Automated Modelling and Assessment of the Impact of Damage Applied on Suspension Bridges' Main Cables Using Natural Frequency Analysis

by

Étienne Noël

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Master of Applied Science degree in Civil Engineering

Department of Civil Engineering
Faculty of Engineering
University of Ottawa

© Étienne Noël, Ottawa, Canada, 2025

Abstract

This thesis describes the analysis of the effect of damage application to the main cables of the Lower Liard River Bridge (LLRB). A program in Python is created to generate custom suspension bridge finite element models in Abaqus CAE based on generalized parameters. While this study focuses strictly on the LLRB, the Python program is flexible to adapt to most suspension bridge designs. From there, over 2,000 test cases are defined with varying damage distributions along the main cables of the model. Damage is interpreted as corrosion based on inspection reports, and distributions are created to observe the effect of these variations on a few critical properties of the bridge. Key outputs studied are shifts in the natural frequencies of the modes of vibration of the deck and the vertical deck displacement under gravity. The Python program was enhanced to automate test execution and analysis. As part of this process, a rudimentary mode identification algorithm calibrated to the LLRB to recognize vibration modes in the deck from the outputs of Abaqus is created. Among these tested cases are a few simple wind response analyses of the bridge under dynamic wind loading. For this, an algorithm is defined to generate high-frequency wind velocity signals that are required to obtain a time history response of the bridge. The results obtained in this research confirm that, as demonstrated in previous studies, there are observable changes in the natural frequencies of the bridge when damage is applied. This research reveals how damage location and intensity affect bridge behaviour, offering insights for future developments. Key findings include the identification of vertical vibration modes of the deck as the clearest indicator of the applied damage parameters. In contrast, only a few torsional modes reveal any correlation. Specific behaviours are observed when damage is implemented at particular locations along the length of the main cables, such as near the anchor, towers, and midspan. Notably, a strong correlation is established between the shifts in natural frequencies of individual cases with single damage locations and the complementary case combining two damage locations in the same case. Findings suggest the potential to parameterize a bridge's behaviour based on single damage case results to estimate the damage applied from the measured dynamic properties. Further work should explore more damage patterns and integrate machine learning to advance this method.

Acknowledgements

I would like to express my appreciation to Dr. Elena Dragomirescu, who served as my thesis supervisor during my M.A.Sc. program. Throughout the course of my research, Dr. Dragomirescu has always encouraged me to push forward and maintained a positive outlook on the project, which helped me persevere through the challenges I encountered.

Je tiens à remercier de tout mon coeur la personne qui a été à mes côtés chaque jour, m'offrant un soutien inestimable : ma conjointe et meilleure amie, Karlie. Je ne pourrai jamais suffisamment te remercier pour tes encouragements, ta patience et ta présence à chaque seconde de ce long parcours. Sans toi, je n'en serais pas arrivé là.

Je me dois d'exprimer toute ma reconnaissance à mes parents, qui ont toujours été une source d'inspiration tant dans ma vie professionnelle que personnelle. Votre support inconditionnel, vos conseils précieux à chaque tournant, ainsi que vos paroles rassurantes ont joué un rôle crucial dans l'accomplissement de ce projet.

À tous les membres de ma famille, de près ou de loin, vous avez été des soutiens indispensables. Chacun à votre manière, vous m'avez offert des moments de réconfort, de sagesse et d'amour, me permettant de rester motivé à travers chaque épreuve.

À mes amis, merci d'avoir enduré l'écoute de toutes mes péripéties incessantes. Votre présence rendait toujours les moments difficiles plus agréables et offrait une distraction bienvenue à mes préoccupations.

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Nous remercions le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) de son soutien.

