is not as clear-cut which distribution is a best-fit. However, the logistic, normal and Weibull distributions were good solutions. After carrying out Monte Carlo simulations, the normal proved to be the best fit, owing to the central limit theorem (Rice, 2007). The best-fit percentage errors were within 0.09 and 0.21% of the mean and standard deviation respectively.

While these inferences are very useful, they still do not provide insight as to how the various predictor variables affect the response variable: corrosion rate. The derived regression equations enabled the evaluation of the corrosion rate sensitivity with respect to each of the

predictor variables. The flow velocity and pH were found to be the most and least influential predictor variables respectively on corrosion rate, for the low pressure CO₂ corrosion dataset. Also, the temperature and flow velocity were found to be most and least influential predictor variables respectively on corrosion rate for the high pressure CO₂ corrosion dataset.

Overall, it was challenging to model the low pressure data due to a high degree of variation. The presence of outliers in this dataset causes skewness which the models struggled to cope with thereby increasing the errors in prediction and lowering their accuracy. For the high pressure dataset, variation within the dataset did not prove problematic in modelling as the dataset was small however because of the few data-points, the nonlinear regression model, for instance was unable to converge to a solution. This is because the model applies numerical techniques to compute the solution which relies on the use of a large dataset to be more effective. Table 8-1 is summary of the results obtained for the derived models.

Table 8-1. R^2 -values for the derived models

Model	Low Pressure CO₂ Dataset	High Pressure CO₂ Dataset
Neural Network	0.91	0.91
Mamdani FIS	0.32	0.63
Multiple Linear Regression	0.65	0.70
Nonlinear Regression	0.67	0.10

8.4 Concluding Remarks on the Performances of each of the Models

The empirical Norsok formulation has a strong positive correlation with experimental results for the given test dataset while the mechanistic Freecorp model has a weak positive correlation. The Norsok formulation shows clear evidence of over-prediction but it must be borne in mind that since CO₂ corrosion mitigation is the principal aim in modelling, it follows that over-prediction is more favourable than under-prediction. The Freecorp model predicts the test dataset corrosion rate inaccurately, producing two separate trends on the model vs experiment graph, of closely packed low-magnitude corrosion rate values and a more scattered high-magnitude corrosion rate values.

The 1991 De Waard model with an R-squared coefficient of 0.54 showed a positive correlation between model and experimental results however it generally under-predicted the test dataset. This inadequate representation of the data is apparently due to the correlation only having two predictor variables: P_{CO_2} and temperature and as such misses out completely

on the effects of the other two missing variables on the data. The 1995 De Waard model performed better than each of all the other models. In addition to attaining a high R^2 -value, all the test data-points are in very close proximity to the 95% confidence bounds.

The Freecorp model has a low R^2 -coefficient, even for the low pressure CO₂ test dataset owing to its lack of scaling modelling in its predictions. As a result, it has a great tendency to over-predict. Predictions can be partially split into two groups, as seen in Figure 8-4: test data-points less than 25mm/year and test data-points greater than 25mm/year. Model performances are summarised in Table 8-1.

For high pressures, all the established models performed poorly, as indicated by the low R^2 -values in Table 8-2. This is the case because all these models were developed with the sole purpose of predicting corrosion under low pressure CO₂ conditions, characteristic of oil and gas pipelines. The wide 95% confidence bounds for all models indicate that they are imprecise and model predictions are unreliable.

Table 8-2. R^2 -values for the established models

Model	Low Pressure CO₂ Dataset	High Pressure CO₂ Dataset
The 1991 De Waard	0.54	0.05
The 1995 De Waard	0.88	0.13
Norsok	0.89	0.14
Freecorp	0.24	(-)*

*The Freecorp model R^2 -value was not computed for the high pressure CO₂ test dataset because it was excessively conservative which led to massive over-predictions

Chapter 9. Conclusions and Recommendations for Further Work

9.1 Conclusions

The effect of carbon dioxide (CO₂) corrosion on pipelines is of great relevance to the petroleum as well as the CCS industries. CO₂ corrosion is responsible for lost production as it brings about the gradual degradation of pipe internals with time. CO₂ corrosion was discovered in gas wells in Texas in the 1940's and ever since there have been multiple research undertakings on the topic.

The cost of corrosion cannot be overstated - general corrosion costs are said to be between 3 to 5% of an industrialised nation's gross domestic product. The cost of CO₂ corrosion in the U.S. oil and gas industry as of 2014 was said to be ~\$18 billion. Due to the negative impact of current global emissions of the gas on the World climate, geo-sequestration efforts are currently being explored by the CCS industries which involve the use of abandoned oilfields and coal mines for the storage of the gas for geologically significant timescales.

CO₂ as a dry gas is non-corrosive at oilfield and CCS conditions, however when it becomes dissolved in water, it forms carbonic acid, which is corrosive. CO₂ corrosion potentially poses a greater risk when it manifests in the localised mesa-form as opposed to uniform corrosion. In mesa-type corrosion, the medium velocity fluid flow coupled with the 'right' fluid composition gives rise to localised corrosive attack on the metal surface. Also the iron carbide (Fe₃C) scale is prevalent in this form of corrosion. Unlike the ferrous carbonate (FeCO₃) scale, Fe₃C provides less of a protective barrier to corrosion for the metal surface beneath.

These forms of corrosion risk coupled with the aforementioned costs necessitate the importance of understanding the core concepts underpinning CO₂ corrosion hence the need for reliable corrosion rate prediction models. Various models have been developed over the years for predicting corrosion rates in pipelines. Some of these include the 1975, 1991 and 1995 De Waard, Norsok and Freecorp models. These models were developed for application in the oil and gas setting, known for their characteristically low CO₂ partial pressures, typically less than 2MPa (20bar). While their prediction performances are good for low CO₂ partial pressure conditions, their performances for high CO₂ pressure environments have been shown to be poor.

Separate low and high pressure CO₂ datasets obtained from research literature were used to evaluate model performance in terms of their accuracy in prediction. The statistical measure used is the R^2 -coefficient, with a minimum value of 0 and maximum value of 1, for a positive correlation between the test dataset and model predictions.

Various models were derived based on neural networks (NN), fuzzy, statistical and Monte Carlo simulations. These will be discussed in the order in which they are listed. Neural networks are inspired by the operation of the human brain and its neurons in the nervous system. For modelling CO₂ corrosion, performance-related tests were carried out on Matlab 2012a transfer and training functions.

The derived NN models for the low and high pressure CO₂ datasets produced very good results with high R^2 -values of ~0.9 and the narrow 95% confidence bounds indicated model precision and a great certainty in predictions. All predictions were within ± 3 mm/year of the experimental data. In general, NN models are particularly useful when the size of the dataset is small, which is currently the case for high pressure CO₂ corrosion. They are adaptable and are capable of producing excellent results for very small datasets however their drawback is in their need for a good user-selected training set. Without a training set that captures the variation sufficiently enough, results may not be forthcoming.

Fuzzy Inference Systems (FIS) or fuzzy logic systems are based on binary systems, but where binary systems deal with extremes (0 or 1); fuzzy logic systems operate in the 'fuzzy' region: 0-1. In terms of performance, the Mamdani FIS attained a higher R^2 -value for the high pressure CO₂ test set than for the low pressure CO₂ test set. The reason the model struggled in model prediction, particularly for the low pressure CO₂ dataset might be due to a greater degree of variation within this dataset than with the comparatively smaller high pressure test set. The models do however excel at modelling the corrosion rate vs predictor variable profiles realistically

The statistical models were derived for both low and high pressure CO₂ corrosion datasets. Firstly, principal component analysis (PCA) was performed on both datasets. The PCA results reveal the relative contribution of each of the predictor variables to the response variable. The analysis revealed that the temperature and CO₂ partial pressure had the greatest share in terms of the contribution to the variance of both low and high pressure datasets. The multiple linear regression equations were obtained with moderately high R^2 -values. For the high pressure CO₂ dataset, there was a tendency for over-prediction indicating some degree of conservatism.

The Monte Carlo model performances were evaluated by estimating the means and standard deviations of the test dataset as well as for the derived models, and computing the errors between these and expressing them as a percentage error.

The low and high pressure datasets were analysed to find their best-fit theoretical distributions. The low pressure CO₂ dataset was found to match-up well against the Weibull family of distribution. This is because the corrosion rates in the low pressure CO₂ dataset exhibit positive skewness. The outliers within this dataset are in the 60-70mm/year range. The lognormal, Weibull and exponential distributions are good models for naturally-occurring phenomena such as metals undergoing wear and corrosion. They are also good fits for positively skewed data. The high pressure dataset was found to follow the normal distribution although this is not as clear-cut as the low pressure case. This is attributed to the small size of the dataset. The regression equations were applied in the derivation of the corrosion rate sensitivity. Since the regression equations contain predictor variable terms, Monte Carlo simulations were used to determine the variables that the corrosion rate is most sensitive to.

The sensitivity analysis for the low pressure CO₂ data revealed that corrosion rates are most sensitive to flow velocity, closely followed by the temperature and CO₂ partial pressure and least sensitive to the pH. The result implied that the flowrate for the low CO₂ pressure dataset has to be closely monitored as for instance, an initial 10% increment in the standard deviation of the flow velocity results in a corresponding increase in corrosion rates by 5mm/year.

For high pressures, it was observed that the temperature is the most influential variable on corrosion rate followed by the CO₂ partial pressure, pH and flow velocity, in that order. Interestingly, results from PCA revealed the temperature and CO₂ partial pressure to be the predictor variables that had the greatest contribution to the variance of the dataset. This is true for the both low and high pressure datasets. For both datasets, corrosion rates are consistently sensitive to these variables.

Also, as previously stated, temperatures increase corrosion rates as explained by the Kinetic theory, and so is significant. At high pressure conditions, thicker, more compact and denser FeCO₃ scales are formed on the metal surface which offers greater resistance to the sweeping effect of fluid flowrates. Under low pressure CO₂ conditions, FeCO₃ scales are thinner and less dense therefore are more vulnerable to fluid flow effects. These can be easily washed off thus exposing the metal surface beneath to further corrosivity.

The limitations of the Monte Carlo sensitivity analyses are that they are based on fairly accurate regression equations however they are not flawless and do not completely capture the variability in the datasets. Also, in some cases, the distributions selected as best-fit were the best-match distributions, which do not necessarily mean that they describe every data-point

within the predictor variable dataset accurately. For some predictor variables, the best-fit distributions fit the central set of data-points well but perform relatively poorly at the tail-points (very low and very high-magnitude data). In these instances, the assumptions for best-fit distributions introduce more uncertainties to further modelling.

9.2 Recommendations for Further Work

In this thesis, a Mamdani FIS model was derived for the purpose of CO₂ corrosion prediction for both low and high partial pressures of CO₂. However, a Sugeno FIS model was not developed for low and high CO₂ partial pressures. Also, due to the fact that the derived models had limited accuracy, which is somewhat a feature of these models when used as stand-alone tools, a neuro-fuzzy model can be developed. This model would combine the good characteristics of both the NN and fuzzy models, which can potentially result in a more robust tool for corrosion prediction.

Appendices

A1.1 Project Gantt Chart

Years →		2012				2013				2014				2015			
Months →		Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec
Project Tasks ↓	Sub-Project Tasks ↓																
Research Proposal			I				I, II				I, II				I		
Literature Review																	
Database Development	CO ₂ + Other additives (H ₂ S, acetic acid)																
	CO ₂ corrosion at elevated pressures																
Model Development	Fugacity/Co-solubility relationships of CO ₂ , H ₂ O + other additives																
	Investigating weighting of input parameters in neural network modelling																
	GUI Development (Graphic User Interface)																
	Fuzzy logic modelling																
	Current CO ₂ models, NORSOK model																
	Statistical models + Monte Carlo Simulations																
Testing & Iterations																	
Comparison of results with corrosion rate data from Published literature																	
Writing up of Thesis																	A
Milestones:		Key:															
I = Postgraduate Researchers (PGR) Conference		Primary activity															
II = Marine Technology Postgraduate Conference (MTPC)		Secondary activity															
A = Journal Paper Submission/Presentation - SPE, NACE, Supercritical Fluids, Greenhouse Gas Control International journals																	

A1.2 Microsoft Excel Corrosion Database - Sample

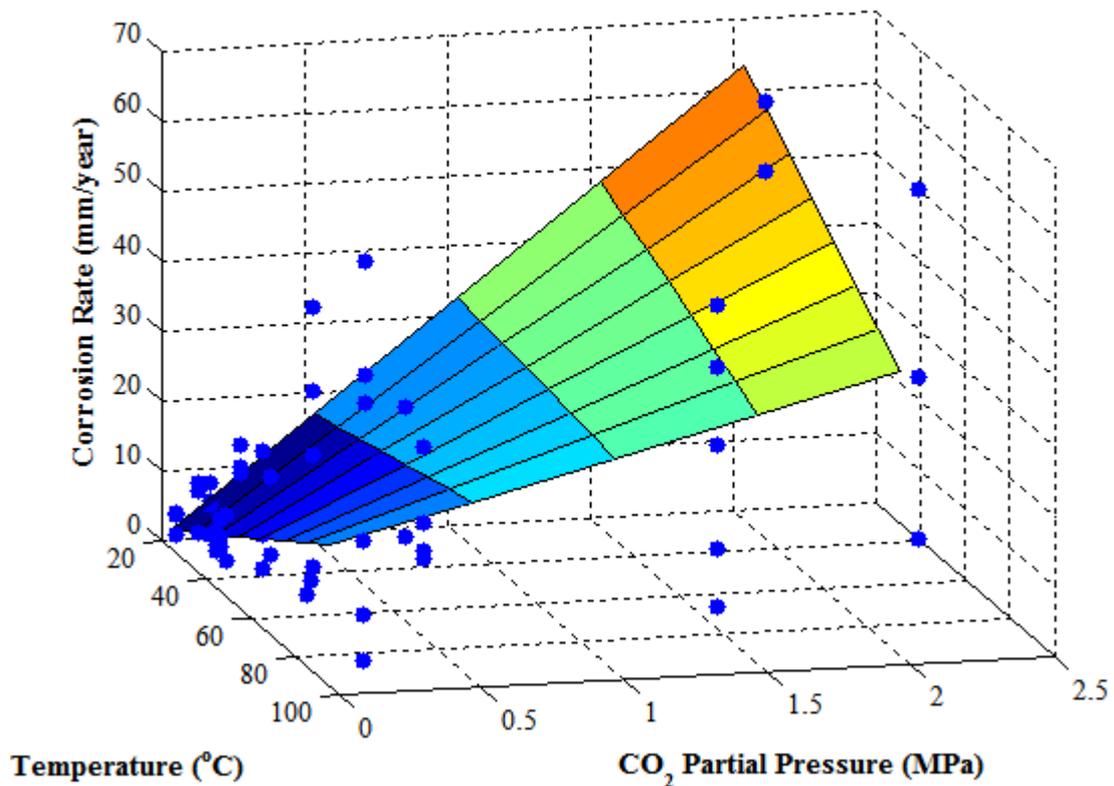
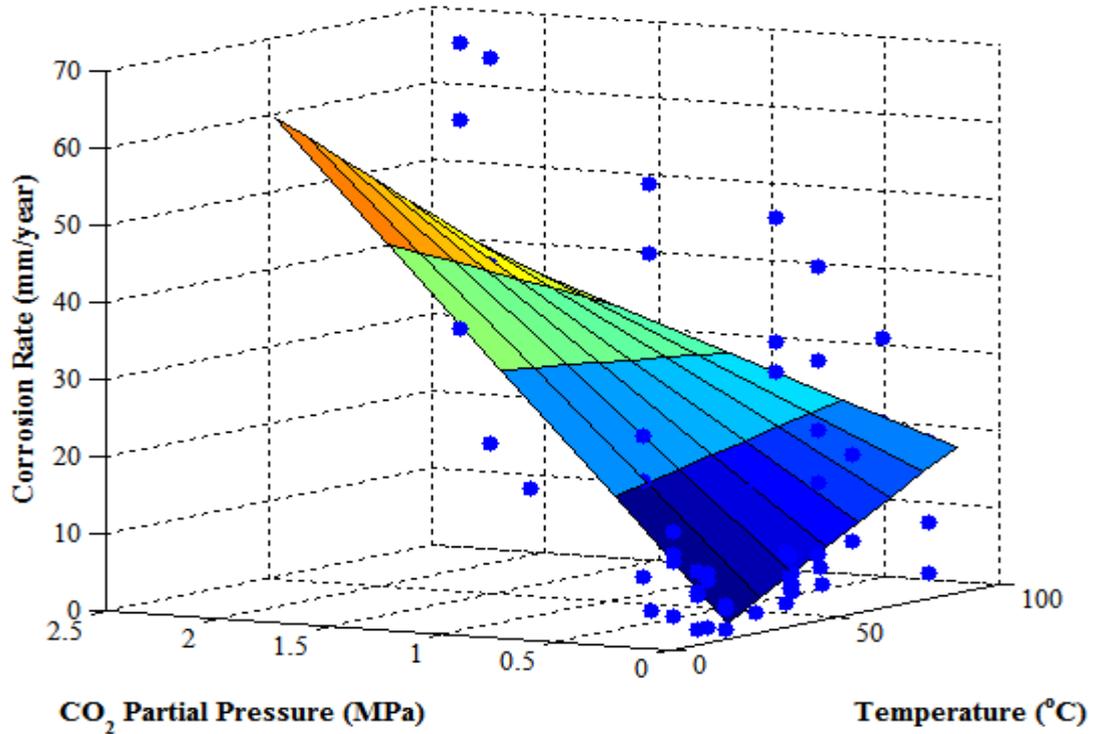
Temperature (°C)	P_{CO_2} (MPa)	Velocity (m/s)	pH	Corrosion Rate (mm/year)	Data Source	Temperature (°C)	P_{CO_2} (MPa)	Velocity (m/s)	pH	Corrosion Rate (mm/year)
20	0.05	0.1	4.1	0.7	Dugstad et al., 1994	20	0.05	0.1	4.1	0.7
20	0.13	0.1	3.8	0.9	Halvorsen et al., 1999	20	0.13	0.1	3.8	0.9
20	0.17	0.1	3.7	0.5	Nordsveen et al., 2003	20	0.17	0.1	3.7	0.5
20	0.28	0.1	3.7	1.9		20	0.28	0.1	3.7	1.9
20	0.38	0.1	3.7	2.5		20	0.38	0.1	3.7	2.5
20	0.05	13	4.1	3.5	Zhang et al., 2012a	20	0.05	13	4.1	3.5
20	0.05	8.5	4.1	4	Hesjevik et al., 2010	20	0.05	8.5	4.1	4
20	0.28	13	3.7	13		20	0.28	13	3.7	13
20	0.28	8.5	3.7	10		20	0.28	8.5	3.7	10
20	0.28	3.1	3.7	9		20	0.28	3.1	3.7	9
20	0.13	13	3.8	7		20	0.13	13	3.8	7
20	0.13	8.5	3.8	8		20	0.13	8.5	3.8	8
20	0.17	3.1	3.7	4.8		20	0.17	3.1	3.7	4.8
20	0.17	8.5	3.7	5.4		20	0.17	8.5	3.7	5.4
20	0.17	13	3.7	8		20	0.17	13	3.7	8
40	0.2	3.1	3.8	1		20	0.205	1.13	4.8	2.5
40	0.2	8.5	3.8	6		20	0.225	1.13	4.7	3
40	0.04	8.5	4.15	5		20	0.305	2.25	4.6	6

A4.1 Statistical Student *t*-table

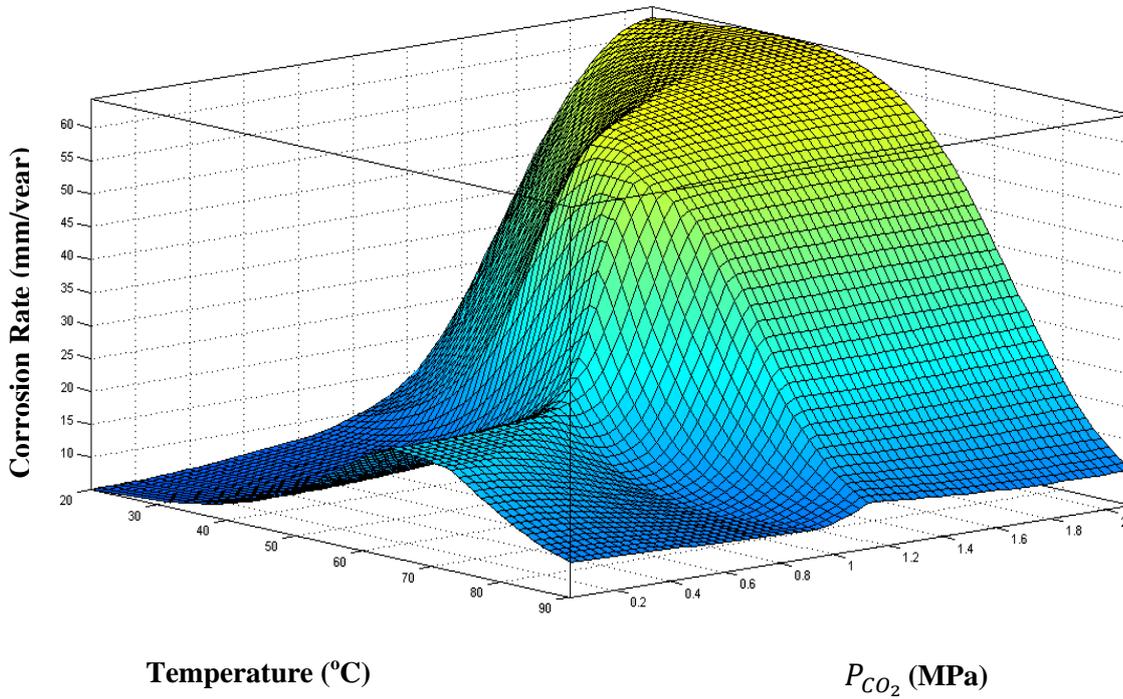
***t* Table**

cum. prob	<i>t</i> _{.50}	<i>t</i> _{.75}	<i>t</i> _{.80}	<i>t</i> _{.85}	<i>t</i> _{.90}	<i>t</i> _{.95}	<i>t</i> _{.975}	<i>t</i> _{.99}	<i>t</i> _{.995}	<i>t</i> _{.999}	<i>t</i> _{.9995}
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
Z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

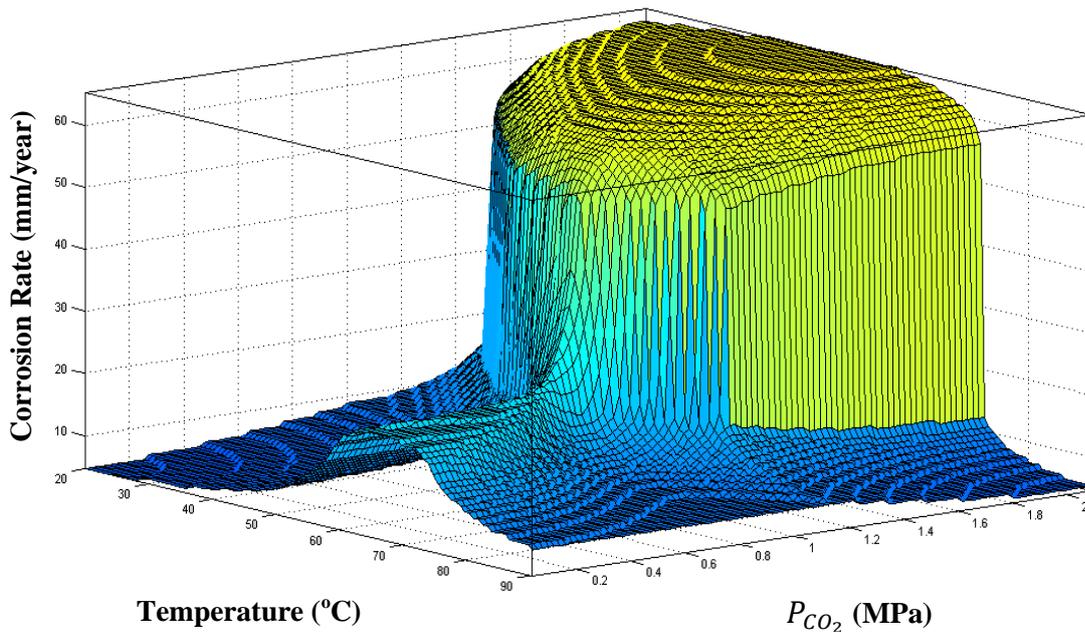
A4.2 Three-dimensional Plot of Corrosion Rate against CO₂ Partial Pressure and Temperature for the Low Pressure CO₂ Corrosion Dataset from Alternate Viewing Angles



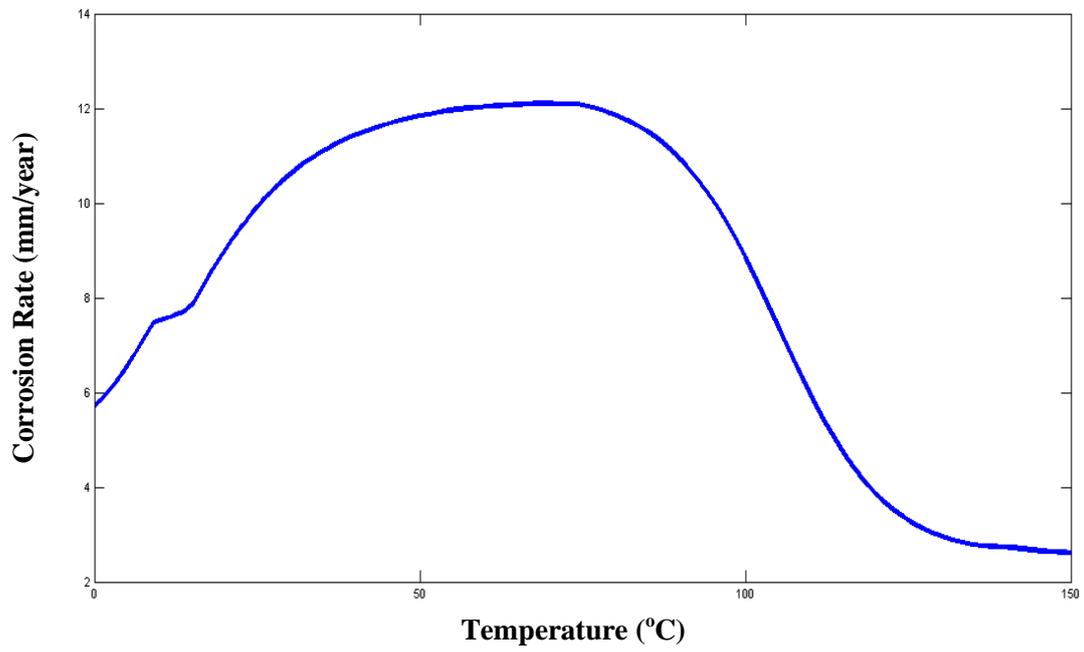
A6.1a Three-dimensional Mamdani FIS Plot of Corrosion Rate against Temperature and CO₂ partial pressure for the Low Pressure CO₂ Dataset (Centroid Defuzzification)



A6.1b Three-dimensional Mamdani FIS Plot of Corrosion Rate against Temperature and CO₂ partial pressure for the Low Pressure CO₂ Dataset (Bisector Defuzzification)



A6.2 Mamdani FIS Plot of Corrosion Rate against Temperature for the High Pressure CO₂ Dataset



Developed Model Code

Regression Model Code

%Script to analyse Low Pressure CO₂ corrosion data from Dugstad 1994 journal article using multiple linear

%regression and non-linear regression

clc, clear, close all

%The data in columns are low pressure CO₂ corrosion data from Dugstad 1994

%First column => Temperature (^oC)

%Second column => CO₂ Pressure (MPa)

%Third column => Velocity/flow (m/s)

%Fourth column => pH

%Fifth column => CO₂ Corrosion rate (mm/year)

Dugstad_1994_data_01=...

20.0000	0.0500	0.1000	4.1000	0.7000
20.0000	0.1300	0.1000	3.8000	0.9000
20.0000	0.1700	0.1000	3.7000	0.5000
20.0000	0.2800	0.1000	3.7000	1.9000
20.0000	0.3800	0.1000	3.7000	2.5000
20.0000	0.0500	13.0000	4.1000	3.5000
20.0000	0.0500	8.5000	4.1000	4.0000
20.0000	0.2800	13.0000	3.7000	13.0000
20.0000	0.2800	8.5000	3.7000	10.0000
20.0000	0.2800	3.1000	3.7000	9.0000
20.0000	0.1300	13.0000	3.8000	7.0000
20.0000	0.1300	8.5000	3.8000	8.0000
20.0000	0.1700	3.1000	3.7000	4.8000
20.0000	0.1700	8.5000	3.7000	5.4000
20.0000	0.1700	13.0000	3.7000	8.0000
40.0000	0.2000	3.1000	3.8000	1.0000
40.0000	0.2000	8.5000	3.8000	6.0000
40.0000	0.2000	13.0000	3.8000	18.0000
40.0000	0.0400	3.1000	4.1500	4.0000
40.0000	0.0400	8.5000	4.1500	5.0000
40.0000	0.0400	13.0000	4.1500	6.5000
40.0000	0.0530	3.1000	4.1000	4.5000
40.0000	0.0530	8.5000	4.1000	6.0000
40.0000	0.0530	13.0000	4.1000	8.5000
40.0000	0.0700	8.5000	4.0500	2.5000
40.0000	0.0700	13.0000	4.0500	9.0000
40.0000	0.7000	3.1000	3.6000	4.6000
40.0000	0.7000	8.5000	3.6000	17.0000
40.0000	0.7000	13.0000	3.6000	23.0000
40.0000	1.2000	8.5000	3.4000	15.0000
60.0000	0.2000	0.1000	3.8000	3.0000
60.0000	0.2100	0.1000	3.8000	5.0000

60.0000	0.2200	0.1000	3.8000	6.8000
60.0000	0.2200	3.1000	3.8000	16.0000
60.0000	0.2200	8.5000	3.8000	23.0000
60.0000	0.2200	13.0000	3.8000	32.0000
60.0000	0.2200	13.0000	3.8000	44.0000
60.0000	1.8000	3.1000	3.4000	33.0000
60.0000	1.8000	8.5000	3.4000	60.0000
60.0000	1.8000	13.0000	3.4000	70.0000
60.0000	0.0700	3.1000	4.1500	8.8000
60.0000	0.0700	13.0000	4.1500	20.0000
60.0000	0.4000	3.1000	3.7000	30.0000
60.0000	0.4000	8.5000	3.7000	34.0000
60.0000	0.4000	13.0000	3.7000	50.0000
90.0000	0.1600	3.1000	4.1000	1.9000
90.0000	0.1600	8.5000	4.1000	8.5000
90.0000	0.1600	13.0000	4.1000	19.0000
90.0000	0.3700	3.1000	3.8000	16.0000
90.0000	0.3700	3.1000	3.8000	17.0000
90.0000	0.3700	8.5000	3.8000	21.0000
90.0000	0.3700	13.0000	3.8000	32.0000
90.0000	1.4000	3.1000	3.6000	6.8000
90.0000	1.4000	3.1000	3.6000	15.0000
90.0000	1.4000	8.5000	3.6000	30.0000
90.0000	1.4000	13.0000	3.6000	41.0000
90.0000	1.4000	8.5000	3.6000	50.0000
90.0000	2.1000	3.1000	3.5000	15.0000
90.0000	2.1000	3.1000	3.5000	38.0000
90.0000	2.1000	13.0000	3.5000	65.0000];

%'Temperature ' 'CO_2 Partial Pressure (MPa)' ' Flowrate(m/s) ' 'pH
' 'Corrosion Rate (mm/year) '');

X_01=Dugstad_1994_data_01(:,1:3);

x_01=Dugstad_1994_data_01(:,1);

%x_01, x_02, x_03 and x_04 are column matrices comprising temp, pressure, velocity and
%pH values respectively

x_02=Dugstad_1994_data_01(:,2);

x_03=Dugstad_1994_data_01(:,3);

x_04=Dugstad_1994_data_01(:,4);

y_01=Dugstad_1994_data_01(:,5);

%y_01 represents a column matrix comprising the Corrosion rate values

b_1=0.7;

b_2=1.9;

b_3=2.4;

b_4=2.1;

b_5=1.92;

beta_01=[b_1 b_2 b_3 b_4 b_5];

beta_02=[b_1; b_2; b_3; b_4; b_5];

```
%beta is a matrix of five values - containing the values of b_1,  
%b_2, b_3, b_4 and b_5 above  
  
%rate_01=Sc_data_01_Fuzzy_data(:,3);  
  
%nlintool(x_01,y_01,@model_fun_ScCO2_01)  
  
%xname={'Temperature (oC)' 'CO2 Partial Pressure, PCO2 (MPa)' 'Flowrate (m/s)'};  
%yname={'Corrosion Rate (mm/year)'};  
  
%nlintool(X_01,y_01,@hougen,beta_01,0.01,xname,yname)  
%Using the Hougen non-linear system to analyse this problem  
%X_01 is a matrix comprising three columns: x_01, x_02 and x_03  
  
%X_001=lsqcurvefit(@hougen,[1],X_01,y_01);  
  
%betahat_01=nlinfit(X_01,y_01,@hougen,beta_02);  
  
%[betahat_01,resid,J]=nlinfit(X_01,y_01,@hougen,beta_02);  
%betaci=nlparci(betahat_01,resid,J);  
  
%--using rstool--%  
X_02=Dugstad_1994_data_01(:,1:4);  
X_04=[x_01 x_02 x_04];  
%X_02 is a matrix containing all the input variables for Temperature,  
%P_CO2, Velocity and pH and will be used with the rstool  
  
%beta_star=randn(nVars,1);  
  
mdl_10=NonLinearModel.fit(X_01,y_01,@hougen,beta_02);  
plotSlice(mdl_10)  
  
figure('Name','Nonlinear Regression Model Case Order Diagnostic Plots - Leverage and  
Residuals')  
  
subplot(1,2,1)  
theta_01=plotDiagnostics(mdl_10,'leverage');  
set(theta_01,'LineWidth',2.5,'MarkerSize',8.0)  
set(gca,'FontName','Times New Roman')  
title('')  
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold')  
ylabel('Leverage','FontWeight','Bold')  
  
%--Original labels for the plot--%  
%title('Case Order Plot of Leverage')  
%xlabel('Row Number')  
%ylabel('Leverage')
```

```
subplot(1,2,2)
theta_02=plotResiduals mdl_10,'fitted');
set(theta_02,'LineWidth',2.5,'MarkerSize',8.0)
set(gca,'FontName','Times New Roman')
title('')
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold')
ylabel('Residuals','FontWeight','Bold')

%--Original labels for the plot--%
%title('Plot of Residuals vs Fitted Values')
%xlabel('Fitted Values')
%ylabel('Residuals')

display mdl_10

%--mdl_11 is a nonlinear model identical to mdl_10 but with a pH variable in
%place of the velocity variable in order to ascertain that mdl_10 with the
%velocity variable has a higher r-squared value
mdl_11=NonLinearModel.fit(X_04,y_01,@hougen,beta_02);
plotSlice(mdl_11)

figure()

subplot(1,2,1)
plotDiagnostics(mdl_11,'leverage')

subplot(1,2,2)
plotResiduals(mdl_11,'fitted')
%mdl_11=NonLinearModel.fit(X_02,y_01,@hougen,beta_02);

display(mdl_11)

%These next two lines for carrying out the regression analysis without errors%
ones_01=ones(60,1);
X_03=[X_02 ones_01];

xname_01={'Temperature' 'PCO2' 'Flowrate (m/s)' 'pH'};
yname={'Corrosion Rate'};

rstool(X_01,y_01,'interaction'); %For Temp, P_C_O_2 and Vel variables

rstool(X_02,y_01,'linear');% ,xname_01,yname)
%For all four variables

%rstool(X_02,y_01,{'Temperature' 'PCO2' 'Flowrate (m/s)' 'pH','Corrosion Rate
(mm/year)'})
```

`%--Variable Interaction plots--%`

```
%mdl_01=fitlm('y_01 ~ x_01 + x_02 + x_03 + x_04 + x_01:x_02:x_03:x_04');
```

```
[b,bint,r,rint,stats] = regress(y_01,X_03);
```

```
%mdl_01=stepwise(X_02,y_01);
```

`%--Linear Model Fit--%`

```
mdl_02=LinearModel.fit(X_02,y_01);  
display (mdl_02)
```

```
%mdl_03=LinearModel.fit(X_02,y_01,'interactions');  
%display (mdl_03)
```

```
%plotEffects(mdl_03);
```

```
d_01=dataset(x_01,x_02,x_03,x_04,y_01);
```

```
mdl_04=LinearModel.fit(d_01,'y_01 ~ x_01 + x_02 + x_03 + x_04 + x_01:x_02:x_03:x_04 +  
x_01^2');
```

`%--Interaction Plots for all the Variables (Using 'predictions')--%`

```
figure('Name','Interaction Plots of Temperature with CO2 Partial Pressure and Temperature  
with Velocity')
```

```
subplot(1,2,1)
```

```
kappa_01=plotInteraction(mdl_04,'x_01','x_02','predictions');
```

```
set(gca,'FontName','Times New Roman')
```

```
set(kappa_01,'LineWidth',2.5)
```

```
xlabel('CO2 Partial Pressure,  $P_{CO_2}$  (MPa)', 'FontWeight', 'Bold')
```

```
ylabel('Adjusted Corrosion Rate (mm/year)', 'FontWeight', 'Bold')
```

```
legend('it T\rm=20^oC', 'it T\rm=55^oC', 'it T\rm=90^oC', 2)
```

```
title('')
```

```
subplot(1,2,2)
```

```
kappa_02=plotInteraction(mdl_04,'x_01','x_03','predictions');
```

```
set(gca,'FontName','Times New Roman')
```

```
set(kappa_02,'LineWidth',2.5)
```

```
axis([3.3 Inf 0 Inf]);
```

```
xlabel('Velocity (m/s)', 'FontWeight', 'Bold')
```

```
ylabel('Adjusted Corrosion Rate (mm/year)', 'FontWeight', 'Bold')
```

```
legend('it T\rm=20^oC', 'it T\rm=55^oC', 'it T\rm=90^oC', 2)
```

```
title('')
```

```
figure('Name','Interaction Plots of Temperature with pH and CO2 Partial Pressure  
with Velocity')
```

```
subplot(1,2,1)
kappa_03=plotInteraction mdl_04,'x_01','x_04','predictions');
set(gca,'FontName','Times New Roman')
set(kappa_03,'LineWidth',2.5)
axis([3.3 Inf 0 Inf]);
xlabel('pH','FontWeight','Bold')
ylabel('Adjusted Corrosion Rate (mm/year)','FontWeight','Bold')
legend('\it T\rm=20^oC','\it T\rm=55^oC','\it T\rm =90^oC',2)
title("")

subplot(1,2,2)
kappa_04=plotInteraction mdl_04,'x_02','x_03','predictions');
set(gca,'FontName','Times New Roman')
set(kappa_04,'LineWidth',2.5)
xlabel('Velocity (m/s)','FontWeight','Bold')
ylabel('Adjusted Corrosion Rate (mm/year)','FontWeight','Bold')
legend('\it P\rm=0.04MPa','\it P\rm=1.07MPa','\it P\rm=2.1MPa',2)
title("")

figure('Name','Interaction Plots of CO_2 Partial Pressure with pH and Velocity with pH')
subplot(1,2,1)
kappa_05=plotInteraction mdl_04,'x_02','x_04','predictions');
set(gca,'FontName','Times New Roman')
set(kappa_05,'LineWidth',2.5)
axis([3.3 Inf 0 Inf]);
xlabel('pH','FontWeight','Bold')
ylabel('Adjusted Corrosion Rate (mm/year)','FontWeight','Bold')
legend('\it P\rm=0.04MPa','\it P\rm=1.07MPa','\it P\rm=2.1MPa',3)
title("")

subplot(1,2,2)
kappa_06=plotInteraction mdl_04,'x_03','x_04','predictions');
set(gca,'FontName','Times New Roman');
set(kappa_06,'LineWidth',2.5)
axis([3.3 Inf 0 Inf]);
xlabel('pH','FontWeight','Bold')
ylabel('Adjusted Corrosion Rate (mm/year)','FontWeight','Bold')
legend('\it U\rm=0.1m/s','\it U\rm=6.55m/s','\it U\rm=13m/s',3)
title("")
```