Natural Sciences and Engineering Research Council of Canada
Conseil de recherches en sciences naturelles et en génie du Canada
Canada

Je veux aussi remercier le support du Fonds de recherche du Québec – Nature et technologies (FRQNT). [Numéro de dossier : 320169]

Fonds de recherche Nature et technologies
Québec

Contents

1 Introduction **1**

1.1 Background . . . 1
1.2 Research Motivation . . . 4
1.3 Research Objectives . . . 5
1.4 Research Methodology . . . 5
1.5 Conceptual Vision of the Project Framework . . . 6
1.6 Thesis Layout . . . 7

2 Literature Review **10**

2.1 Suspension Bridge General Theory . . . 10
2.1.1 Introduction to Suspension Bridge Theory . . . 10
2.1.2 Cable System and Load Distribution . . . 11
2.1.3 Dynamic Behaviour of Suspension Bridges . . . 12
2.1.4 Cable Slipping at Tower Saddles . . . 15
2.2 Finite Element Modelling of Suspension Bridges . . . 15
2.2.1 Basic Theory of Finite Element Method . . . 15
2.2.2 Introduction to Structural Dynamics in FEM . . . 21
2.2.3 FEM-Based Fishbone Models for Suspension Bridges . . . 25
2.3 Corrosion Process in Steel Wires . . . 27
2.3.1 Mechanisms of Corrosion in Steel Wires of Suspension Bridges . . . 27
2.3.2 Impact of Corrosion on Structural Properties of Steel Cables . . . 30
2.3.3 Higher Risk Locations For Main Cable Corrosion in Suspension Bridges . . . 34
2.4 Health Monitoring in Suspension Bridges . . . 36
2.4.1 Introduction to Structural Health Monitoring . . . 36
2.4.2 SHM Techniques for Suspension Bridges . . . 37
2.4.3 Natural Frequency-Based Methods . . . 38

2.5 Wind Loading and Signal Generation . . . 41
2.5.1 Introduction to Wind Loading on Suspension Bridges . . . 41
2.5.2 Wind Velocity Signal Generation . . . 43
2.6 Genetic Algorithms . . . 44

3 LLRB Finite Element Modelling 46
3.1 Chapter Overview . . . 46
3.1.1 Software Selection . . . 46
3.1.2 Structural Drawings Disclaimer . . . 46
3.1.3 Design Particularities of the LLRB . . . 47
3.1.4 LLRB Bridge Model Terminology . . . 48
3.2 Bridge Parts Modelling . . . 50
3.2.1 Towers Parts Modelling . . . 51
3.2.2 Deck Parts Modelling . . . 63
3.2.3 Hangers Parts Modelling . . . 74
3.2.4 Main Cables Parts Modelling . . . 77
3.3 Model Assembly Definition . . . 86
3.4 Model Steps Creation . . . 87
3.5 Model Constraints Development . . . 90
3.6 Model Boundary Conditions Definition . . . 97
3.7 Loading Generation and Definition . . . 98
3.8 Predefined Fields Modelling . . . 104
3.9 Summary . . . 106

4 Scripting and Bridge Model Automation 107
4.1 Chapter Overview . . . 107
4.2 Automatic Model Creation . . . 108
4.2.1 Inputs Description and Processing . . . 109
4.2.2 Abaqus Launch . . . 109
4.2.3 Model Definition . . . 110
4.2.3.1 Program Files Organization . . . 110
4.2.3.2 Inputs Deserialization . . . 112
4.2.3.3 Blank Model Definition . . . 113
4.2.3.4 Materials Definition . . . 113

4.2.3.5 Steps Definition . . . 114
4.2.3.6 Blank Assembly Definition . . . 114
4.2.3.7 Geometry Calculations and Parts Setup . . . 115
4.2.3.8 Parts and Instances Definition . . . 115
4.2.3.9 Constraints Creation . . . 118
4.2.3.10 Boundary Conditions Creation . . . 118
4.2.3.11 Loading Definition and Application . . . 119
4.2.3.12 Predefined Fields Creation . . . 120
4.2.4 Abaqus Job Creation and Submission . . . 121
4.3 Model Refinement . . . 121
4.3.1 Deck Elevations Iterations . . . 121
4.3.2 Cable Pretension Iterations . . . 127
4.4 Test Cases Definition . . . 130
4.4.1 FREQ11 Test Series . . . 130
4.4.2 FREQ34 Test Series . . . 134
4.4.3 WIND11 Test Series . . . 135
4.4.4 Tests Summary and Discussion . . . 135
4.5 Output Automatic Analysis . . . 136
4.5.1 Output JSON Object . . . 136
4.5.2 Mode Identification Algorithm . . . 137