!--Interaction Plots for all the Variables (Using 'effects')--

```
Temperature=x_01;
PCO2=x_02;
Velocity=x_03;
pH=x_04;
```

```
Corrosion_Rate=y_01;
```

```
d_02=dataset(Temperature,PCO2,Velocity,pH,Corrosion_Rate);  
mdl_05=LinearModel.fit(d_02,'Corrosion_Rate ~ Temperature + PCO2 + Velocity + pH +  
Temperature*PCO2*Velocity*pH + Temperature^2 + PCO2^2 + Velocity^2');  
mdl_06=LinearModel.fit(d_02,'Corrosion_Rate ~ Temperature + PCO2 +  
Temperature:PCO2');  
mdl_07=LinearModel.fit(d_02,'Corrosion_Rate ~ PCO2 + Temperature:PCO2 + PCO2:pH +  
Temperature^2 + PCO2^2 + Temperature:PCO2:pH - 1 - Temperature');  
mdl_08=LinearModel.fit(d_02,'Corrosion_Rate ~ Temperature + PCO2 + Velocity + pH +  
Temperature:PCO2:Velocity:pH + Temperature^2');  
%mdl_08 is the same as mdl_04 except that the x and y's in mdl_4 represent  
%temperature and co. in mdl_08
```

```
figure()  
subplot(2,3,1)  
plotInteraction(mdl_05,'Temperature','PCO2','effects')  
%xlabel('CO_2 Partial Pressure, P_C_O_2 (MPa)')  
%ylabel('Adjusted Corrosion Rate (mm/year)')  
%legend('Green=>T=20^oC','Red=>T=55^oC','Blue=>T=90^oC',2)  
title('Interaction of Temperature and CO_2 Partial Pressure')  
xlabel('Effect on Corrosion Rate')
```

```
subplot(2,3,2)  
plotInteraction(mdl_05,'Temperature','Velocity','effects')  
%axis([3.3 Inf 0 Inf]);  
%xlabel('Velocity (m/s)')  
%ylabel('Adjusted Corrosion Rate (mm/year)')  
%legend('Green=>T=20^oC','Red=>T=55^oC','Blue=>T=90^oC',2)  
title('Interaction of Temperature and Velocity')  
xlabel('Effect on Corrosion Rate')
```

```
subplot(2,3,3)  
plotInteraction(mdl_05,'Temperature','pH','effects')  
%axis([3.3 Inf 0 Inf]);  
%xlabel('pH')  
%ylabel('Adjusted Corrosion Rate (mm/year)')  
%legend('Green=>T=20^oC','Red=>T=55^oC','Blue=>T=90^oC',2)  
title('Interaction of Temperature and pH')  
xlabel('Effect on Corrosion Rate')
```

```
subplot(2,3,4)  
plotInteraction(mdl_05,'PCO2','Velocity','effects')  
%xlabel('Velocity (m/s)')  
%ylabel('Adjusted Corrosion Rate (mm/year)')  
%legend('Green=>P=0.04MPa','Red=>P=1.07MPa','Blue=>P=2.1MPa',2)  
title('Interaction of CO_2 Partial Pressure and Velocity')  
xlabel('Effect on Corrosion Rate')
```

```
subplot(2,3,5)  
plotInteraction(mdl_05,'PCO2','pH','effects')
```

```
%axis([3.3 Inf 0 Inf]);  
%xlabel('pH')  
%ylabel('Adjusted Corrosion Rate (mm/year)')  
%legend('Green=>P=0.04MPa','Red=>P=1.07MPa','Blue=>P=2.1MPa',3)  
title('Interaction of CO_2 Partial Pressure and pH')  
xlabel('Effect on Corrosion Rate')
```

```
subplot(2,3,6)  
plotInteraction mdl_05, 'Velocity', 'pH', 'effects')  
%axis([3.3 Inf 0 Inf]);  
%xlabel('pH')  
%ylabel('Adjusted Corrosion Rate (mm/year)')  
%legend('Green=>V=0.1m/s','Red=>P=6.55m/s','Blue=>P=13m/s',3)  
title('Interaction of Velocity and pH')  
xlabel('Effect on Corrosion Rate')
```

```
display mdl_04);  
display mdl_05);  
display mdl_06);  
display mdl_07);  
display mdl_08);
```

```
figure('Name','Multiple Lin Regression Case Order Plot of Leverage and Plot of Residuals  
against Fitted Values')
```

```
subplot(1,2,1)  
zeta_01=plotDiagnostics mdl_08);  
set(zeta_01,'LineWidth',2.5,'MarkerSize',8.0)  
set(gca,'FontName','Times New Roman')  
%xlim([0 80])  
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold')  
ylabel('Leverage','FontWeight','Bold')  
title('')  
%Leverage Plot -> Identifies outliers within the data
```

```
subplot(1,2,2)  
zeta_02=plotResiduals mdl_08, 'fitted');  
set(zeta_02,'LineWidth',2.5,'MarkerSize',8.0)  
set(gca,'FontName','Times New Roman')  
%Plot of Residuals -> plot should be an even random scatter  
xlim([0 80])  
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold')  
ylabel('Residuals','FontWeight','Bold')  
title('')
```

```
%figure()  
%plotEffects mdl_04);
```

```
%--ANOVA (Analysis of Variance)--%  
tbl_01=anova mdl_02, 'summary');  
display(tbl_01)
```

```
tbl_02=anova mdl_06,'summary');
display(tbl_02)
```

```
%Corr_Rate_01=11.787-(4.0357.*Temperature)-(1214.3.*PCO2)+(0.90548.*Velocity)-
(4.5541.*pH)+(75.02.*Temperature.*PCO2)+(0.18747.*Temperature.*Velocity)...
%      -(31.663.*PCO2.*Velocity)+(0.94402.*Temperature.*pH)+(357.1.*PCO2.*pH)-
(0.35984.*Velocity.*pH)+(0.0121.*Temperature.^2)-(188.67.*PCO2.^2)...
%      -(0.02112.*Velocity.^2)+(0.59099.*Temperature.*PCO2.*Velocity)-
(20.312.*Temperature.*PCO2.*pH)-(0.037319.*Temperature.*Velocity.*pH)...
%      +(9.3645.*PCO2.*Velocity.*pH)-(0.17275.*Temperature.*PCO2.*Velocity.*pH);
```

```
test_set_01=[ 20.0000  0.1300  13.0000  3.8000  6.7000
 20.0000  0.3600  3.1000  3.6000  12.5000
 20.0000  0.3600  8.5000  3.6000  17.0000
 40.0000  0.1400  3.1000  3.8000  13.0000
 40.0000  0.1400  8.5000  3.8000  16.0000
 40.0000  0.2500  3.1000  3.7500  19.0000
 40.0000  0.2500  13.0000  3.7500  24.0000
 60.0000  0.1500  8.5000  3.9000  20.0000
 60.0000  0.1500  13.0000  3.9000  27.0000
 60.0000  0.6500  3.1000  3.7000  30.0000
 60.0000  0.6500  13.0000  3.7000  60.0000
 90.0000  0.2400  8.5000  3.8500  17.0000
 90.0000  0.2400  13.0000  3.8500  26.0000
 90.0000  0.3300  8.5000  3.8000  15.0000
 90.0000  0.3300  13.0000  3.8000  37.0000];
```

```
Temperature_01=test_set_01(:,1);
PCO2_01=test_set_01(:,2);
Velocity_01=test_set_01(:,3);
pH_01=test_set_01(:,4);
Corr_Rate_01=test_set_01(:,5);
```

--Error Plots for Multiple Linear Regression -> Plots of Multiple Linear Corr Rate vs Experimental Corr Rate--

```
Corr_Rate_01_test=78.2+(0.9923.*Temperature_01)-
(2.0.*PCO2_01)+(0.9826.*Velocity_01)-(25.0.*pH_01)...
-(7.0e-3.*Temperature_01.^2)+(5.3011e-
3.*Temperature_01.*PCO2_01.*Velocity_01.*pH_01);
```

```
%display(Corr_Rate_01_test)
```

```
Error_test=Corr_Rate_01-Corr_Rate_01_test;
```

```
Error_test_omic_01=[-6.74 4.50 -1.81 -13.58 -4.72 -9.18 2.95 2.90 -4.67 -3.80 -6.58 -2.93
5.84 1.30 18.54];
```

```
%figure()
```

```
%bar(Error_test)
```

```
%xlabel('Test Data')
%ylabel('Error')

%figure()
X_001=linspace(0,60,15);
Y_001=X_001;
%plot(X_01,Y_01,'k-')

%hold on
%plot(Corr_Rate_01,Corr_Rate_01_test,'bx')
%xlabel('Experimental Corrosion Rate (mm/year)')
%ylabel('Multiple Linear Regression Model Corrosion Rate (mm/year)')
%legend('Blue Cross => Multiple Linear Regression')

figure('Name','Multiple Linear Regression Prediction with Error Bar Chart')
subplot(1,2,1)
plot(X_001,Y_001,'k-','LineWidth',1.5)
set(gca,'FontName','Times New Roman')

hold on
plot(Corr_Rate_01,Corr_Rate_01_test,'bx','MarkerSize',7,'LineWidth',2.5)

%-- 95% Confidence Intervals for Multiple Linear Regression --%
X_01_conf=[0
    4.2857
    8.5714
    12.8571
    17.1428
    21.4285
    25.7142
    29.9999
    34.2856
    38.5713
    42.857
    47.1427
    51.4284
    55.7141
    59.9998];

Y_01_multi_uci=[8.27397568
    11.27638841
    14.32847528
    17.47026027
    20.78720942
    24.45262515
    28.69280475
    33.53005299
    38.74822417
    44.15978721
    49.66952557
```

55.23300039
60.82832323
66.44383875
72.07287628];

Y_01_multi_lci=[-8.27397568
-2.704988409
2.814324717
8.243939729
13.49839058
18.40437485
22.73559525
26.46974701
29.82297583
32.98281279
36.04447443
39.05239961
42.02847677
44.98436125
47.92672372];

plot(X_01_conf,Y_01_multi_uci,'k--','LineWidth',1.5)
plot(X_01_conf,Y_01_multi_lci,'k--','LineWidth',1.5)

xlim([0 60]);
ylim([0 60]);
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','Bold')
ylabel('Multiple Lin Regression Model Corr Rate (mm/year)','FontWeight','Bold')
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)
text('Position',[33 17],'String',' R^2 \rm\bf =
0.65','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New
Roman','FontWeight','Bold');

subplot(1,2,2)
bar(Error_test_omic_01)
set(gca,'FontName','Times New Roman')
xlabel('Test Data','FontWeight','Bold')
ylabel('Corrosion Rate Error (mm/year)','FontWeight','Bold')
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1)

%--Error Plots for Surface Model Plot -> Plots of Surface Model Corr Rate vs Experimental
Corr Rate--%

Sur_mod_01=[17.1679
7.8581
16.5356
5.2500
13.9275
7.2684
23.1771

```
15.4265  
22.6577  
15.2441  
31.1528  
23.1853  
30.4165  
25.0632  
32.2945];
```

```
%figure()
```

```
Error_test_01=Corr_Rate_01-Sur_mod_01;  
Error_test_omic_02=[-10.47 4.64 7.75 -10.06 2.07 -6.19 0.46 11.73 4.57 0.82 -4.42 4.34  
14.76 4.71 28.85];
```

```
R_coef_01=corrcoef(Corr_Rate_01,Sur_mod_01);  
R_squared_coef_01=(R_coef_01(1,2))^2;
```

```
display(R_squared_coef_01)
```

```
%figure()
```

```
%bar(Error_test_01)  
%xlabel('Test Data')  
%ylabel('Error')
```

```
figure('Name','Response Surface Model Prediction and Error Bar Chart')  
subplot(1,2,1)  
plot(X_001,Y_001,'k-', 'LineWidth',1.5)  
set(gca, 'FontName', 'Times New Roman')
```

```
hold on
```

```
plot(Corr_Rate_01,Sur_mod_01,'bx', 'MarkerSize',7, 'LineWidth',2.5)
```

```
%-- 95% Confidence Intervals for Surface Model --%
```

```
Y_01_sur_mod_uci=[8.942072077  
11.58287086  
14.34067466  
17.33684729  
20.83055263  
25.15391021  
30.28646651  
35.88890152  
41.71546102  
47.65277063  
53.6501071  
59.68291495  
65.7382078  
71.80857065
```

```
77.88949395];
```

```
Y_01_sur_mod_lci=[-8.942072077  
-3.011470859  
2.802125339  
8.377352713  
13.45504737  
17.70308979  
21.14193349  
24.11089848  
26.85573898  
29.48982937  
32.0638929  
34.60248505  
37.1185922  
39.61962935  
42.11010605];
```

```
plot(X_01_conf,Y_01_sur_mod_uci,'k--','LineWidth',1.5)  
plot(X_01_conf,Y_01_sur_mod_lci,'k--','LineWidth',1.5)
```

```
xlim([0 60]);  
ylim([0 60]);  
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','Bold')  
ylabel('Response Surface Model Corrosion Rate (mm/year)','FontWeight','Bold')  
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)  
text('Position',[33 17],'String','\it R^2 \rm\bf=  
0.39','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New  
Roman','FontWeight','Bold');
```

```
subplot(1,2,2)  
bar(Error_test_omic_02)  
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1,'FontName','Times New  
Roman')  
xlabel('Test Data','FontWeight','Bold')  
ylabel('Corrosion Rate Error (mm/year)','FontWeight','Bold')
```

%--Error Plots for Mamdani FIS -> Plots of Mamdani FIS Corr Rate vs Experimental Corr Rate--%

```
Mamdani_Corr_Rate_01=[9.9800  
5.7500  
5.8300  
7.1200  
14.1000  
6.5400  
20.8000  
22.8000  
23.5000
```

```
7.6600  
28.9000  
23.6000  
23.8000  
24.4000  
24.7000];
```

```
Error_test_02=Corr_Rate_01-Mamdani_Corr_Rate_01;  
Error_test_omic_03=[-3.28 6.75 5.88 -9.40 1.90 -6.60 11.17 12.46 -2.80 3.20 2.20 3.50 22.34  
12.30 31.10];
```

```
figure('Name','Mamdani FIS Model Prediction and Error Bar Chart')  
subplot(1,2,1)  
plot(X_001,Y_001,'k-','LineWidth',1.5)  
set(gca,'FontName','Times New Roman')
```

```
hold on  
plot(Corr_Rate_01,Mamdani_Corr_Rate_01,'bx','MarkerSize',7,'LineWidth',2.5)
```

%-- 95% Confidence Intervals for Mamdani FIS Model--%

```
Y_01_mamdani_uci=[8.454310412  
11.05344341  
13.83667082  
17.00891198  
20.93037816  
25.79286637  
31.31271579  
37.15760942  
43.15780204  
49.23867426  
55.36553781  
61.52073541  
67.69450053  
73.88104499  
80.07673936];
```

```
Y_01_mamdani_lci=[-8.454310412  
-2.482043405  
3.306129176  
8.705288023  
13.35522184  
17.06413363  
20.11568421  
22.84219058  
25.41339796  
27.90392574  
30.34846219  
32.76466459  
35.16229947  
37.54715501
```

39.92286064];

```

plot(X_01_conf,Y_01_mamdani_uci,'k--','LineWidth',1.5)
plot(X_01_conf,Y_01_mamdani_lci,'k--','LineWidth',1.5)

xlim([0 60]);
ylim([0 60]);
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','Bold')
ylabel('Mamdani FIS Model Corrosion Rate (mm/year)','FontWeight','Bold')
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)
text('Position',[33 17],'String','\it R^2 \rm\bf=
0.32','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times
Roman','FontWeight','Bold');

subplot(1,2,2)
bar(Error_test_omic_03)
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1,'FontName','Times New
Roman')
xlabel('Test Data','FontWeight','Bold')
ylabel('Corrosion Rate Error (mm/year)','FontWeight','Bold')

%--Error Plots for Sugeno FIS -> Plots of Sugeno FIS Corr Rate vs Experimental Corr Rate--
%
Sugeno_Corr_Rate_01=[5.3700
9.7200
26.5000
4.2300
6.0200
4.6800
8.0100
11.8000
12.0000
5.0200
29.4000
29.4000
30.0000
29.8000
30.0000];

Error_test_03=Corr_Rate_01-Sugeno_Corr_Rate_01;

figure()
subplot(1,2,1)
plot(X_001,Y_001,'k-')
hold on
plot(Corr_Rate_01,Sugeno_Corr_Rate_01,'bx')
xlim([0 60]);

```

```
ylim([0 60]);
xlabel('Experimental Corrosion Rate (mm/year)')
ylabel('Sugeno FIS Model Corrosion Rate (mm/year)')
%legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)

subplot(1,2,2)
bar(Error_test_03)
xlabel('Test Data')
ylabel('Corrosion Rate Error (mm/year)')

%--Hougen-Watson Equation Results--%
Corr_Rate_02_test=((20.675.*PCO2_01)-(Velocity_01./-1.2831e7))./(1+(-
0.0012539.*Temperature_01)+(0.31458.*PCO2_01)+(-0.062364.*Velocity_01));

Error_test_Hougen=Corr_Rate_01-Corr_Rate_02_test;
Error_test_omic_04=[-6.41 4.18 9.38 0.20 9.76 3.66 5.53 12.81 12.98 0.26 -6.66 7.76 15.64 -
0.86 17.81];

figure('Name','Nonlinear Model Prediction and Error Bar Chart')

subplot(1,2,1)
plot(X_001,Y_001,'k-','LineWidth',1.5)
set(gca,'FontName','Times New Roman')
hold on
plot(Corr_Rate_01,Corr_Rate_02_test,'bx','MarkerSize',7,'LineWidth',2.5)

%-- 95% Confidence Intervals for Hougen-Watson Nonlinear Model--%

Y_01_hougen_uci=[6.504021634
9.729126352
13.11129082
16.76113024
20.8202204
25.35997741
30.30121796
35.50227022
40.85542096
46.29788671
51.79505695
57.32742911
62.88351577
68.45622592
74.04098918];

Y_01_hougen_lci=[-6.504021634
-1.157726352
4.031509175
8.953069755
13.4653796
```

```
17.49702259  
21.12718204  
24.49752978  
27.71577904  
30.84471329  
33.91894305  
36.95797089  
39.97328423  
42.97197408  
45.95861082];
```

```
plot(X_01_conf,Y_01_hougen_uci,'k--','LineWidth',1.5)  
plot(X_01_conf,Y_01_hougen_lci,'k--','LineWidth',1.5)
```

```
xlim([0 60]);  
ylim([0 60]);  
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','Bold')  
ylabel('Nonlinear Model Corrosion Rate (mm/year)','FontWeight','Bold')  
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)  
text('Position',[33 17],'String','\it R^2 \rm\bf=  
0.67','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New  
Roman','FontWeight','Bold');
```

```
subplot(1,2,2)  
bar(Error_test_omic_04)  
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1,'FontName','Times New  
Roman')  
xlabel('Test Data','FontWeight','Bold')  
ylabel('Corrosion Rate Error (mm/year)','FontWeight','Bold')
```

!--Error Plots for Neural Network Corrosion Model -> Plots of Neural Network Model Corr
Rate vs Experimental Corr Rate--%

```
Neural_Net_Corr_Rate_01=[13.4935  
16.1991  
13.8781  
18.9542  
14.7386  
19.2873  
25.0607  
20.3173  
30.9718  
25.8677  
36.0933  
11.2166  
17.6583  
14.9526  
19.9719];
```

```
Error_test_neu_net_01=Corr_Rate_01-Neural_Net_Corr_Rate_01;  
Error_test_omic_05=[-6.80 -3.70 -5.95 0.05 1.26 3.12 5.78 -0.29 -0.32 -1.06 8.34 -3.97 4.13  
17.03 23.91];
```

```
R_coef_nn_01=corrcoef(Corr_Rate_01,Neural_Net_Corr_Rate_01);  
R_sq_coef_nn_01=(R_coef_nn_01(1,2))^2;
```

```
display(R_sq_coef_nn_01)
```

```
%figure()  
%bar(Error_test_01)  
%xlabel('Test Data')  
%ylabel('Error')
```

```
figure('Name','Neural Network Model Prediction and Error Bar Chart')  
subplot(1,2,1)  
plot(X_001,Y_001,'k-','LineWidth',1.5)  
set(gca,'FontName','Times New Roman')  
hold on  
plot(Corr_Rate_01,Neural_Net_Corr_Rate_01,'bx','MarkerSize',7,'LineWidth',2.5)
```

```
%-- 95% Confidence Intervals for Neural Network Model--%
```

```
Y_01_neunet_uci=[6.895212776  
9.865263005  
12.89318747  
16.04817034  
19.51909385  
23.68002714  
28.62121496  
33.97367112  
39.49038024  
45.07878625  
50.70325877  
56.34807536  
62.00539645  
67.67091947  
73.34209999];
```

```
Y_01_neunet_lci=[-6.895212776  
-1.293863005  
4.249612535  
9.666029655  
14.76650615  
19.17697286  
22.80718504  
26.02612888  
29.08081976  
32.06381375  
35.01074123
```

37.93732464
40.85140355
43.75728053
46.65750001];

```
plot(X_01_conf,Y_01_neunet_uci,'k--','LineWidth',1.5)  
plot(X_01_conf,Y_01_neunet_lci,'k--','LineWidth',1.5)
```

```
xlim([0 60]);  
ylim([0 60]);  
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','Bold')  
ylabel('Neural Network Corrosion Rate (mm/year)','FontWeight','Bold')  
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)  
text('Position',[33 17],'String','R^2 = 0.6444','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New Roman','FontWeight','Bold');
```

```
subplot(1,2,2)  
bar(Error_test_omic_05)  
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1,'FontName','Times New Roman')  
xlabel('Test Data','FontWeight','Bold')  
ylabel('Corrosion Rate Error (mm/year)','FontWeight','Bold')
```

%---Empirical models and other models---%

%--Freecorp Model--%

```
Freecorp_Corr_Rate_01=[4.2  
5.6  
6.4  
6.2  
7.1  
7.9  
9.1  
10.8  
11.1  
23.9  
25.3  
28.2  
28.6  
34.6  
35];
```

```
Error_test_freecorp_01=Corr_Rate_01-Freecorp_Corr_Rate_01;  
Error_test_omic_06=[2.50 6.90 6.80 -19.60 8.90 -11.20 10.60 11.10 9.20 14.90 -2.60 15.90  
6.10 2.00 34.70];
```

```
R_coef_freecorp_01=corrcoef(Corr_Rate_01,Freecorp_Corr_Rate_01);  
R_sq_coef_freecorp_01=(R_coef_freecorp_01(1,2))^2;
```

```
display(R_sq_coef_freecorp_01)
```

-- 95% Confidence Intervals for Freecorp Model--

```
Y_01_freecorp_uci=[9.41690706  
12.09367008  
15.02585986  
18.40373002  
22.4606524  
27.27686736  
32.68102457  
38.44260606  
44.40643611  
50.48737145  
56.639728  
62.83804391  
69.06737523  
75.31849927  
81.58546403];
```

```
Y_01_freecorp_lci=[-9.41690706  
-3.52227008  
2.116940139  
7.310469983  
11.8249476  
15.58013264  
18.74737543  
21.55719394  
24.16476389  
26.65522855  
29.074272  
31.44735609  
33.78942477  
36.10970073  
38.41413597];
```

```
figure('Name','Freecorp Model Prediction and Error Bar Chart')
```

```
subplot(1,2,1)  
plot(Corr_Rate_01,Freecorp_Corr_Rate_01,'bx','MarkerSize',7,'LineWidth',2.5)  
set(gca,'FontName','Times New Roman')
```

```
hold on
```

```
plot(X_001,Y_001,'k-','LineWidth',1.5)  
plot(X_01_conf,Y_01_freecorp_uci,'k--','LineWidth',1.5)  
plot(X_01_conf,Y_01_freecorp_lci,'k--','LineWidth',1.5)
```

```
xlim([0 60]);  
ylim([0 60]);  
xlabel('Experimental Corrosion Rate (mm/year)', 'FontWeight', 'Bold')  
ylabel('Freecorp Model Corrosion Rate (mm/year)', 'FontWeight', 'Bold')  
legend('Datapoints', 'Experiment=Model Line', '95% Confidence Interval', 2)  
text('Position', [33 17], 'String', '\it R^2 \rm\bf=0.24', 'Background', 'w', 'EdgeColor', 'black', 'LineWidth', 1.0, 'FontName', 'Times New Roman', 'FontWeight', 'Bold');
```

```
subplot(1,2,2)  
bar(Error_test_omic_06)  
set(gca, 'XMinorTick', 'on', 'XTick', [0 5 10 15 20], 'LineWidth', 1, 'FontName', 'Times New Roman')  
xlabel('Test Data', 'FontWeight', 'Bold')  
ylabel('Corrosion Rate Error (mm/year)', 'FontWeight', 'Bold')
```

%--Norsok Model--%

```
Norsok_Corr_Rate_01=[11.921  
11.865  
16.8267  
15.6008  
21.7844  
21.7068  
35.7804  
26.9424  
31.1412  
44.4108  
76.0114  
24.979  
29.0564  
30.7831  
35.9498];
```

```
Error_test_norsok_01=Corr_Rate_01-Norsok_Corr_Rate_01;  
Error_test_omic_07=[-5.22 0.64 -2.60 -15.78 -5.78 -7.98 0.17 -2.71 -6.94 -11.78 -3.06 -4.14 -14.41 1.05 -16.01];
```

```
R_coef_norsok_01=corrcoef(Corr_Rate_01,Norsok_Corr_Rate_01);  
R_sq_coef_norsok_01=(R_coef_norsok_01(1,2))^2;
```

```
display(R_sq_coef_norsok_01)
```

%-- 95% Confidence Intervals for Freecorp Model--%

```
Y_01_norsok_uci=[ 6.059941559  
10.86825435  
15.74434531
```

```
20.73596666  
25.92585717  
31.42548738  
37.30173923  
43.50275517  
49.91717948  
56.45663953  
63.0691332  
69.72606932  
76.41143338  
83.11585138  
89.83356939];
```

```
Y_01_norsok_lci=[-6.059941559  
0.560345647  
7.11285469  
13.54983334  
19.78854283  
25.71751262  
31.26986077  
36.49744483  
41.51162052  
46.40076047  
51.2168668  
55.98853068  
60.73176662  
65.45594862  
70.16683061];
```

```
X_01_norsok_conf=linspace(0,80,15);
```

```
X_002=linspace(0,80,15);  
Y_002=X_002;
```

```
figure('Name','Norsok Model Prediction and Error Bar Chart')
```

```
subplot(1,2,1)
```

```
plot(Corr_Rate_01,Norsok_Corr_Rate_01,'bx','MarkerSize',7,'LineWidth',2.5)  
set(gca,'FontName','Times New Roman')
```

```
hold on
```

```
plot(X_002,Y_002,'k-','LineWidth',1.5)  
plot(X_01_norsok_conf,Y_01_norsok_uci,'k--','LineWidth',1.5)  
plot(X_01_norsok_conf,Y_01_norsok_lci,'k--','LineWidth',1.5)
```

```
xlim([0 80]);  
ylim([0 80]);  
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','Bold')  
ylabel('Norsok Model Corrosion Rate (mm/year)','FontWeight','Bold')
```

```
legend('Datapoints','Experiment=Model Line','95% Confidence Interval',2)
text('Position',[33 17],'String','it R^2 \rm\bf=
0.89','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New
Roman','FontWeight','Bold');
```

```
subplot(1,2,2)
bar(Error_test_omic_07)
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1,'FontName','Times New
Roman')
xlabel('Test Data','FontWeight','Bold')
ylabel('Corrosion Rate Error (mm/year)','FontWeight','Bold')
```

%--De Waard 1991 Model--%

```
DeWaard_1991_Corr_Rate_01=[1.0969
2.1704
2.1704
2.7205
2.7205
4.012
4.012
6.0652
6.0652
16.2001
16.2001
10.6522
10.6522
10.8924
10.8924];
```

```
Error_test_dewaard_1991_01=Corr_Rate_01-DeWaard_1991_Corr_Rate_01;
Error_test_omic_08=[5.60 10.33 10.28 4.11 13.28 6.35 14.83 14.99 13.93 19.99 15.35 20.93
13.80 26.11 43.80];
```

```
R_coef_dewaard_1991_01=corrcoef(Corr_Rate_01,DeWaard_1991_Corr_Rate_01);
R_sq_coef_dewaard_1991_01=(R_coef_dewaard_1991_01(1,2))^2;
```

```
display(R_sq_coef_dewaard_1991_01)
```

%-- 95% Confidence Intervals for the De Waard 1991 Model--%

```
Y_01_dewaard_1991_uci=[ 3.234752517
6.409148669
10.4951783
15.69246417
21.32898385
27.09907835
32.92067353
38.76674327
44.62621408
```

```
50.49378029
56.36659849
62.24301274
68.12199507
74.00287433
79.88519401];

Y_01_dewaard_1991_lci=[-3.234752517
2.162251331
6.647621697
10.02173583
12.95661615
15.75792165
18.50772647
21.23305673
23.94498592
26.64881971
29.34740151
32.04238726
34.73480493
37.42532567
40.11440599];

%X_01_norsok_conf=linspace(0,80,15);

%X_002=linspace(0,80,15);
%Y_002=X_002;

figure('Name','The 1991 De Waard Model Prediction and Error Bar Chart')

subplot(1,2,1)
plot(Corr_Rate_01,DeWaard_1991_Corr_Rate_01,'bx','MarkerSize',7,'LineWidth',2.5)
set(gca,'FontName','Times New Roman')

hold on

plot(X_001,Y_001,'k-', 'LineWidth',1.5)
plot(X_01_conf,Y_01_dewaard_1991_uci,'k--','LineWidth',1.5)
plot(X_01_conf,Y_01_dewaard_1991_lci,'k--','LineWidth',1.5)

xlim([0 60]);
ylim([0 60]);
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','Bold')
ylabel('The 1991 De Waard Model Corrosion Rate (mm/year)','FontWeight','Bold')
legend('Datapoints','Experiment=Model Line','95% Confidence Interval',2)
text('Position',[33 17],'String','\it R^2 \rm\bf=
0.54','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times
New Roman','FontWeight','Bold');
```

```
bar(Error_test_omic_08)
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1,'FontName','Times New Roman')
xlabel('Test Data','FontWeight','Bold')
ylabel('Corrosion Rate Error (mm/year)','FontWeight','Bold')
```

%--De Waard 1995 Model--%

```
DeWaard_1995_Corr_Rate_01=[10.4845
14.6379
19.1235
9.0386
14.4027
14.4644
24.5685
18.9571
22.8202
38.1299
65.4
20.0906
25.1775
22.9145
28.3876];
```

```
Error_test_dewaard_1995_01=Corr_Rate_01-DeWaard_1995_Corr_Rate_01;
Error_test_omic_09=[-3.78 -2.14 3.96 -7.91 1.60 -2.12 -3.09 4.54 1.04 -0.57 0.82 4.18 -8.13
8.61 -5.40];
```

```
R_coef_dewaard_1995_01=corrcoef(Corr_Rate_01,DeWaard_1995_Corr_Rate_01);
R_sq_coef_dewaard_1995_01=(R_coef_dewaard_1995_01(1,2))^2;
```

```
display(R_sq_coef_dewaard_1995_01)
```

%-- 95% Confidence Intervals for the De Waard 1995 Model--%

```
Y_01_dewaard_1995_uci=[5.096154396
9.297678133
13.57457427
17.98218398
22.6110453
27.55929483
32.84441886
38.38226898
44.07394949
49.85414067
55.6865275
61.5512445
67.43702294
73.33715252
79.24744344];
```

```
Y_01_dewaard_1995_lci=[-5.096154396
    0.702321867
    6.425425728
    12.01781602
    17.3889547
    22.44070517
    27.15558114
    31.61773102
    35.92605051
    40.14585933
    44.3134725
    48.4487555
    52.56297706
    56.66284748
    60.75255656];

X_01_dewaard_1995_conf=linspace(0,70,15);

X_003=linspace(0,70,15);
Y_003=X_003;

figure('Name','The 1995 De Waard Model Prediction and Error Bar Chart')

subplot(1,2,1)
plot(Corr_Rate_01,DeWaard_1995_Corr_Rate_01,'bx','MarkerSize',7,'LineWidth',2.5)
set(gca,'FontName','Times New Roman')
%set(gca,'YGrid','on')

hold on

plot(X_003,Y_003,'k-','LineWidth',1.5)
plot(X_01_dewaard_1995_conf,Y_01_dewaard_1995_uci,'k--','LineWidth',1.5)
plot(X_01_dewaard_1995_conf,Y_01_dewaard_1995_lci,'k--','LineWidth',1.5)

xlim([0 70]);
ylim([0 70]);
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','Bold')
ylabel('The 1995 De Waard Model Corrosion Rate (mm/year)','FontWeight','Bold')
legend('Datapoints', 'Experiment=Model Line', '95% Confidence Interval',2)
text('Position',[33 17],'String','it R^2 \rm\bf=
0.88','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New
Roman','FontWeight','Bold');
```

Neural Network Model Code

```
% Same neural network as the first except with a tansig transfer function in place
% of the logsig transfer function in the first hidden layer

clc, clear, close all

%tic

%R_coef_nn_01=0;

%while R_coef_nn_01<0.85

%repeat

%net_03=network; % Create network

%net.numInputs=1; % Set number of inputs
%net.inputs{1}.size

Raw_01=xlsread('G:\University Files and More\University of Newcastle Upon Tyne
Files\PhD\Dugstad_Data_From_Original_Journal_Article_01.xlsx','Training_01');

%[x_1,t_1]=size(Raw_01);

x_1=Raw_01;

%x_1=[40,50,60,80,40;          5.98,10.93,14.13,17.35,13.64];          %2.7,2.7,2.7,2.7,2.7;
4.5,4.5,4.5,4.5,4.4; 5.98,10.93,14.13,17.35,13.64];
%t_1=[20,40,60,80,100]; %;5.5,7.5,14,11.5,10.0];

net_01=feedforwardnet([10 5]);%Custom neural network with 10 hidden layers

Raw_02=xlsread('G:\University Files and More\University of Newcastle Upon Tyne
Files\PhD\Dugstad_Data_From_Original_Journal_Article_01.xlsx','Corr_Rates_01');

CR_01_target=Raw_02;

%[trainInd,valInd,testInd]=dividerand(98,0.6,0.2,0.2);

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
% The following syntax makes sure that the input is divided up exactly
% in the stated percentages at all times
%net_01.divideFcn = 'dividerand'; % Divide data randomly
%[trainInd,valInd,testInd] = divideint(60,0.6,0.2,0.2);

%net_01.divideMode = 'sample'; % Divide up every sample
net_01.divideParam.trainRatio = 60/100;
net_01.divideParam.valRatio = 20/100;
```

```
net_01.divideParam.testRatio = 20/100;

% For help on training function 'trainlm' type: help trainlm
% For a list of all training functions type: help nntrain
net_01.trainFcn = 'trainlm'; % Levenberg-Marquardt

net_01.trainParam.epochs=100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net_01.performFcn = 'mse'; % Mean squared error

%net_01=train(net_01,x_1,CR_01_target);
[net_01,tr]=train(net_01,x_1,CR_01_target);

net_01.layers{1}.transferFcn = 'tansig';% 'tansig';'tansig';
net_01.layers{2}.transferFcn = 'tansig';
net_01.layers{3}.transferFcn = 'tansig';

net_01.inputWeights{1}.initFcn = 'initzero'; % set input weight init function
net_01.inputWeights{1}.learnFcn = 'learnp'; % set input weight learning function
net_01.inputWeights{2}.learnFcn = 'learnp';
net_01.inputWeights{2}.initFcn = 'initzero';

%view(net_01);
y_1=net_01(x_1);
%size(y_1)=196;
perf_01=perform(net_01,y_1,CR_01_target);

x_2=xlsread('G:\University Files and More\University of Newcastle Upon Tyne
Files\PhD\Dugstad_Data_From_Original_Journal_Article_01.xlsx','Test_Dataset_01');

%x_2=[16 17 3 10 20 25 32 11 5 19];
y_2=sim(net_01,x_2); % Simulating the neural network, in terms of net_01 and x_1
%figure()
%plot(t_1,y_1,'k-x')
display (y_2)

X_01=linspace(0,60,15);
Y_01=X_01;
x_test_01=xlsread('G:\University Files and More\University of Newcastle Upon Tyne
Files\PhD\Dugstad_Data_From_Original_Journal_Article_01.xlsx','Test_Corr_Rates_01');
error_01=x_test_01-y_2;
E_01=sum(abs(error_01));
display(E_01)

R_coef_nn_01=corrcoef(x_test_01,y_2);
```

```
R_sq_coef_nn_01=(R_coef_nn_01(1,2))^2;

%until R_coef_nn_01==0.85 end_repeat

%else

    %display(R_sq_coef_nn_01)

%end

%if R_sq_coef_nn_01<0.65

    while R_sq_coef_nn_01<0.85

        Raw_01=xlsread('G:\University Files and More\University of Newcastle Upon Tyne
Files\PhD\Dugstad_Data_From_Original_Journal_Article_01.xlsx','Training_01');

        %[x_1,t_1]=size(Raw_01);

        x_1=Raw_01;

        %x_1=[40,50,60,80,40;          5.98,10.93,14.13,17.35,13.64];          %2.7,2.7,2.7,2.7,2.7;
4.5,4.5,4.5,4.5,4.4; 5.98,10.93,14.13,17.35,13.64];
        %t_1=[20,40,60,80,100]; %;5.5,7.5,14,11.5,10.0];

        net_01=feedforwardnet([10 5]);%Custom neural network with 10 hidden layers

        Raw_02=xlsread('G:\University Files and More\University of Newcastle Upon Tyne
Files\PhD\Dugstad_Data_From_Original_Journal_Article_01.xlsx','Corr_Rates_01');

        CR_01_target=Raw_02;

        %[trainInd,valInd,testInd]=dividerand(98,0.6,0.2,0.2);

        % Setup Division of Data for Training, Validation, Testing
        % For a list of all data division functions type: help nndivide
        % The following syntax makes sure that the input is divided up exactly
        % in the stated percentages at all times
        %net_01.divideFcn = 'dividerand'; % Divide data randomly
        %[trainInd,valInd,testInd] = divideint(60,0.6,0.2,0.2);

        %net_01.divideMode = 'sample'; % Divide up every sample
        net_01.divideParam.trainRatio = 60/100;
        net_01.divideParam.valRatio = 20/100;
        net_01.divideParam.testRatio = 20/100;

        % For help on training function 'trainlm' type: help trainlm
        % For a list of all training functions type: help nntrain
```

```
net_01.trainFcn = 'trainlm'; % Levenberg-Marquardt

net_01.trainParam.epochs=100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net_01.performFcn = 'mse'; % Mean squared error

%net_01=train(net_01,x_1,CR_01_target);
[net_01,tr]=train(net_01,x_1,CR_01_target);

net_01.layers{1}.transferFcn = 'tansig';% 'tansig';'tansig';
net_01.layers{2}.transferFcn = 'tansig';
net_01.layers{3}.transferFcn = 'tansig';

net_01.inputWeights{1}.initFcn = 'initzero'; % set input weight init function
net_01.inputWeights{1}.learnFcn = 'learnp'; % set input weight learning function
net_01.inputWeights{2}.learnFcn = 'learnp';
net_01.inputWeights{2}.initFcn = 'initzero';

%view(net_01);
y_1=net_01(x_1);
%size(y_1)=196;
perf_01=perform(net_01,y_1,CR_01_target);

x_2=xlsread('G:\University Files and More\University of Newcastle Upon Tyne
Files\PhD\Dugstad_Data_From_Original_Journal_Article_01.xlsx','Test_Dataset_01');

%x_2=[16 17 3 10 20 25 32 11 5 19];
y_2=sim(net_01,x_2); % Simulating the neural network, in terms of net_01 and x_1
%figure()
%plot(t_1,y_1,'k-x')
display (y_2)

X_01=linspace(0,60,15);
Y_01=X_01;
x_test_01=xlsread('G:\University Files and More\University of Newcastle Upon Tyne
Files\PhD\Dugstad_Data_From_Original_Journal_Article_01.xlsx','Test_Corr_Rates_01');
error_01=x_test_01-y_2;
E_01=sum(abs(error_01));
display(E_01)

R_coef_nn_01=corrcoef(x_test_01,y_2);
R_sq_coef_nn_01=(R_coef_nn_01(1,2))^2;

display(R_sq_coef_nn_01)
end
```

tic

```
figure()  
subplot(1,2,1)  
plot(X_01,Y_01,'k-')
```

hold on

```
plot(x_test_01,y_2,'bx')  
xlabel('Experimental Corrosion Rate (mm/year)','FontName','Times New Roman')  
ylabel('Neural Network Corrosion Rate (mm/year)','FontName','Times New Roman')
```

```
subplot(1,2,2)  
bar(error_01)  
xlabel('Test Data','FontName','Times New Roman')  
ylabel('Error','FontName','Times New Roman')
```

```
figure()  
plot(CR_01_target,y_1,'x')  
%title('Neural Network Corrosion Rate Against Experimental Corrosion Rate for  
Supercritical CO2 (ScCO2) Pressures')  
xlabel('Experimental Corrosion Rate (mm/year)','FontName','Times New Roman')  
ylabel('Neural Network Corrosion Rate (mm/year)','FontName','Times New Roman')
```

```
figure()  
plotregression(CR_01_target,y_1)  
%title('Neural Network Corrosion Rate Against Experimental Corrosion Rate for  
Supercritical CO2 (ScCO2) Pressures')  
xlabel('Experimental Corrosion Rate (mm/year)','FontName','Times New Roman')  
ylabel('Neural Network Corrosion Rate (mm/year)','FontName','Times New Roman')
```

```
figure()  
plotperform(tr)
```