5 Results and Discussion 141
5.1 Natural Frequencies Variations Analysis . . . 141
5.1.1 All_ WiresNumber . . . 145
5.1.2 Quad WiresNumber . . . 152
5.1.3 Sing WiresNumber . . . 155
5.1.4 Doub WiresNumber . . . 165
5.1.5 Discussion of Natural Frequencies Results . . . 176
5.2 Deck Vertical Displacement Analysis . . . 179
5.3 Wind Scenarios Output Analysis . . . 183

6 Conclusion 187
6.1 Main Research Findings and Contributions . . . 188
6.2 Research Limitations . . . 190

6.3 Future Research 191

Bibliography **192**

A Temperature Calculation Functions **202**

B Input Specification for the Bridge Modelling Program **203**

B.1 Steps JSON 203
B.2 Materials JSON 207
B.3 General Parameters Sheet 208
B.4 Deck Properties Sheet 209
B.5 Tower Properties Sheet 210
B.6 Cable Properties Sheet 212
B.7 Hanger Coordinates Sheet 212
B.8 Deck Spans Sheet 213
B.9 Main Cable Restraints Sheet 213
B.10 Hanger Beams Sheet 214
B.11 Cable Pretensions Sheet 214
B.12 Cable Damages Sheet 215
B.13 Wind Loading Sheet 216
B.14 Inputs Parsing and Processing 216

C Cable Pretension Values **218**

D Basic Python Commands for Abaqus **219**

E Polynomial Root Finding Algorithm **227**

F Genetic Algorithm **228**

G Wind Velocity Signal Generation **230**

H Equivalent Cable Damage Calculations **235**

I Additional Natural Frequencies Results **237**

List of Figures

1.1 Longest Unsupported Bridge Span Through the Years . . . 2
1.2 Suspension Bridges of Canada - Years of Activity . . . 3

2.1 Typical Suspension Bridge Sketch [1] . . . 11
2.2 Cable Connection Forces . . . 11
2.3 Typical Deck Modes of Vibration [2] . . . 14
2.4 First Natural Frequencies of Deck's Vibration Modes Against Main Span Lengths [3] 15
2.5 Finite Element Shapes [4] . . . 17
2.6 Degrees of Freedom in a Truss and a Frame Element . . . 18
2.7 Fishbone Model Idealization [5] . . . 26
2.8 Iron Corrosion Process [6] . . . 27
2.9 Corrosion Pit Mechanical Properties Changes For Depth-To-Width Ratio [7] . . . 31
2.10 Cable Corrosion Damage Pattern Types [8] . . . 32
2.11 Corrosion Pit Mechanical Properties Changes For Section Loss Per Damage Pattern Type [8] . . . 33
2.12 Lower Liard River Bridge Cable Damage Picture From Visual Inspection [9] . . . 36
2.13 Wind Loads on a Body: Drag, Lift, and Moment [5] . . . 42
2.14 Simple Genetic Algorithm [10] . . . 45

3.1 Aerial Picture of the Lower Liard River Bridge [11] . . . 47
3.2 Main Cable Bundle Wire Notation . . . 48
3.3 Nomenclature for the Complete Bridge Model . . . 49
3.4 Nomenclature of the Main Components of the Tower . . . 52
3.5 General Tower Dimensions . . . 53
3.6 Cross Girder Sets Naming . . . 55
3.7 Vertical Tower Sets Naming . . . 56
3.8 Tower Mesh . . . 57