%--95% Confidence Interval Data--%

```
X_01_conf=[0  
4.2857  
8.5714  
12.8571  
17.1428  
21.4285  
25.7142  
29.9999  
34.2856  
38.5713  
42.857  
47.1427
```

51.4284
55.7141
59.9998];

```
Neural_Net_Corr_Rate_01=[12.3923  
7.0497  
11.1169  
12.1380  
15.7352  
13.6138  
17.2251  
23.5501  
37.0132  
28.0422  
44.6256  
20.5530  
27.3942  
20.8959  
29.3329];
```

```
Error_test_neu_net_01=x_test_01.'-Neural_Net_Corr_Rate_01;  
R_coef_nn_01=corrcoef(x_test_01.',Neural_Net_Corr_Rate_01);  
R_sq_coef_nn_01=(R_coef_nn_01(1,2))^2;
```

```
display(R_sq_coef_nn_01)
```

```
%figure()  
%bar(Error_test_01)  
%xlabel('Test Data')  
%ylabel('Error')
```

```
figure()  
subplot(1,2,1)  
plot(X_01,Y_01,'k-', 'LineWidth',1.5)  
hold on  
plot(x_test_01.',Neural_Net_Corr_Rate_01,'bx', 'MarkerSize',7, 'LineWidth',2.5)
```

```
%-- 95% Confidence Intervals for Neural Network Model--%
```

```
Y_01_neunet_uci=[6.895212776  
9.865263005  
12.89318747  
16.04817034  
19.51909385  
23.68002714  
28.62121496  
33.97367112  
39.49038024  
45.07878625  
50.70325877  
56.34807536
```

```
62.00539645
67.67091947
73.34209999];
```

```
Y_01_neunet_lci=[-6.895212776
-1.293863005
4.249612535
9.666029655
14.76650615
19.17697286
22.80718504
26.02612888
29.08081976
32.06381375
35.01074123
37.93732464
40.85140355
43.75728053
46.65750001];
```

```
plot(X_01_conf,Y_01_neunet_uci,'k--','LineWidth',1.5)
plot(X_01_conf,Y_01_neunet_lci,'k--','LineWidth',1.5)
```

```
xlim([0 60]);
ylim([0 60]);
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','bold','FontName','Times New Roman')
ylabel('Neural Network Corrosion Rate (mm/year)','FontWeight','bold','FontName','Times New Roman')
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)
text('Position',[33 17],'String','R^2 = 0.6444','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New Roman');
```

```
subplot(1,2,2)
bar(Error_test_neu_net_01)
xlabel('Test Data','FontWeight','bold','FontName','Times New Roman')
ylabel('Corrosion Rate Error (mm/year)','FontWeight','bold','FontName','Times New Roman')
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1)
```

```
%%%---For Updated Dugstad 1994 NN Model Report-%%%
```

```
%--95% Confidence Interval Data--%
```

```
X_02_conf=[0
4.2857
8.5714
12.8571
17.1428
21.4285
```

```
25.7142  
29.9999  
34.2856  
38.5713  
42.857  
47.1427  
51.4284  
55.7141  
59.9998];
```

```
Neural_Net_Corr_Rate_02=[14.4904  
20.3834  
14.7004  
16.2215  
22.1838  
20.5459  
20.8682  
18.7632  
39.8937  
29.00017  
48.1738  
17.4176  
27.727  
19.1816  
34.316];
```

```
Error_test_neu_net_02=x_test_01.'-Neural_Net_Corr_Rate_02;  
R_coef_nn_02=corrcoef(x_test_01.',Neural_Net_Corr_Rate_02);  
R_sq_coef_nn_02=(R_coef_nn_02(1,2))^2;
```

```
%display(R_sq_coef_nn_02)
```

```
figure()
```

```
subplot(1,2,1)
```

```
plot(X_01,Y_01,'k-', 'LineWidth',1.5)
```

```
hold on
```

```
plot(x_test_01.',Neural_Net_Corr_Rate_02,'bx','MarkerSize',7,'LineWidth',2.5)
```

```
%-- 95% Confidence Intervals for Neural Network Model--%
```

```
Y_02_neunet_uci=[6.270715157  
9.608877368  
12.98495857  
16.42805428  
19.99812187  
23.81239276  
28.02546799  
32.67008875  
37.60927109  
42.70562029
```

```
47.88366413  
53.10684567  
58.35686651  
63.62391643  
68.90236576];
```

```
Y_02_neunet_lci=[-6.270715157  
-1.037477368  
4.15784143  
9.28614572  
14.28747813  
19.04460724  
23.40293201  
27.32971125  
30.96192891  
34.43697971  
37.83033587  
41.17855433  
44.49993349  
47.80428357  
51.09723424];
```

```
plot(X_02_conf,Y_02_neunet_uci,'k--','LineWidth',1.5)  
plot(X_02_conf,Y_02_neunet_lci,'k--','LineWidth',1.5)
```

```
xlim([0 60]);  
ylim([0 60]);  
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','bold','FontName','Times New Roman')  
ylabel('Neural Network Corrosion Rate (mm/year)','FontWeight','bold','FontName','Times New Roman')  
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)  
text('Position',[33 17],'String','R^2 = 0.81','Background','w','EdgeColor','black','LineWidth',1.0);
```

```
subplot(1,2,2)  
bar(Error_test_neu_net_02)  
xlabel('Test Data','FontWeight','bold')  
ylabel('Corrosion Rate Error (mm/year)','FontWeight','bold')  
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1)
```

```
figure()
```

```
plot(X_01,Y_01,'k-','LineWidth',1.5)  
hold on  
plot(x_test_01.',Neural_Net_Corr_Rate_02,'bx','MarkerSize',7,'LineWidth',2.5)  
plot(X_02_conf,Y_02_neunet_uci,'k--','LineWidth',1.5)  
plot(X_02_conf,Y_02_neunet_lci,'k--','LineWidth',1.5)
```

```
xlim([0 60]);
```

```
ylim([0 60]);
xlabel('Experimental Corrosion Rate (mm/year)',FontWeight,'bold',FontName,'Times New Roman')
ylabel('Neural Network Corrosion Rate (mm/year)',FontWeight,'bold',FontName,'Times New Roman')
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)
text('Position',[33 17],String,'R^2 = 0.81','Background','w','EdgeColor','black','LineWidth',1.0,FontName,'Times New Roman');

figure()

bar(Error_test_neu_net_02)
xlabel('Test Data',FontWeight,'bold',FontName,'Times New Roman')
ylabel('Corrosion Rate Error (mm/year)',FontWeight,'bold',FontName,'Times New Roman')
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],LineWidth,1)

%t_1=linspace(20,90,15);

%figure()
%plot(t_1,y_1,'x')
%xlabel('Temperature (^oC)')
%ylabel('Corrosion Rate (mm/year)')

%[P_1,S_1]=polyfit(t_1,y_1,3);
%Y_1=polyval(P_1,t_1);
%hold on
%plot(t_1,Y_1,'k');

y_2_pub=[17.1358 15.9235 16.9200 15.6105 20.4090 16.2710 28.5162 16.5049
23.3663 29.6982 54.1438 15.6518 ...
26.1009 21.1353 35.7521];
%y_2_pub are the NN predicted results for publication

%y_2_pub_bar=mean(y_2_pub);
%y_2_pub_bar is the mean of y_2_pub

%y_2_pub_std=std(y_2_pub);
%y_2_pub_std is the standard deviation of y_2_pub

%n_pub=length(y_2_pub);
%n_pub is the number of NN predicted data-points in y_2_pub

%ci=0.95;
%ci is the confidence interval. In this case it's set at 95%

%alpha=1-ci;

%T_multiplier=tinv(1-alpha/2,n_pub-1);
```

%T_multiplier is part of the formulae for computing the confidence interval
%tinv is the matlab function which enables the computation of T_multiplier
%n_pub-1 is so because one degree of freedom is used in calculating the
%mean or standard deviation

```
%ci95=(T_multiplier.*y_2_pub_std)./sqrt(n_pub);
```

```
%conf_inf_95_01=[y_2_pub-ci95, y_2_pub+ci95];
```

```
%for n=1:length(y_2_pub)  
%conf_int_95_01=y_2_pub-ci95;  
%conf_int_95_02=y_2_pub+ci95;
```

```
%display(conf_int_95_01)  
%display(conf_int_95_02)  
%end
```

```
%display(conf_inf_95_01)
```

```
Y_02_net_pub_01=[4.347933016  
7.977103099  
11.6382261  
15.35538299  
19.17613313  
23.18477432  
27.47249051  
32.03845381  
36.79096383  
41.64620233  
46.5570312  
51.49958737  
56.46146411  
61.43581576  
66.41862842];
```

```
Y_02_net_pub_02=[-4.347933016  
0.594296901  
5.504573904  
10.35881701  
15.10946687  
19.67222568  
23.95590949  
27.96134619  
31.78023617  
35.49639767  
39.1569688  
42.78581263  
46.39533589  
49.99238424
```

53.58097158];

figure()

```
plot(X_01,Y_01,'k-', 'LineWidth',1.5)
set(gca,'FontName','Times New Roman')
```

```
hold on
```

```
plot(x_test_01,y_2_pub,'bx','MarkerSize',7,'LineWidth',2.5)
plot(X_02_conf,Y_02_net_pub_01,'k--','LineWidth',1.5)
plot(X_02_conf,Y_02_net_pub_02,'k--','LineWidth',1.5)
%plot(conf_int_95_01,y_2_pub,'k--','LineWidth',1.5)
%plot(conf_int_95_02,y_2_pub,'k--','LineWidth',1.5)
```

```
xlim([0 60]);
```

```
ylim([0 60]);
```

```
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','bold','FontName','Times New Roman')
```

```
ylabel('Neural Network Corrosion Rate (mm/year)','FontWeight','bold','FontName','Times New Roman')
```

```
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)
```

```
text('Position',[33 17],'String',' $\hat{R}^2$  = 0.91','FontWeight','bold','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New Roman');
```

```
%Error_pub_01=x_test_01-y_2_pub;
```

```
Error_pub_02=[-10.44 -3.42 -2.61 -6.14 -4.41 0.08 1.35 2.73 3.50 -4.52 -0.10 3.63 0.30 1.25 5.86];
```

figure()

```
bar(Error_pub_02)
```

```
xlabel('Test Data','FontWeight','bold','FontName','Times New Roman')
```

```
ylabel('Corrosion Rate Error (mm/year)','FontWeight','bold','FontName','Times New Roman')
```

```
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1)
```

figure()

```
subplot(1,2,1)
```

```
plot(X_01,Y_01,'k-', 'LineWidth',1.5)
```

```
set(gca,'FontName','Times New Roman')
```

```
hold on
```

```
plot(x_test_01,y_2_pub,'bx','MarkerSize',7,'LineWidth',2.5)
```

```
plot(X_02_conf,Y_02_net_pub_01,'k--','LineWidth',1.5)
plot(X_02_conf,Y_02_net_pub_02,'k--','LineWidth',1.5)
%plot(conf_int_95_01,y_2_pub,'k--','LineWidth',1.5)
%plot(conf_int_95_02,y_2_pub,'k--','LineWidth',1.5)

xlim([0 60]);
ylim([0 60]);
xlabel('Experimental Corrosion Rate (mm/year)','FontWeight','bold','FontName','Times New Roman')
ylabel('Neural Network Corrosion Rate (mm/year)','FontWeight','bold','FontName','Times New Roman')
legend(' Experiment=Model Line',' Datapoints',' 95% Confidence Interval',2)
text('Position',[5 37],'String','\itR^2\rm\bf = 0.91','FontSize',12,'FontWeight','bold','Background','w','EdgeColor','black','LineWidth',1.0,'FontName','Times New Roman');
```



```
subplot(1,2,2)
Error_pub_01=x_test_01-y_2_pub;

bar(Error_pub_02)
xlabel('Test Data','FontWeight','bold','FontName','Times New Roman')
ylabel('Corrosion Rate Error (mm/year)','FontWeight','bold','FontName','Times New Roman')
set(gca,'XMinorTick','on','XTick',[0 5 10 15 20],'LineWidth',1)

toc
```

Monte Carlo Simulation Model Code

clc, clear, close all

%The data in columns are low pressure CO₂ corrosion data from Dugstad 1994
%First column => Temperature (^oC)
%Second column => CO₂ Pressure (MPa)
%Third column => Velocity/flow (m/s)
%Fourth column => pH
%Fifth column => CO₂ Corrosion rate (mm/year)

```
Dugstad_1994_data_01=...  
[20.0000 0.0500 0.1000 4.1000 0.7000  
20.0000 0.1300 0.1000 3.8000 0.9000  
20.0000 0.1700 0.1000 3.7000 0.5000  
20.0000 0.2800 0.1000 3.7000 1.9000  
20.0000 0.3800 0.1000 3.7000 2.5000  
20.0000 0.0500 13.0000 4.1000 3.5000  
20.0000 0.0500 8.5000 4.1000 4.0000  
20.0000 0.2800 13.0000 3.7000 13.0000  
20.0000 0.2800 8.5000 3.7000 10.0000  
20.0000 0.2800 3.1000 3.7000 9.0000  
20.0000 0.1300 13.0000 3.8000 7.0000  
20.0000 0.1300 8.5000 3.8000 8.0000  
20.0000 0.1700 3.1000 3.7000 4.8000  
20.0000 0.1700 8.5000 3.7000 5.4000  
20.0000 0.1700 13.0000 3.7000 8.0000  
40.0000 0.2000 3.1000 3.8000 1.0000  
40.0000 0.2000 8.5000 3.8000 6.0000  
40.0000 0.2000 13.0000 3.8000 18.0000  
40.0000 0.0400 3.1000 4.1500 4.0000  
40.0000 0.0400 8.5000 4.1500 5.0000  
40.0000 0.0400 13.0000 4.1500 6.5000  
40.0000 0.0530 3.1000 4.1000 4.5000  
40.0000 0.0530 8.5000 4.1000 6.0000  
40.0000 0.0530 13.0000 4.1000 8.5000  
40.0000 0.0700 8.5000 4.0500 2.5000  
40.0000 0.0700 13.0000 4.0500 9.0000  
40.0000 0.7000 3.1000 3.6000 4.6000  
40.0000 0.7000 8.5000 3.6000 17.0000  
40.0000 0.7000 13.0000 3.6000 23.0000  
40.0000 1.2000 8.5000 3.4000 15.0000  
60.0000 0.2000 0.1000 3.8000 3.0000  
60.0000 0.2100 0.1000 3.8000 5.0000  
60.0000 0.2200 0.1000 3.8000 6.8000  
60.0000 0.2200 3.1000 3.8000 16.0000  
60.0000 0.2200 8.5000 3.8000 23.0000  
60.0000 0.2200 13.0000 3.8000 32.0000  
60.0000 0.2200 13.0000 3.8000 44.0000  
60.0000 1.8000 3.1000 3.4000 33.0000
```

```

60.0000  1.8000  8.5000  3.4000  60.0000
60.0000  1.8000  13.0000  3.4000  70.0000
60.0000  0.0700  3.1000  4.1500  8.8000
60.0000  0.0700  13.0000  4.1500  20.0000
60.0000  0.4000  3.1000  3.7000  30.0000
60.0000  0.4000  8.5000  3.7000  34.0000
60.0000  0.4000  13.0000  3.7000  50.0000
90.0000  0.1600  3.1000  4.1000  1.9000
90.0000  0.1600  8.5000  4.1000  8.5000
90.0000  0.1600  13.0000  4.1000  19.0000
90.0000  0.3700  3.1000  3.8000  16.0000
90.0000  0.3700  13.0000  3.8000  32.0000
90.0000  0.3700  3.1000  3.8000  17.0000
90.0000  0.3700  8.5000  3.8000  21.0000
90.0000  1.4000  3.1000  3.6000  6.8000
90.0000  1.4000  3.1000  3.6000  15.0000
90.0000  1.4000  8.5000  3.6000  30.0000
90.0000  1.4000  13.0000  3.6000  41.0000
90.0000  1.4000  8.5000  3.6000  50.0000
90.0000  2.1000  3.1000  3.5000  15.0000
90.0000  2.1000  3.1000  3.5000  38.0000
90.0000  2.1000  13.0000  3.5000  65.0000];
%Temperature          '%CO_2 Partial Pressure (MPa)' Flowrate(m/s)          'pH
%'Corrosion Rate (mm/year)  '};

X_01=Dugstad_1994_data_01(:,1:3);

x_01=Dugstad_1994_data_01(:,1);
%x_01, x_02, x_03 and x_04 are column matrices comprising temp, pressure, velocity and
%pH values respectively
x_02=Dugstad_1994_data_01(:,2);
x_03=Dugstad_1994_data_01(:,3);
x_04=Dugstad_1994_data_01(:,4);

y_01=Dugstad_1994_data_01(:,5);
%y_01 represents a column matrix comprising the Corrosion rate values

%Temperature dataset%
%normal probability plot for temperature distribution
figure()
%comparing the fit of different probability distributions against the
%temperature dataset
subplot(3,2,1)
probplot('normal',x_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

set(gca,'FontName','Times New Roman','LineWidth',1)