3.9 Horizontal Brace From the 2011 Visual Inspection [9] . . . 59
3.10 Horizontal Brace CSiBridge Section Designer [12] . . . 60
3.11 Cross Girder CSiBridge Section Designer [12] . . . 61
3.12 Vertical Tower CSiBridge Section Designer [12] . . . 62
3.13 Deck Fishbone Model of the LLRB . . . 64
3.14 Assigned Terminology for the Girder Sets . . . 65
3.15 Assigned Terminology for the Girder at Tower Sets . . . 66
3.16 Deck Lateral View of the LLRB . . . 67
3.17 LLRB Deck Segment Solid Model . . . 69
3.18 Deck Solid Model Displacements . . . 71
3.19 Main Cable Bundle Variations . . . 79
3.20 Cable Connector From the 2011 Visual Inspection [9] . . . 81
3.21 Cable Connector Part . . . 83
3.22 LLRB Cable Restraint [9] . . . 85
3.23 Cable Restraint Part . . . 85
3.24 LLRB Model Assembly . . . 87
3.25 Cable Connector Behaviour Display . . . 92
3.26 Anchor Bearings Picture From the 2011 Visual Inspection [9] . . . 95
3.27 Anchor Bearings Behaviour Display . . . 96
3.28 Main Cable Wires Anchorage From the 2011 Visual Inspection [9] . . . 98
3.29 Complete Wind Speed Signal . . . 101
3.30 Wind Signal Frequencies Distribution . . . 101
3.31 Short Wind Velocity Signal . . . 102

4.1 General Program Process . . . 109
4.2 Program Files Organization . . . 112
4.3 Hanger Tension Comparison - Root Finding Algorithm . . . 124
4.4 Fitness Score Evolution per Generation . . . 126
4.5 Hanger Tension Comparison - Genetic Algorithm . . . 127
4.6 Hanger Tension Comparison - Pretension Algorithm . . . 129
4.7 Main Cable Tensions - Pretension Algorithm . . . 129
4.8 Damage Cases Notation . . . 132
4.9 Damage Symmetry Patterns . . . 133
4.10 Deck Displacement Factors for Mode Type Identification . . . 139

4.11 Deck Symmetry Factors for Mode Symmetry Identification 140

5.1 FREQ11 Baseline Model First Symmetrical Modes of Vibration 142
5.2 FREQ11 Baseline Model First Asymmetrical Modes of Vibration 143
5.3 FREQ11 Baseline First 50 Natural Frequencies with Identified Modes 144
5.4 FREQ34 Baseline First 50 Natural Frequencies with Identified Modes 144
5.5 FREQ11 All_ Vertical Modes of Vibration 146
5.6 FREQ34 All_ Vertical Modes of Vibration 147
5.7 FREQ11 All_ Lateral Modes of Vibration 148
5.8 FREQ34 All_ Lateral Modes of Vibration 149
5.9 FREQ11 All_ Torsional Modes of Vibration 150
5.10 FREQ34 All_ Torsional Modes of Vibration 151
5.11 FREQ11 Quad Vertical Modes of Vibration 153
5.12 FREQ34 Quad Vertical Modes of Vibration 154
5.13 FREQ34 Quad Torsional Modes of Vibration 155
5.14 FREQ11 Sing Symmetrical Vertical Modes of Vibration 157
5.15 FREQ11 Sing Asymmetrical Vertical Modes of Vibration 159
5.16 FREQ34 Sing Symmetrical Lateral Modes of Vibration 160
5.17 FREQ11 Sing Symmetrical Torsional Modes of Vibration 162
5.18 FREQ11 Sing Asymmetrical Torsional Modes of Vibration 163
5.19 FREQ34 Sing Symmetrical Torsional Modes of Vibration 164
5.20 FREQ34 Sing Asymmetrical Torsional Modes of Vibration 165
5.21 FREQ11 Doub Symmetrical Vertical Modes of Vibration 166
5.22 FREQ11 Symmetrical Vertical Modes of Vibration - Combined Analysis 168
5.23 FREQ11 Doub Asymmetrical Vertical Modes of Vibration 169
5.24 FREQ11 Asymmetrical Vertical Modes of Vibration - Combined Analysis 170
5.25 FREQ11 Symmetrical Lateral Modes of Vibration - Combined Analysis 171
5.26 FREQ11 Asymmetrical Lateral Modes of Vibration - Combined Analysis 172
5.27 FREQ11 Doub Symmetrical Torsional Modes of Vibration 173
5.28 FREQ11 Symmetrical Torsional Modes of Vibration - Combined Analysis 174
5.29 FREQ11 Doub Asymmetrical Torsional Modes of Vibration 175
5.30 FREQ11 Asymmetrical Torsional Modes of Vibration - Combined Analysis 176
5.31 FREQ11 Baseline Deck Elevation and Displacement 179
5.32 Deck Vertical Displacement for Location Index Independent Cases with 0.30 Damage 181