```

```
xlabel('Temperature (^oC)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)

subplot(3,2,2)
histfit(x_01)
xlim([0 150])
xlabel('Temperature (^oC)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Frequency', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)

%probplot('rayleigh',x_01)

%rayleigh probability plot for temperature distribution
%figure()

subplot(3,2,3)
probplot('rayleigh',x_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

set(gca, 'FontName', 'Times New Roman', 'LineWidth', 1)
xlabel('Temperature (^oC)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)

subplot(3,2,4)
histfit(x_01,4,'rayleigh')
xlim([0 150])
xlabel('Temperature (^oC)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Frequency', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)

%weibull probability plot for temperature distribution
%figure()
%comparing the fit of different probability distributions against the
%temperature dataset
subplot(3,2,5)
probplot('weibull',x_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

set(gca, 'FontName', 'Times New Roman', 'LineWidth', 1)
xlabel('Temperature (^oC)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
subplot(3,2,6)
histfit(x_01,4,'rayleigh')
xlim([0 150])
xlabel('Temperature (^oC)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Frequency', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
%PCO2 dataset%
%normal probability plot for PCO2 distribution
figure()
%comparing the fit of different probability distributions against the
%temperature dataset
```

```
subplot(3,2,1)
probplot('normal',x_02)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

set(gca, 'FontName', 'Times New Roman', 'LineWidth', 1)
xlabel('CO_2 Partial Pressure (MPa)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
subplot(3,2,2)
histfit(x_02)
xlim([0 3])
xlabel('CO_2 Partial Pressure (MPa)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Frequency', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
%probplot('rayleigh',x_01)
```

```
%rayleigh probability plot for PCO2 distribution
%figure()
```

```
subplot(3,2,3)
probplot('rayleigh',x_02)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

set(gca, 'FontName', 'Times New Roman', 'LineWidth', 1)
xlabel('CO_2 Partial Pressure (MPa)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
subplot(3,2,4)
histfit(x_02,4,'rayleigh')
xlim([0 3])
xlabel('CO_2 Partial Pressure (MPa)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
%weibull probability plot for PCO2 distribution
%figure()
%comparing the fit of different probability distributions against the
%temperature dataset
subplot(3,2,5)
probplot('weibull',x_02)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')
```

```
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('CO_2 Partial Pressure (MPa)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
subplot(3,2,6)
histfit(x_02,4,'rayleigh')
xlim([0 3])
xlabel('CO_2 Partial Pressure (MPa)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
%Velocity dataset%
%normal probability plot for Velocity distribution
figure()
%comparing the fit of different probability distributions against the
%temperature dataset
subplot(3,2,1)
probplot('normal',x_03)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Velocity (m/s)','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
subplot(3,2,2)
histfit(x_03)
xlim([0 15])
```

```
xlabel('Velocity (m/s)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)  
ylabel('Frequency', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
%probplot('rayleigh', x_01)
```

```
%exponential probability plot for Velocity distribution  
%figure()
```

```
subplot(3,2,3)  
probplot('exponential', x_03)  
%hold on  
%plot(conf_int_low_01, prob_frac_01, 'k--')  
%plot(conf_int_up_01, prob_frac_01, 'k--')
```

```
set(gca, 'FontName', 'Times New Roman', 'LineWidth', 1)  
xlabel('Velocity (m/s)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)  
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
subplot(3,2,4)  
histfit(x_03, 4, 'exponential')  
xlim([0 15])  
xlabel('Velocity (m/s)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)  
ylabel('Frequency', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
%extreme value probability plot for Velocity distribution  
%figure()  
%comparing the fit of different probability distributions against the  
%temperature dataset
```

```
subplot(3,2,5)  
probplot('extreme value', x_03)  
%hold on  
%plot(conf_int_low_01, prob_frac_01, 'k--')  
%plot(conf_int_up_01, prob_frac_01, 'k--')
```

```
set(gca, 'FontName', 'Times New Roman', 'LineWidth', 1)  
xlabel('Velocity (m/s)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)  
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
subplot(3,2,6)  
histfit(x_03, 4, 'extreme value')  
xlim([0 15])  
xlabel('Velocity (m/s)', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)  
ylabel('Frequency', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
%pH dataset%  
%normal probability plot for pH distribution  
figure()  
%comparing the fit of different probability distributions against the
```

```
%temperature dataset
subplot(3,2,1)
probplot('normal',x_04)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('pH','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)

subplot(3,2,2)
histfit(x_04)
xlim([3 5])
xlabel('pH','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)

%probplot('rayleigh',x_01)

%rayleigh probability plot for pH distribution
%figure()

subplot(3,2,3)
probplot('rayleigh',x_04)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('pH','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)

subplot(3,2,4)
histfit(x_04,4,'rayleigh')
xlim([3 5])
xlabel('pH','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)

%weibull probability plot for pH distribution
%figure()
%comparing the fit of different probability distributions against the
%temperature dataset
subplot(3,2,5)
probplot('weibull',x_04)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')
```

```
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('pH','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
subplot(3,2,6)
histfit(x_04,4,'rayleigh')
xlim([3 5])
xlabel('pH','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
%CO2 corrosion rate dataset%
%normal probability plot for Corrosion Rate distribution
figure()
%comparing the fit of different probability distributions against the
%temperature dataset
subplot(3,2,1)
probplot('normal',y_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')
```

```
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
subplot(3,2,2)
histfit(y_01,10,'normal')
xlim([0 80])
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
%probplot('rayleigh',x_01)
```

```
%Rayleigh probability plot for Corrosion Rate distribution
%figure()
```

```
subplot(3,2,3)
probplot('rayleigh',y_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')
```

```
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
```

```
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
subplot(3,2,4)
histfit(y_01,10,'rayleigh')
xlim([0 80])
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
% Weibull probability plot for Corrosion Rate distribution
%figure()
%comparing the fit of different probability distributions against the
%temperature dataset
subplot(3,2,5)
probplot('weibull',y_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')
```

```
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
subplot(3,2,6)
histfit(y_01,10,'weibull')
xlim([0 80])
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
tic
```

```
%--Monte Carlo simulation--%
sample_num_01=100000;
```

```
%-Using a uniform random distribution (rand)-%
temp_01=20+((90-20).*rand(sample_num_01,1));
pCO2_01=0.04+((2.10-0.04).*rand(sample_num_01,1));
vel_01=0.1+((13-0.1).*rand(sample_num_01,1));
pH_02=3.4+((4.15-3.4).*rand(sample_num_01,1));
corr_rate_01=0.5+((70-0.5).*rand(sample_num_01,1));
```

```
toc
```

```
%Multiple linear regression equation for CO2 Corrosion rate with T, PCO2,
%Vel and pH as the independent variables
Corr_Rate_01_test=(78.2+(0.9923.*x_01)-(2.0.*x_02)+(0.9826.*x_03)-(25.0.*x_04)...
```

```
-(7.0e-3.*x_01.^2)+(5.3011e-3.*x_01.*x_02.*x_03.*x_04));  
%Corr_rate_01_test is the original multiple linear equation derived from  
%regression analysis  
  
Corr_Rate_01_monte=abs(78.2+(0.9923.*temp_01)-(2.0.*pCO2_01)+(0.9826.*vel_01)-  
(25.0.*pH_02)...  
-(7.0e-3.*temp_01.^2)+(5.3011e-3.*temp_01.*pCO2_01.*vel_01.*pH_02));  
  
figure()  
subplot(1,2,1)  
histfit(Corr_Rate_01_monte,100,'normal')  
set(gca,'FontName','Times New Roman','LineWidth',1)  
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New  
Roman','LineWidth',1)  
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)  
xlim([0 80])  
  
sample_01=wblrnd(15.9056,0.8561,1e5,1);  
subplot(1,2,2)  
histfit(Corr_Rate_01_monte,100,'weibull')  
set(gca,'FontName','Times New Roman','LineWidth',1)  
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New  
Roman','LineWidth',1)  
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)  
xlim([0 80])  
  
%sample_01=wblrnd(12.34,1.56,1e5,1);  
  
figure()  
%comparing the fit of different probability distributions against the  
%Corrosion Rate dataset  
%subplot(3,2,1)  
subplot(1,2,1)  
pp_normal=probplot('normal',y_01);  
%probplot('normal',y_01)  
%hold on  
%plot(conf_int_low_01,prob_frac_01,'k--')  
%plot(conf_int_up_01,prob_frac_01,'k--')  
title('')  
%axes;  
set(gca,'FontName','Times New Roman','LineWidth',1)  
  
%line;  
%set(gca,'Color','b--','LineWidth',2)  
  
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New  
Roman','LineWidth',1)  
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)  
set(pp_normal(1),'Color','b') % line
```

```
set(pp_normal(2),'Color','k') % markers
%set(pp1(2),'MarkerSize',5)probplot

subplot(1,2,2)
pp_weibull=probplot('weibull',y_01);
%probplot('weibull',y_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')

%axes;
set(gca,'FontName','Times New Roman','LineWidth',1)%,'XTick',0:20:80)
%xlim([0 80])
%line;
%set(gca,'Color','b--','LineWidth',2)

%figure()
%subplot(121)
%sample = wblrnd(15.9056,0.8561,100,1);
%histfit(sample,100,'wbl')
%title('100 draws')

%subplot(122)
%sample = wblrnd(15.9056,0.8561,1e5,1);
%histfit(sample,100,'wbl')
%title('100,000 draws')

title("")
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
set(pp_weibull(1),'Color','b') % line
set(pp_weibull(2),'Color','k') % markers

%-Descriptive statistics (Mean and Standard deviation calculation)-%
mean_exp_corr=mean(y_01); %Experimental corrosion rate mean
std_exp_corr=std(y_01); %Experimental corrosion rate standard deviation

mean_03=mean(Corr_Rate_01_monte); %Mean of Monte Carlo CO2 corrosion model
std_dev_03=std(Corr_Rate_01_monte); %Standard deviation of Monte Carlo CO2 corrosion
model

%display(mean_03)
%display(std_dev_03)
```

```
fprintf('The mean and standard deviation of the Experimental CO2 Corr rate data are %.2f
mm/year and %.2f mm/year respectively \n',mean_exp_corr,std_exp_corr)
fprintf('The mean and standard deviation of the Monte Carlo CO2 Corr model are: %.2f
mm/year and %.2f mm/year respectively \n',mean_03,std_dev_03)

mean_per_error_01=((mean_03-mean_exp_corr)./mean_exp_corr).*100; %Percentage error
in means of the two results
std_per_error_01=((std_dev_03-std_exp_corr)./std_exp_corr).*100; %Percentage error in the
standard deviations of the two results

fprintf('The percentage errors in the means and standard deviations of experimental data and
the Monte Carlo model are: %.2f and %.2f respectively
\n',mean_per_error_01,std_per_error_01)

%--Sensitivity Analysis--%

percent_change_01=[-50 -30 -20 -10 -5 0 5 10 20 30 50];

temp_ave=(min(x_01)+max(x_01))./2;
pCO2_ave=(min(x_02)+max(x_02))./2;
vel_ave=(min(x_03)+max(x_03))./2;
pH_ave=(min(x_04)+max(x_04))./2;

temp_sens_01=mean(temp_01)+([-5 -3 -2 -1 -.05 0 .05 .1 .2 .3 .5].*temp_ave);
pCO2_sens_01=mean(pCO2_01)+([-5 -3 -2 -1 -.05 0 .05 .1 .2 .3 .5].*pCO2_ave);
vel_sens_01=mean(vel_01)+([-5 -3 -2 -1 -.05 0 .05 .1 .2 .3 .5].*vel_ave);
pH_sens_02=mean(pH_02)+([-5 -3 -2 -1 -.05 0 .05 .1 .2 .3 .5].*pH_ave);

for i_1=temp_sens_01

corr_rate_01_temp_change=(78.2+(0.9923.*temp_sens_01)-
(2.0.*pCO2_ave)+(0.9826.*vel_ave)-(25.0.*pH_ave)...
-(7.0e-3.*temp_sens_01.^2)+(5.3011e-
3.*temp_sens_01.*pCO2_ave.*vel_ave.*pH_ave));

end

for i_2=pCO2_sens_01

corr_rate_01_pCO2_change=(78.2+(0.9923.*temp_ave)-
(2.0.*pCO2_sens_01)+(0.9826.*vel_ave)-(25.0.*pH_ave)...
-(7.0e-3.*temp_ave.^2)+(5.3011e-3.*temp_ave.*pCO2_sens_01.*vel_ave.*pH_ave));

end
```

```
for i_3=vel_sens_01
```

```
corr_rate_01_vel_change=(78.2+(0.9923.*temp_ave)-  
(2.0.*pCO2_ave)+(0.9826.*vel_sens_01)-(25.0.*pH_ave)...  
-(7.0e-3.*temp_ave.^2)+(5.3011e-3.*temp_ave.*pCO2_ave.*vel_sens_01.*pH_ave));
```

```
end
```

```
for i_4=pH_sens_02
```

```
corr_rate_01_pH_change=(78.2+(0.9923.*temp_ave)-(2.0.*pCO2_ave)+(0.9826.*vel_ave)-  
(25.0.*pH_sens_02)...  
-(7.0e-3.*temp_ave.^2)+(5.3011e-3.*temp_ave.*pCO2_ave.*vel_ave.*pH_sens_02));
```

```
end
```

```
display(corr_rate_01_temp_change)
```

```
figure()
```

```
plot(percent_change_01,corr_rate_01_temp_change,'k-p','LineWidth',2.0)
```

```
hold on
```

```
plot(percent_change_01,corr_rate_01_pCO2_change,'g-o','LineWidth',2.0)
```

```
plot(percent_change_01,corr_rate_01_vel_change,'b-*','LineWidth',2.0)
```

```
plot(percent_change_01,corr_rate_01_pH_change,'r-x','LineWidth',2.0,'MarkerSize',10.0)
```

```
xlim([-50 50])
```

```
xlabel('Percentage Change in Input Magnitude (%)','FontName','Times New  
Roman','FontWeight','Bold')
```

```
ylabel('Corrosion Rate (mm/year)','FontName','Times New Roman','FontWeight','Bold')
```

```
legend('Temperature','CO_2 Partial Pressure','Velocity','pH')
```

```
grid on
```

```
ylim([0 80])
```

```
figure()
```

```
subplot(1,2,1)
```

```
histfit(y_01,10,'normal')
```

```
set(gca,'FontName','Times New Roman','LineWidth',1)
```

```
xlim([0 80])
```

```
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New  
Roman','LineWidth',1)
```

```
ylabel('Frequency','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
```

```
subplot(1,2,2)
```

```
%figure()
histfit(y_01,10,'weibull')
xlim([0 80])
xlabel('Corrosion Rate (mm/year)', 'FontWeight', 'Bold', 'FontName', 'Times New
Roman', 'LineWidth', 1)
ylabel('Frequency', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
figure()
%comparing the fit of different probability distributions against the
%Corrosion Rate dataset
%subplot(3,2,1)
subplot(1,2,1)
pp1=probplot('normal',y_01);
%probplot('normal',y_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')
title("")
%axes;
set(gca, 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
%line;
%set(gca, 'Color', 'b--', 'LineWidth', 2)
```

```
xlabel('Corrosion Rate (mm/year)', 'FontWeight', 'Bold', 'FontName', 'Times New
Roman', 'LineWidth', 1)
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
set(pp1(1), 'Color', 'k') % line
set(pp1(2), 'MarkerSize', 4, 'Color', 'b') % markers
%set(pp1(2), 'MarkerSize', 5)probplot
```

```
subplot(1,2,2)
pp1=probplot('weibull',y_01);
%probplot('normal',y_01)
%hold on
%plot(conf_int_low_01,prob_frac_01,'k--')
%plot(conf_int_up_01,prob_frac_01,'k--')
```

```
%axes;
set(gca, 'FontName', 'Times New Roman', 'LineWidth', 1)
```

```
%line;
%set(gca, 'Color', 'b--', 'LineWidth', 2)
title("")
xlabel('Corrosion Rate (mm/year)', 'FontWeight', 'Bold', 'FontName', 'Times New
Roman', 'LineWidth', 1)
ylabel('Probability', 'FontWeight', 'Bold', 'FontName', 'Times New Roman', 'LineWidth', 1)
set(pp1(1), 'Color', 'k') % line
```

```
set(pp1(2),'MarkerSize',4,'Color','b') % markers

%figure()
%[f,x_values,flo,fup] = ecdf(y_01);
%F=plot(x_values,f);
%set(F,'LineWidth',2,'k');
%hold on

figure()
%[H_01,STAT_01]=cdfplot(y_01);

%G=plot(x_values,normcdf(x_values,0,1),'b-');
%set(G,'LineWidth',2);
%legend([F G],...
%  'Empirical CDF','Standard Normal CDF',...
%  'Location','SE');
%display(STAT_01)

set(gca,'FontName','Times New Roman','LineWidth',1)
cdfplot(y_01)
title("")
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
ylabel('F(x)','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
grid off

figure()
%[f_monte,x_values_monte,flo_monte,fup_monte] = ecdf(Corr_Rate_01_monte);
%plot(x_values_monte,f_monte)
%hold on

%[H_03,STAT_03]=cdfplot(Corr_Rate_01_monte);
%display(STAT_03)

set(gca,'FontName','Times New Roman','LineWidth',1)
cdfplot(Corr_Rate_01_monte)
title("")
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
ylabel('F(x)','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
grid off

%test_cdf= [y_01,cdf('normal',17.03,17.02,1)];

%[H,P,KSSTAT,CV] = kstest(y_01,'cdf',test_cdf,0.05,'unequal');
```

```
[H,P,KSSTAT,CV] = kstest(y_01); %Carrying out a Kolmogorov-Smirnov (kstest) normality  
test on the experimental CO2 corrosion rate data(y_01)
```

```
display(H) %H is the hypothesis, when H=0, accept the nul hypothesis; when H=1, reject the  
null hypothesis (hence data does not follow Normal distribution)  
display(P) %P is the p-value of the test (asymptotic p-value)  
display(KSSTAT) %KSSTAT is the magnitude of the kstest statistic  
display(CV) %CV is the critical Value
```

```
[H_02,P_02,KSSTAT_02,CV_02] = kstest(Corr_Rate_01_monte); %Carrying out a  
Kolmogorov-Smirnov (kstest) normality test on the Monte Carlo-simulated CO2 corrosion  
rate data(Corr_Rate_01_monte)
```

```
display(H_02) %H is the hypothesis, when H=0, accept the nul hypothesis; when H=1, reject  
the null hypothesis (hence data does not follow Normal distribution)  
display(P_02) %P is the p-value of the test (asymptotic p-value)  
display(KSSTAT_02) %KSSTAT is the magnitude of the kstest statistic  
display(CV_02) %CV is the critical Value
```

```
%R_01=wblrnd(31.30,2.236,[10000,1]);
```

```
%[H_03,P_03,KSSTAT_03] = kstest2(Corr_Rate_01_monte,R_01,0.05,'unequal');
```

```
%display(H_03) %H is the hypothesis, when H=0, accept the nul hypothesis; when H=1,  
reject the null hypothesis (hence data does not follow Normal distribution)  
%display(P_03) %P is the p-value of the test (asymptotic p-value)  
%display(KSSTAT_03) %KSSTAT is the magnitude of the kstest statistic  
%display(CV_03) %CV is the critical Value
```

```
figure()  
% Output fitted probablility distributions: PD1,PD2,PD3
```

```
% Data from dataset "Corr_Rate_01_monte data":  
% Y = Corr_Rate_01_monte
```

```
% Force all inputs to be column vectors  
Corr_Rate_01_monte = Corr_Rate_01_monte(:);
```

```
% Prepare figure  
clf;  
hold on;  
LegHandles = []; LegText = {};
```

```
% --- Plot data originally in dataset "Corr_Rate_01_monte data"  
[CdfF,CdfX] = ecdf(Corr_Rate_01_monte,'Function','cdf'); % compute empirical cdf  
BinInfo.rule = 1;  
[~,BinEdge] = internal.stats.histbins(Corr_Rate_01_monte,[],[],BinInfo,CdfF,CdfX);
```

```
[BinHeight,BinCenter] = ecdfhist(CdfF,CdfX,'edges',BinEdge);
hLine = bar(BinCenter,BinHeight,'hist');
set(hLine,'FaceColor','none','EdgeColor',[0 0 0],...
    'LineStyle','-','LineWidth',1);
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1);
ylabel('Density','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
%LegHandles(end+1) = hLine;
%LegText{end+1} = 'Corr_Rate_01_monte data';

% Create grid where function will be computed
XLim = get(gca,'XLim');
XLim = XLim + [-1 1] * 0.01 * diff(XLim);
XGrid = linspace(XLim(1),XLim(2),100);

xlim([0 80])
% --- Create fit "fit 2"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
% pd1 = ProbDistUnivParam('weibull',[ 29.69797925362, 2.093948068127])
pd1 = fitdist(Corr_Rate_01_monte, 'weibull');
% This fit does not appear on the plot

% --- Create fit "Weibull"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
% pd2 = ProbDistUnivParam('weibull',[ 29.69797925362, 2.093948068127])
pd2 = fitdist(Corr_Rate_01_monte, 'weibull');
YPlot = pdf(pd2,XGrid);
hLine = plot(XGrid,YPlot,'Color',[0 0 1],...
    'LineStyle','-','LineWidth',2.5,...
    'Marker','none','MarkerSize',6);
LegHandles(end+1) = hLine;
LegText{end+1} = 'Weibull';

% --- Create fit "Normal"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
% pd3 = ProbDistUnivParam('normal',[ 26.44988114611, 12.93956295892])
pd3 = fitdist(Corr_Rate_01_monte, 'normal');
YPlot = pdf(pd3,XGrid);
hLine = plot(XGrid,YPlot,'Color',[0.666667 0.333333 0],...
    'LineStyle','-','LineWidth',2.5,...
    'Marker','none','MarkerSize',6);
LegHandles(end+1) = hLine;
LegText{end+1} = 'Normal';
```

```
% Adjust figure
box on;
hold off;

% Create legend from accumulated handles and labels
hLegend = legend(LegHandles,LegText,'Orientation','vertical','Location','NorthEast');
set(hLegend,'Interpreter','none');

figure()
% Output fitted probability distributions: PD1,PD2,PD3

% Data from dataset "Corr_Rate_01_monte data":
% Y = Corr_Rate_01_monte

% Force all inputs to be column vectors
Corr_Rate_01_monte_cdf = Corr_Rate_01_monte(:);

% Prepare figure
%clf;
hold on;
LegHandles_01 = []; LegText_01 = {};

% --- Plot data originally in dataset "Corr_Rate_01_monte data"
[CdfY_01,CdfX_01] = ecdf(Corr_Rate_01_monte_cdf,'Function','cdf'); % compute empirical
function
hLine_01 = stairs(CdfX_01,CdfY_01,'Color',[0 0 0],'LineStyle','-','LineWidth',2);
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1);
ylabel('Cumulative probability','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1)
LegHandles_01(end+1) = hLine_01;
LegText_01{end+1} = 'Corr_Rate_01_monte data';

% Create grid where function will be computed
XLim_01 = get(gca,'XLim');
XLim_01 = XLim_01 + [-1 1] * 0.01 * diff(XLim_01);
XGrid_01 = linspace(XLim_01(1),XLim_01(2),100);
xlim([0 80])

% --- Create fit "fit 2"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
% pd1 = ProbDistUnivParam('weibull',[ 29.69797925362, 2.093948068127])
pd4 = fitdist(Corr_Rate_01_monte_cdf, 'weibull');
```

```
% This fit does not appear on the plot

% --- Create fit "Weibull"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%   pd2 = ProbDistUnivParam('weibull',[ 29.69797925362, 2.093948068127])
pd5 = fitdist(Corr_Rate_01_monte_cdf, 'weibull');
YPlot_01 = cdf(pd5,XGrid_01);
hLine_01 = plot(XGrid_01,YPlot_01,'Color',[0 0 1],...
    'LineStyle','-','LineWidth',2,...
    'Marker','none','MarkerSize',6);
LegHandles_01(end+1) = hLine_01;
LegText_01{end+1} = 'Weibull';

% --- Create fit "Normal"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%   pd3 = ProbDistUnivParam('normal',[ 26.44988114611, 12.93956295892])
pd6 = fitdist(Corr_Rate_01_monte_cdf, 'normal');
YPlot_01 = cdf(pd6,XGrid_01);
hLine_01 = plot(XGrid_01,YPlot_01,'Color',[0.666667 0.333333 0],...
    'LineStyle','-','LineWidth',2,...
    'Marker','none','MarkerSize',6);
LegHandles_01(end+1) = hLine_01;
LegText_01{end+1} = 'Normal';

% Adjust figure
box on;
hold off;

% Create legend from accumulated handles and labels
hLegend_01 = legend(LegHandles_01,LegText_01,'Orientation','vertical','Location',
'SouthEast');
set(hLegend_01,'Interpreter','none');

%Corr_rate_01_monte_weib=wblrnd(17.136,1.0149,[sample_num_01 1]);

%figure()
%histfit(Corr_rate_01_monte_weib,100,'weibull')

%display(mean(Corr_rate_01_monte_weib))
%display(std(Corr_rate_01_monte_weib))

Sim_run_01=[1e7 1e6 1e5 1e4 1e3]; %Simulation Run to determine best number for Monte
Carlo simulations

Time_01=[0.9657 0.0979 0.0094 0.0008 0.0001]; %Time-taken to run Monte Carlo
simulations
```

```
figure()

semilogx(Sim_run_01,Time_01,'b-*','LineWidth',2)
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Number of Monte Carlo Simulations','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1);
ylabel('Time Elapsed (s)','FontWeight','Bold','FontName','Times New Roman','LineWidth',1')

hold on
text(0.3e4,0.3,'Chosen Number of Simulation Runs','FontName','Times New
Roman','FontWeight','Bold')
annotation('arrow',[0.4289 0.5125],...
    [0.3143 0.1310]);

figure()
% Output fitted probability distributions: PD1,PD2

% Data from dataset "y_01 data":
% Y = y_01

% Force all inputs to be column vectors
y_01 = y_01(:);

% Prepare figure
clf;
hold on;
LegHandles_02 = []; LegText_02 = { };

% --- Plot data originally in dataset "y_01 data"
[CdfF,CdfX] = ecdf(y_01,'Function','cdf'); % compute empirical cdf
BinInfo.rule = 1;
[~,BinEdge] = internal.stats.histbins(y_01,[],[],BinInfo,CdfF,CdfX);
[BinHeight,BinCenter] = ecdfhist(CdfF,CdfX,'edges',BinEdge);
hLine_02 = bar(BinCenter,BinHeight,'hist');
set(hLine_02,'FaceColor',[0 0 0.8],'EdgeColor',[0 0 0],...
    'LineStyle','-','LineWidth',1.5);
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1);
ylabel('Density','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
% xlabel('Data');
% ylabel('Density')
% LegHandles_02(end+1) = hLine_02;
% LegText_02{end+1} = 'y_01 data';

% Create grid where function will be computed
XLim_02 = get(gca,'XLim');
XLim_02 = XLim_02 + [-1 1] * 0.01 * diff(XLim_02);
```

```
XGrid_02 = linspace(XLim_02(1),XLim_02(2),100);
xlim([0 80])

% --- Create fit "Weibull"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
% pd1 = ProbDistUnivParam('weibull',[ 17.13597550619, 1.014898003019])
pd7 = fitdist(y_01, 'weibull');
YPlot_02 = pdf(pd7,XGrid_02);
hLine_02 = plot(XGrid_02,YPlot_02,'Color',[1 0 0],...
    'LineStyle','-','LineWidth',2,...
    'Marker','none','MarkerSize',6);
LegHandles_02(end+1) = hLine_02;
LegText_02{end+1} = 'Weibull';

% --- Create fit "Normal"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
% pd2 = ProbDistUnivParam('normal',[ 17.026666666667, 17.01911056118])
pd8 = fitdist(y_01, 'normal');
YPlot_02 = pdf(pd8,XGrid_02);
hLine_02 = plot(XGrid_02,YPlot_02,'Color',[0 0 0],...
    'LineStyle','--','LineWidth',2,...
    'Marker','none','MarkerSize',6);
LegHandles_02(end+1) = hLine_02;
LegText_02{end+1} = 'Normal';

% Adjust figure
box on;
hold off;

% Create legend from accumulated handles and labels
hLegend_02 = legend(LegHandles_02,LegText_02,'Orientation','vertical','Location',
'NorthEast');
set(hLegend_02,'Interpreter','none');

figure()
% Output fitted probability distributions: PD1,PD2

% Data from dataset "y_01 data":
% Y = y_01

% Force all inputs to be column vectors
y_01 = y_01(:);

% Prepare figure
clf;
```

```
hold on;
LegHandles_03 = []; LegText_03 = { };

% --- Plot data originally in dataset "y_01 data"
hLine_03 = probplot('normal',y_01,[],[],'noref');
set(hLine_03,'Color',[0 0 0],'Marker','o','MarkerSize',6,'LineWidth',2);
set(gca,'FontName','Times New Roman','LineWidth',1)
xlabel('Corrosion Rate (mm/year)','FontWeight','Bold','FontName','Times New
Roman','LineWidth',1);
ylabel('Probability','FontWeight','Bold','FontName','Times New Roman','LineWidth',1)
title("")
%LegHandles_03(end+1) = hLine_03;
%LegText_03{end+1} = 'y_01 data';

% --- Create fit "Weibull"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
% pd1 = ProbDistUnivParam('weibull',[ 12.02705976236, 2.406601738499])
pd9 = fitdist(y_01, 'weibull');
hLine_03 = probplot(gca,pd9);
set(hLine_03,'Color',[1 0 0],'LineStyle','-','LineWidth',2);
LegHandles_03(end+1) = hLine_03;
LegText_03{end+1} = 'Weibull';

% --- Create fit "Normal"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
% pd2 = ProbDistUnivParam('normal',[ 10.78125, 4.746257297422])
pd10 = fitdist(y_01, 'normal');
hLine_03 = probplot(gca,pd10);
set(hLine_03,'Color',[0 0 1],'LineStyle','-','LineWidth',2);
LegHandles_03(end+1) = hLine_03;
LegText_03{end+1} = 'Normal';

% Adjust figure
box on;
hold off;

% Create legend from accumulated handles and labels
hLegend_03 = legend(LegHandles_03,LegText_03,'Orientation', 'vertical');
set(hLegend_03,'Units','normalized');
Position = get(hLegend_03,'Position');
Position(1:2) = [0.153828,0.675584];
set(hLegend_03,'Interpreter','none','Location','NorthWest');
```

%Optimising the Mean and Standard Deviation of the Monte Carlo dataset

%Soln_01 is the difference between the mean of the Monte Carlo dataset and
%that of the Experimental corrosion rate dataset

```
a_01=mean(y_01);  
a_02=mean(Corr_Rate_01_monte);
```

```
b_01=std(y_01);  
b_02=std(Corr_Rate_01_monte);
```

```
%Soln_01=fzero(@mean_optim_01,5,optimset('TolFun',1e-12));  
%Soln_02=fzero(@stand_dev_optim_01,5,optimset('TolFun',1e-12));
```

```
%figure()  
%histfit(Corr_Rate_01_monte,100,'Weibull')
```

```
 %[x_pH,fval,exitflag,output] = Auto_gen_01(x0)  
 %% This is an auto generated MATLAB file from Optimization Tool.
```

```
%% Start with the default options
```

```
options = optimset;
```

```
%% Modify options setting
```

```
options = optimset(options,'Display', 'iter');
```

```
options = optimset(options,'PlotFcns', { @optimplotx @optimplotfunccount @optimplotfval  
});
```

```
[x_vel,fval_vel,exitflag_vel,output_vel] = ...
```

```
fzero(@mean_optim_02,5.0,options);
```

```
display(x_vel)
```

```
display(fval_vel)
```

```
display(exitflag_vel)
```

```
display(output_vel)
```

```
%% This is an auto generated MATLAB file from Optimization Tool.
```

```
%% Start with the default options
```

```
options = optimset;
```

```
%% Modify options setting
```

```
options = optimset(options,'Display', 'iter');
```

```
options = optimset(options,'PlotFcns', { @optimplotx @optimplotfunccount @optimplotfval  
});
```

```
[x_vel_01,fval_vel_01,exitflag_vel_01,output_vel_01] = ...
```

```
fzero(@stand_dev_optim_02,5.0,options);
```

```
display(x_vel_01)  
display(fval_vel_01)  
display(exitflag_vel_01)  
display(output_vel_01)
```

References

Abyaneh, Z. (2014). *Evaluation of Multivariate Linear Regression and Artificial Neural Networks in Prediction of Water Quality Parameters*, Journal of Environmental Health Science & Engineering, Volume 12, Issue 40.

Ali, S., Smith, K. A. (2006). *On Learning Algorithm Selection for Classification*, Applied Soft Computing, Vol. 6, pp. 119-138.

Alsmadi, M. K. S., Omar, K. B., Noah, S. A. (2009). *Back Propagation Algorithm: The Best Algorithm among the Multi-layer Perceptron Algorithm*, International Journal of Computer Science and Network Security, Vol. 9(4), 2009, pp. 378 – 383.

Al-Fattah, S.M. and Startzman, R.A. (May 2003). *Neural Network Approach Predicts U.S. Natural Gas Production*, SPE production and Facilities, Vol. 18, No. 2, pp. 84-91.

American Petroleum Institute. (1981). *Recommended Practice for Design and Installation of offshore Production Platform Piping Systems*, American Petroleum Institute, Recommended Practice 14E (API RP 14E), Washington DC, USA.

Baboian, R. (eds.), Dean Jr., S. W. D. (eds.), Hack, H. P. (eds.), Hibner, E. L. (eds.), Scully, J. R. (eds.) (2005). *Corrosion Tests and Standards: Application and Interpretation*, 2nd Edition, ASTM International, 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959, Chapter 5: Statistical Treatment of Data, Data Interpretation and Reliability, Pages 84, 85.

Beale, M. H., Hagan, M. T., Demuth H. B. (2014). *Neural Network Toolbox User's Guide R2014b*. The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098. 2014, Pages 2-4, 2-5, 2-17, 2-18, 2-19.

Belohlavek, R. (ed.) and Klir, G. (ed.) (2011). *Concepts and Fuzzy Logic*, Massachusetts Institute of Technology, USA.

Benning, L. G., Wilkin, R. T., Barnes, H. L. (2000). *Chemical Geology*, 167, 25-51.

Bondos, J. C., Reddy, R. V., Pugh, D. V. and Norman, D. A., Pacheco, J. L. and Nelson, J. L. (2007). *Accurate Corrosion Prediction through an Integrated Approach*, SPE Production & Operations: 2007, Vol. 22, No. 2, pp. 161-167.

Campbell, B. [Future Refining Capacity Changes, Chapter 3] [in *Annual Outlook for Oil and Gas 1991*], Energy Information Administration (EIA), Office of Oil and Gas, U.S. Department of Energy (DOE), Washington DC 20585, USA. Page 42.

Carberry, J., J. (2001). *Chemical and Catalytic Reaction Engineering*, Dover Edition, General Publishing Company, Ltd., 30 Lesmill Road, Don Mills, Toronto, Ontario, Canada, p. 383.

Chatterjee, S., Hadi, A., S. (1986). *Influential Observations, High Leverage Points, and Outliers in Linear Regression*, Statistical Science, Vol. 1, No. 3, pp. 379-416.

Choi, Y-S., Nescic, S. (2009). *Corrosion Behavior of Carbon Steel in Supercritical CO₂-Water Environments*, NACE International, Corrosion 2009, Paper 09256, Atlanta, Georgia, USA.

Choi, Y-S, Nešić, S. (2011). *Determining the Corrosive Potential of CO₂ Transport Pipeline in High pCO₂-Water Environments*, International Journal of Greenhouse Gas Control, Volume 5, Issue 4, pp. 788-797.

Choi, Y-S., Nesic, S. and Young, D. (2010). *Effect of Impurities on the Corrosion Behavior of CO₂ Transmission Pipeline Steel in Supercritical CO₂-Water Environments*, Environmental Science and Technology, Vol. 44, Issue 23, pp. 9233-9238.

Churchland, P. S. and Sejnowski, T. J. (1992). *The Computational Brain*, The Massachusetts Institute of Technology (MIT), The MIT Press, Cambridge, Massachusetts, USA, Chapter 3, Pages 107-115.

Clarke, L., Edmonds, J., Jacoby, H., Pitcher, H., Reilly, J., Richels, R. (2007). *Scenarios of Greenhouse Gas Emissions and Atmospheric Concentrations*. Sub-report 2.1A of Synthesis and Assessment Product 2.1 by the U.S. Climate Change Science Program and the Subcommittee on Global Change Research. Department of Energy, Office of Biological & Environmental Research, Washington, 7 DC., USA, 154 pp. 59-86.

Criaud, A., Fouillac, C. and Marty, B. (1989). *Low Enthalpy Geothermal fluids from the Paris basin. 2—Oxidation-reduction state and consequences for the prediction of corrosion and sulfide scaling*, Geothermics, Vol. 18, Issues 5-6, 711-727.

Cottis, R. A., Qing, L., Owen, G., Gartland, S. J., Helliwell, I. A., Turega, M. (1999). *Neural Network methods for Corrosion Data Reduction*, Materials and Design: 1999, Vol. 20, No. 4, pp. 169-178.

CO₂ pipeline Infrastructure: An analysis of global challenges and opportunities report, International Energy Agency Greenhouse Gas (IEA GHG), Conference /2010, (Cambridge, UK, 2010).

Crolet, J. L., Bonis, M. R. (1991). *Prediction of the Risks of CO₂ Corrosion in Oil and Gas Wells*, SPE Production Engineering, Society of Petroleum Engineers (SPE), Volume 6, Number 4, pp. 449-453.

Crolet, J. L. (1994). "Which CO₂ Corrosion, Hence Which Prediction?," in *Predicting CO₂ Corrosion in the Oil and Gas Industry*, European Federation of Corrosion Publication no. 13 (London, U.K.: Institute of Materials, 1994), p 1.

Crolet, J-L., Thevenot, N., Dugstad, A. (1999). *Role of Free Acetic Acid on the CO₂ Corrosion of Steels*, CORROSION 99, Paper 99024, NACE International, San Antonio, Texas, USA.

Crolet, J. L. (2002). *Corrosion in Oil and Gas Production*, in *Corrosion and Anticorrosion*, eds. G. Beranger, H. Mazille (Paris, France: Hermes Science, 2002).

Crolet, J. L., Thevenot, N., Nestic, S. (1996). *Role of Conductive Corrosion Products on the Protectiveness of Corrosion Layers*, Corrosion 96, paper no. 4 (Houston, TX: NACE, 1996).

Cui, Z. D., Wu, S. L., Zhu, S. L., Yang, X. J. (2006), *Study on Corrosion Properties of Pipelines in Simulated Produced Water with Supercritical CO₂*, Journal of Applied Surface Science, Volume 252, Issue 6, pp.2368-2374.

Demuth, H., Beale, M., Hagan, M. (2009). *Neural Network Toolbox 6 User's Guide*, The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, Pages 2-4, 2-5, 2-17, 2-18, 2-19.

De Waard, C., Lotz, U. and Dugstad, A. (1995). *Influence of Liquid Flow Velocity on CO₂ Corrosion: A Semi-Empirical Model*, CORROSION/95, Paper 128, NACE International, Houston, Texas, USA.

De Waard, C., Lotz, U. and Milliams, D. E. (1991). *Predictive model for CO₂ Corrosion Engineering in Wet Natural Gas Pipelines*, Corrosion: 1991, Vol. 47, No. 12, pp. 976-985.

Dlugokencky, E. J., Hall, B. D., Montzka, S. A., Dutton, G., Muhle, J., and Elkins, J. W. (2014): [Global climate] Long-lived greenhouse gases [in *State of the Climate in 2013*]. Bulletin of the American Meteorological Society, Volume 95, No.7, S33–S34.

Downie, M., Race, J., Seevam, P. (2007). *Transport of CO₂ for Carbon Capture and Storage in the UK*, Offshore Europe 2007, Society of Petroleum Engineers (SPE), Aberdeen, Scotland, UK.

Draper, N. R.; Smith, H. (1998). *Applied Regression Analysis*. Wiley-Interscience, Chapter 1, pp. 15-20, 40-44.

Dugstad, A. (1998). *Mechanism of Protective Film Formation during CO₂ Corrosion of Carbon Steel*, Corrosion 98, Paper no. 31, NACE International, San Diego, California, USA.

Dugstad, A., Lunde, L. and Nesic, S. (1994a). *Control of Internal Corrosion in Multi-phase Oil and Gas Pipelines*, Prevention of Pipeline Corrosion Conference, Pipe Line Industry and Pipes & Pipelines International, Houston, Texas, USA.

Dugstad, A., Lunde, L. and Videm, K. (1994b). *Parametric Study of CO₂ Corrosion of Carbon Steel*, Corrosion/94, Paper no. 14, (Houston, TX: NACE International, 1994).

Eriksrud, E., Sjøntvedt, T. (1983). *Effect of Flow on CO₂ Corrosion Rates in Real and Synthetic Formation Waters*, Paper 44, NACE International.

Fahidy, T. Z. (2005). *Electrochemical Horizons for the Poisson-Lognormal Distribution of Probability Theory*, Journal of Electroanalytical Chemistry, Volume 581, Issue 1, pp. 11-15.

Fujino, J., Nair, R., Kainuma, M., Masui, T. and Matsuoka, Y. (2006). *Multi-gas Mitigation Analysis on Stabilization Scenarios Using Aim Global Model*, The Energy Journal, Special Issue: Multi-Greenhouse Gas Mitigation and Climate Policy, Volume 27, pp. 344-354.

Fuzzy Logic Toolbox User's Guide Matlab R2015b, (2015). The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098.

Garverick, L. (eds.) (1994). *Corrosion in the Petrochemical Industry*, ASM International, Section 5: Corrosion in Petroleum Production Operations, pages 280, 281 and 285.

Giourntas, L., Hodgkiess, T., Galloway, A. M. (2015). *Enhanced Approach of Assessing the Corrosive Wear of Engineering Materials under Impingement*, Wear, Volumes 338-339, pp. 155-163.

Grau, R., M. (2000). *Mechanisms for Improved Corrosion Resistance of Steels in Carbonic Acid Environments*, PhD Thesis, University of Leeds.

Gray, L. G. S., Anderson, B. G., Danysh, M. J and Tremaine, P. R. (1989). *Mechanism of Carbon Steel Corrosion in Brines Containing Dissolved Carbon Dioxide at pH 4*, Corrosion/89, Paper no. 464, NACE International, Houston, Texas, USA.

Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., Schmidhuber, J. (2009). *A Novel Connectionist System for Improved Unconstrained Handwriting Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 31, No. 5, pp. 855-868.

Gronh lm, T., Annila, A. (2007). *Natural Distribution*, Mathematical Biosciences, Volume 210, Issue 2, pp. 659-667.

Guo, X. P., Chen, Z. Y. and Liu, D. (2005). *The Effect of Acetic Acid and Acetate on CO₂ Corrosion of Carbon Steel*, CORROSION 2005, Paper 05306, NACE International, Houston, Texas, USA.

<https://co2corrosionchem409.wikispaces.com/Background+of+CO2+Corrosion>

Hajizadeh, Y. (2006). *Fuzzy Corrosion Modeling, New Horizons, New Opportunities*, SPE Eastern Regional Meeting, Paper 104298, Society of Petroleum Engineers (SPE), Canton, Ohio, USA.

Hajizadeh, Y. (2007). *Neural Networks Can Enhance Fuzzy Corrosion Modeling*, Society of Petroleum Engineers, SPE Annual Technical Conference, Society of Petroleum Engineers (SPE), Anaheim, California, USA.

Halvorsen, A. M. K., S ntvedt, T. (1999). *CO₂ Corrosion Model for Carbon Steel Including a Wall Shear Stress Model for Multiphase Flow and Limits for Production Rate to Avoid Mesa Attack*, Corrosion 1999, Paper 99042, NACE International, San Antonio, Texas, USA.

Harrison, R. L. (2010). *Introduction to Monte Carlo Simulation*, AIP Conference Proceedings, Volume 1204, Number 17, Pages 17-21.

Haykin, S. (1999). *Neural Networks – A Comprehensive Foundation*, Pearson Education Ltd., Second Edition, Chapter 4, pp. 234-244.

Hays, G. F. (2011). *Now is the Time*, World Corrosion Organization (WCO), pp.1-2.

Helliwell, I., Turega, M., Cottis, R. A. (1996). *Neural networks for corrosion data reduction*, Corrosion/96, paper 379, NACE, 1996.

Hernandez, S., Nesic, S., Weckman, G. and Ghai, V. (2006). *Use of Artificial Neural Networks for Predicting Crude Oil Effect on Carbon Dioxide Corrosion of carbon Steels*, Corrosion: 2006, Vol. 62, No. 06, pp. 467-482.

Hesjevik, S. M., Olsen, S., Seiersten, M. (2003). *Corrosion at High CO₂ Pressure*, StatOil and Institute for Energy Technology in Norway, NACE International, Corrosion 2003, Paper 03345, San Diego, California, USA.

Hodgkiess, T., Neville, A., Shrestha, S. (1999). *Electrochemical and Mechanical Interactions during Erosion-Corrosion of a High-Velocity Oxy-fuel Coating and a Stainless Steel*, Wear, Volumes 233-235, pp. 623-634.

Hu, X., Barker, R. J., Neville, A., Gnanavelu, A. (2011). *Case Study on Erosion-Corrosion Degradation of Pipework Located on an Offshore Oil and Gas Facility*, Wear, Volume 271, Issues 9-10, pp. 1295-1301.

International Energy Agency (IEA), 2010: World Energy Outlook 2010, Chapter 14: The energy transformation by sector, pp. 418.

International Energy Agency (IEA), 2012: World Energy Outlook 2012, Chapter 2: Energy Trends to 2035, pp. 49-80.

Ikeda, A., Mukai, S., Ueda, M. (1984). *Prevention of CO₂ Corrosion of Line Pipe and Oil Country Tubular Goods*, Corrosion 84, paper no. 289 (Houston, TX: NACE, 1984).

IPCC, 2014: *Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change* [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)]. IPCC, Geneva, Switzerland, 151, pp. 46, 47.

IPCC, 2007: *Climate Change 2007: Synthesis Report. Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change* [Core Writing Team, Pachauri, R.K and Reisinger, A. (eds.)]. IPCC, Geneva, Switzerland, 104, pp. 17, 60.

Jackson, J. E., *A User's Guide to Principal Components*, John Wiley and Sons, 1991, p. 592.

James, H. L., Fleischer, M. (ed.). (1966). [Chemistry of the Iron-rich Sedimentary Rocks, Chapter W] [in *Data of Geochemistry*], Sixth Edition, U.S. Geological Survey Professional Paper 440-W, United States Government Printing Office, Washington DC 20402, USA. Pages W9, W10 and W11.

Johnson, M. L., & Tomson, M. B. (1991). *Ferrous Carbonate Precipitation Kinetics And its Impact on CO₂ Corrosion*, NACE International, Corrosion 1991, Paper No. 268, Houston, Texas, USA.

Johnson, N. L., Kotz, S., Balakrishnan, N. (1994). *Continuous Univariate Distributions*, Volume 1, Second Edition, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. A Wiley-Interscience Publication, John Wiley & Sons, Inc., New York, pp. 207-209, 628, 629.

Jolliffe, I. T., *Principal Component Analysis*, 2nd edition, Springer, 2002.

Kaufmann, A. and Gupta, M. M. (1985). *Introduction to Fuzzy Arithmetic Theory and Applications*, Van Nostrand Reinhold Electrical/Computer Science and Engineering Series, Van Nostrand Reinhold Company, New York, USA.

Kermani, M. B. and Morshed, A. (2003). *Carbon Dioxide Corrosion in Oil and Gas Production – A Compendium*, Corrosion: August 2003, Vol.59, No.8, pp. 659-683.

Kermani, M. B., Gonzales, J. C., Linne, C., Dougan, M., Cochrane, R. (2001). *Development of Low-Carbon Cr-Mo Steels with Exceptional Corrosion Resistance for Oilfield Applications*, Corrosion /2001, paper no. 01065 (Houston, TX: NACE, 2001).

Kermani, M. B., Gonzales, J. C., Turconi, G. L., Edmonds, D., Dicken, G., Scoppio, L. (2003). *Development of Superior Corrosion Resistance 3%Cr Steels for Downhole Applications,*” Corrosion/ 2003, Paper no. 03116 (Houston, TX: NACE, 2003).

Kermani, M. B., Smith, L. M. (1994). *Predicting CO₂ Corrosion in the Oil and Gas Industry*, European Federation of Corrosion Publication no. 13 (London, U.K.: Institute of Materials, 1994).

Koch, G. H., Brongers, M. P. H., Thompson, N. G., Virmani, Y. P., Payer, J. H. (2002). *Corrosion Costs and Preventive Strategies in the United States*, U.S Department of Transportation: Federal Highway Administration (FHWA), CC Technologies, NACE International. Publication No. FHWA-RD-01-156.

Krogh, A. (2008). *What are artificial neural networks?*, Nature Biotechnology, Nature Publishing Group, Vol. 26, No. 2, pp. 195-197.

Lee, E. T., Wang, J. W. (2003). *Statistical Methods for Survival Data Analysis*, Third Edition, Wiley-Interscience, A John Wiley & Sons, Inc., Publication, Hoboken, New Jersey, USA, Chapters 6, 9, Pages 138, 143, 144, 221.

Lyle, F., Schutt, H. U. (1998). *CO₂/H₂S Corrosion under Wet Gas Pipeline Conditions in the Presence of Bicarbonate, Chloride and Oxygen*, CORROSION 98, Paper 98011, NACE International, San Diego, California, USA.

Mamdani, E., H. and Assilian, S. (1975). *An experiment in linguistic synthesis with a fuzzy logic controller*, International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13.

Man, H. C., and Gabe, D. R. (1981). *The Determination of Pitting Potentials*, Corrosion Science, Volume 21, Issue 4, Pages 323-326.

Matsumoto, M., Nishimura, T. (1998). *Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator*, ACM Transactions on Modelling and Computer Simulation, Vol. 8, No.1, pp. 3-30.

Meinshausen, M., Smith, S. J., Calvin, K., Daniel, J. S., Kainuma, M. L. T., Lamarque, J-F., Matsumoto, K., Montzka, S. A., Raper, S. C. B., Riahi, K., Thomson, A., Velders, G. J.M., van Vuuren, D. P. P. (2011). *The RCP Greenhouse Gas Concentrations and their Extensions from 1765 to 2300*, Climatic Change, Volume 109, Issue 1, pp. 213-241.

Mohamed, M. F., Nor, A. M., Suhor, M. F., Singer, M., Choi, Y. S., Nestic, S. (2011). *Water Chemistry For Corrosion Prediction in High Pressure CO₂ Environments*, Corrosion 2011, Paper 11375, NACE International, Houston, Texas, USA.

Naiman, B. (1948). *The Brönsted Concept of Acids and Bases in Quantitative Analysis*, Journal of Chemical Education, Volume 25, Issue 8, pp. 454-456.

Nesic, S. and Vrhovac, M. (1999). *A Neural Network Model for CO₂ Corrosion of Carbon Steel*, The Journal of Corrosion Science and Engineering (JCSE), Vol. 1, No. 6, ISSN 1466-8858.

Nesic, S., Cai, J., Lee, K. L. J. (2005). *A Multiphase Flow and Internal Corrosion Prediction Model for Mild Steel Pipelines*, Corrosion 2005, Paper no. 05556, NACE International, Houston, Texas, USA.

Nesic, S., Li, H., Sormaz, D. and Huang, J. (2009). *An Open Source Mechanistic Model for Prediction of Mild Steel Corrosion*, CORROSION 2009, Paper 09572, NACE International, Atlanta, Georgia, USA.

Nesic, S., Postlethwaite, J. and Olsen, S. (1996). *An Electrochemical Model for Prediction of Corrosion of Mild Steel in Aqueous Carbon Dioxide Solutions*, Corrosion: April 1996, Volume 52, No. 04, pp. 280-294.

Nesic, S., Wang, S., Fang, H., Sun, W. and Lee, J. K-L. (2008). *A New Updated Model of CO₂/H₂S Corrosion in Multiphase flow*, CORROSION 2008, Paper 08535, NACE International, New Orleans, Louisiana, USA.

Newman, R. (2010). *Pitting Corrosion of Metals*, The Electrochemical Society Interface, Volume 19, No. 1, pp. 33-38.

Nordsveen, M., Nesic, S., Nyborg, R. and Stangeland, A. (2003). *A Mechanistic Model for Carbon Dioxide Corrosion of Mild Steel in the Presence of Protective Iron Carbonate Films- Part 1: Theory and Verification*, CORROSION, NACE International, Volume 59, No. 5, pp. 443-456.

NORSOK STANDARD M-506 Rev. 2: CO₂ Corrosion Rate Calculation Model, Standards Norway, Norway (June 2005).

Nyborg, R. (2002). *Overview of CO₂ Corrosion Models for Wells and Pipelines*, Corrosion 2002, Paper no. 02233, NACE International, Houston, Texas, USA.

Nyborg, R. (2005). *Controlling Internal Corrosion in Oil and Gas Pipelines*, Business Briefing: Exploration and Production, Oil and Gas Review, Issue 2, pp. 70-74.

Nyborg, R. (2010). *CO₂ Corrosion Models for Oil and Gas Production Systems*, Corrosion 2010, Paper no. 10371, NACE International, San Antonio, Texas, USA.

Ohio University (2008), Department of Chemical Engineering, Institute for Corrosion and Multiphase Technology, Ohio University's Research and Enterprise Park, 342 West State Street, Athens, Ohio 45701, USA. Accessed from:

<http://www.corrosioncenter.ohiou.edu/software/freecorp/>

Ohio University (2015), Department of Chemical Engineering, Institute for Corrosion and Multiphase Technology, Ohio University's Research and Enterprise Park, 342 West State Street, Athens, Ohio 45701, USA. Accessed from:

https://www.ohio.edu/mechanical/thermo/Intro/Chapt.1_6/Chapter2a.html

Olsen, S, Halvorsen, A. M., Lunde, P. G., Nyborg, R. (2005). *CO₂ Corrosion Prediction Model – Basic Principles*, CORROSION 2005, Paper 05551, NACE International, Houston, Texas, USA.

Perez, T. E. (2013). *Corrosion in the Oil and Gas Industry: An Increasing Challenge for Materials*, The Journal of Minerals, Metals and Materials Society (TMS), Vol. 65, No. 8, pp. 1033-1041.

Pots, B. F. M., John, R. C., Rippon, I. J., Thomas, M. J. J. S., Kapusta, S. D., Girgis, M. M., Whitham, T. (2002). *Improvements on De Waard-Milliams Correlation Prediction and Applications to Corrosion Management*, Corrosion 2002, Paper no. 02235, NACE International, Houston, Texas, USA.

Rao, A., Rubin, E. (2002). *A Technical, Economic and Environmental Assessment of Amine-based CO₂ Capture Technology for Power Plant Greenhouse Gas Control*, Environmental Science & Technology, Volume 36, pp. 4467-4475.

Riahi, K., Grubler, A., Nakicenovic, N. (2007). *Scenarios of Long-term Socio-economic and Environmental Development under Climate Stabilization*, Technological Forecasting and Social Change, Volume 74, Issue 7, pp. 887-935.

Rice, J. A. (2007). *Mathematical Statistics and Data Analysis*, Third Edition, Thomson Brooks/Cole Publishing, Thomson Higher Education, 10 Davis Drive, Belmont, California 94002-3098, USA, Chapter 5, Pages 181-183.

Sak, H., Senior, A. W., Beaufays, F. (2014). *Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling*, INTERSPEECH-2014, Singapore, pp. 338-342.

Satter, A., Iqbal, G. M., Buchwalter, J. L. (2007). *Practical Enhanced Reservoir Engineering*, PennWell Corporation, 1421 South Sheridan Road, Tulsa, Oklahoma 74112-6600, USA, Chapter 17, Page 556.

Schmitt, G., Feinen, S. (1983). *Effect of Anions and Cations on the Pit Initiation in CO₂ Corrosion of Iron and Steel*, Corrosion 83, NACE, Paper no. 43, Houston, Texas, USA.

Schmitt, G., Engels, D. (1988). *SEM/EDX Analysis of Corrosion Products for Investigations on Metallurgy and Solution Effects in CO₂ Corrosion*, Corrosion 98, NACE, Paper no. 149, Houston, Texas, USA.

Schmitt, G., Feinen, S. (2000). *Effect of Anions and Cations on the Pit Initiation in CO₂ Corrosion of Iron and Steel*, Corrosion 2000, Paper no. 1 (Houston, TX: NACE, 2000).

Schmitt, G., Schütze, M., Hays, G. F., Burns, W., Han, E-H, Poubaix, A. and Jacobson, G. (2009). *Global Needs for Knowledge Dissemination, Research and Development in Materials Deterioration and Corrosion Control*, The World Corrosion Organization (WCO).

Sharma, B. and Venugopalan, K. (2014). *Comparison of Neural Network Training Functions for Haematoma Classification in Brain CT Images*, Journal of Computer Engineering (JCE), Volume 16, Issue 1, pp.31-35.

Smith, S. N., Joosten, M. W. (2006). *Corrosion of Carbon Steel by H₂S in CO₂ Containing Oilfield Environments*, Corrosion 2006, Paper No. 06115, NACE International, San Diego, California, USA.

Sobol, I. M. (1994). *A Primer for the Monte Carlo Method*, CRC Press Inc., 2000 Corporate Boulevard, N. W., Boca Raton, Florida 33431, USA, Pages 1-16.

Song, F. M., Kirk, D. W., Cormack, D. E. (2005). *A Comprehensive Model for predicting CO₂ Corrosion in Oil and Gas Systems*, Corrosion 2005, Paper No. 05180, NACE International, Houston, Texas, USA.

Spycher, N., Pruess, K., Ennis-King, J. (2003). *CO₂-H₂O mixtures in the geological sequestration of CO₂. I. Assessment and calculation of mutual solubilities from 12 to 100°C and up to 600 bar*, Geochimica et Cosmochimica Acta 2003, Vol. 67, No. 16, pp. 3015-3031.

Staelens, N., Reyniers, M.-F., Marin, G. B. (2002). *Langmuir-Hinshelwood-Hougen-Watson rate equation for the transalkylation of methylamines*, Chemical Engineering Journal, Vol. 90, Issues 1-2, pp. 185-193.

Statistics and Machine Learning Toolbox User's Guide R2015b, (2015). The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-209, Pages 4-10, 4-11, 4-15, 4-16, 5-3, 5-8, 5-9, 5-10, 5-13 and 5-17.

Stewart, W. J. (2009). *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*, Princeton University Press, 6 Oxford Street, Woodstock, Oxfordshire OX20 1TW, Chapter 5, Page 105.

Sun, W. and Nescic, S. (2007). *A Mechanistic Model of H₂S Corrosion of Mild Steel*, Corrosion 2007, Paper No. 07655, NACE International, Nashville, Tennessee, USA.

Sun, W., Nescic, S. and Papavinasam, S. (2006). *Kinetics of iron sulfide and mixed iron sulfide/carbonate scale precipitation in CO₂/H₂S corrosion*, Corrosion 2006, Paper No. 06644, NACE International, Houston, Texas, USA.

Sun, Y. and Nescic, S. (2004). *A Parametric Study And Modeling on Localized CO₂ Corrosion in Horizontal Wet Gas Flow*, Corrosion 2004, Paper 04380, NACE International, New Orleans, Louisiana, USA.

Suryanarayana, T. M. V., Mistry, P., B. (2016). *Principal Component Regression for Crop Yield Estimation*, 1st Edition, SpringerBriefs in Applied Sciences and Technology, Springer Nature Publishing, Springer Science+Business Media Singapore Plc Ltd., Springer, Chapter 2, pages 17-25.

Tan, J., Chan, K. S. (2011). *Understanding Advanced Physical Inorganic Chemistry: The Learner's Approach*, World Scientific Publishing Co. Ltd., Chapters 5 and 6, Pages 159-160, 201, 211, 221-22.

Tanupabrungsun, T., Brown, B., Nešić, S. (2013). *Effect of CO₂ Corrosion of Mild Steel at Elevated Temperatures*, Corrosion 2013, Paper No. 2348, NACE International, Orlando, Florida, USA.

Thomson, W. (1848). *On an Absolute Thermometric Scale Founded on Carnot's Theory of the Motive Power of Heat*, and Calculated from Regnault's Observations, Proceedings of the Cambridge Philosophical Society, Volume 1, pp. 66-71.

Ukil, A. (2010). *Intelligent Systems and Signal Processing in Power Engineering*, 2nd Edition, Springer-Verlag Berlin Heidelberg, Chapters 2 and 3, Pages 5, 6, 13, 14, 93 and 95.

University of California Santa Barbara (UCSB), CA 93106, Materials Research Laboratory, California, USA. Accessed from: <http://scienceline.ucsb.edu/getkey.php?key=4120>

van Vuuren, D. P., den Elzen, M. G. J., Lucas, P. L., Eickhout, B., Strengers, B. J., van Ruijven, B., Wonink, S., van Houdt, R. (2007). *Stabilizing Greenhouse Gas Concentrations at Low Levels: An Assessment of Reduction Strategies and Costs*, Climatic Change, Volume 81, Issue 2, pp. 119-159.

Veelenturf, L. P. J. (1995). *Analysis and Applications of Artificial Neural Networks*, Prentice Hall International UK Limited, Campus 400, Maylands Avenue, Hemel Hempstead, Hertfordshire, HP2 7EZ, Simon & Schuster International Group, Chapter 2, pp. 6, 8-10.

Videm, K., Dugstad, A. (1989). *Materials Performance*, Volume 28, No. 3, Page 63.

Vogl, T. P., J.K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon. (1988). *Accelerating the Convergence of the Backpropagation Method*, Biological Cybernetics, Volume 59, pp. 257-263.

Wise, M., Calvin, K., Thomson, A., Clarke, L., Bond-Lamberty, B., Sands, R., Smith, S. J., Janetos, A., Edmonds, J. (2009). *Implications of Limiting CO₂ Concentrations for Land Use and Energy*, SCIENCE, Volume 324, pp. 1183-1186.

Woollam, R. C., and Hernandez, S.E. (2006). *Assessment and Comparison of CO₂ Corrosion Prediction Models*, SPE International Oilfield Corrosion Symposium, May 2006, Society of Petroleum Engineers, Aberdeen, UK.

World Energy Council, WEC (2007). *Carbon Capture and Storage: A WEC “Interim Balance”*, World Energy Council, pp.17.

Yap, K.-M., Srinivasian, S. Honeywell International Inc. (2010). *Key Factors In Development of a CO₂/H₂S Corrosion Prediction Model for Multiphase Oil/Gas Production Systems*, Corrosion 2010, NACE International, San Antonio, Texas, USA.

Yin, Z. F., Feng, Y. R., Zhao, W. Z., Bai, Z. Q., and Lin, G. F. (2009). *Effect of Temperature on CO₂ Corrosion of Carbon Steel*, Journal of Surface and Interface Analysis, Volume 41, Issue 6, pp. 517-523.

Yoon-Seok, C., Netic, S., (2009). *Corrosion Behaviour of Carbon Steel in Supercritical CO₂-Water Environments*, Corrosion 2009, NACE International, Atlanta, Georgia, USA.

Zadeh L.A. (1965). *Fuzzy sets*, Information and Control, Vol. 8, Issue 3, pp. 338–353.

Zadeh, L., A. (1973). *Outline of a new approach to the analysis of complex systems and decision processes*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 3, No. 1, pp. 28-44.

Zadeh, L. A. (1975). *The Concept of a Linguistic Variable and its Application to Approximate Reasoning – I*, Information Sciences, Volume 8, Issue 3, pp. 199-249.

Zevenhoven, R. and Kilpinen, P. (2004). Control of Pollutants in Flue Gases and Fuel Gases, 3rd Edition, Sponsors: The Nordic Energy Research Programme, Solid Fuel Committee, Ås, Norway and Helsinki University, Espoo, Finland. Chapter 2: Flue gases and fuel gases, Pages: 2-2, 2-3.

Zhang, Y., Gao, K., Schmitt, G. and Hausler, R. H. (2012a). *Modeling Steel Corrosion under Supercritical CO₂ Conditions*, Journal of Materials and Corrosion, Volume 64, Issue 6, pp.478-485. DOI: 10.1002/maco.201106382

Zhang, Y., Pang, X., Qu, S., Li, X., Gao, K. (2012b). *Discussion of the CO₂ corrosion mechanism between low partial pressure and supercritical condition*, Corrosion Science: 2012, Vol. 59, pp. 186-197.