5.33 Deck Vertical Displacement for Location Index Dependent Cases with 0.30 Damage - XSym . . . 182
5.34 Deck Vertical Displacement for Location Index Dependent Cases with 0.30 Damage - DSym . . . 183
5.35 WIND11 Deck Lateral Translation for Doub DSym 0.30 Damage Cases . . . 185
5.36 WIND11 Deck Rotation for Doub DSym 0.30 Damage Cases . . . 186

G.1 Kaimal Spectrum . . . 230
G.2 White Noise in the Time-Domain . . . 231
G.3 White Noise in the Frequency-Domain . . . 231
G.4 Wind Spectrum . . . 232
G.5 Initial Wind Signal . . . 232
G.6 Mean Wind Speed Variation . . . 233
G.7 Wind Signal With Mean Speed Variation . . . 233
G.8 Wind Signal With Mean Speed Variation and High-Frequency Noise . . . 234

I.1 FREQ11 Quad Lateral Modes of Vibration . . . 237
I.2 FREQ34 Quad Lateral Modes of Vibration . . . 238
I.3 FREQ11 Quad Torsional Modes of Vibration . . . 238
I.4 FREQ34 Sing Symmetrical Vertical Modes of Vibration . . . 239
I.5 FREQ34 Sing Asymmetrical Vertical Modes of Vibration . . . 240
I.6 FREQ11 Sing Symmetrical Lateral Modes of Vibration . . . 241
I.7 FREQ11 Sing Asymmetrical Lateral Modes of Vibration . . . 242
I.8 FREQ34 Sing Asymmetrical Lateral Modes of Vibration . . . 243
I.9 FREQ34 Doub Symmetrical Vertical Modes of Vibration . . . 244
I.10 FREQ34 Doub Asymmetrical Vertical Modes of Vibration . . . 245
I.11 FREQ11 Doub Symmetrical Lateral Modes of Vibration . . . 246
I.12 FREQ34 Doub Symmetrical Lateral Modes of Vibration . . . 247
I.13 FREQ11 Doub Asymmetrical Lateral Modes of Vibration . . . 248
I.14 FREQ34 Doub Asymmetrical Lateral Modes of Vibration . . . 249
I.15 FREQ34 Doub Symmetrical Torsional Modes of Vibration . . . 250
I.16 FREQ34 Doub Asymmetrical Torsional Modes of Vibration . . . 251

List of Tables

2.1 Lower Liard River Bridge Main Cable Corrosion and Damage Locations [9] 35

3.1 Assigned Terminology for the LLRB Tower Parts 53
3.2 Individual Steel Beam Properties [13] 58
3.3 Tower Profiles Final Properties 62
3.4 Assigned Terminology for the LLRB Deck Parts 64
3.5 Start and End Coordinates of Deck Elements 67
3.6 Deck Profiles Final Properties 73
3.7 Assigned Terminology for the LLRB Hanger Parts 74
3.8 Coordinates of Hanger Elements and Ties 75
3.9 Hanger Beam Profile Properties 77
3.10 Assigned Terminology for the LLRB Main Cable Parts 77
3.11 Cable Connection Profile Properties 84
3.12 Constraints Names and Count of Occurrences in the Model 91
3.13 Boundary Conditions Names and Count of Occurrences in the Model 97
3.14 Load Names 104
3.15 Estimated Wind Loads Parameters and Linear Components 104

4.1 FREQ11 Test Cases Enumeration 134
4.2 FREQ34 Test Cases Enumeration 135
4.3 WIND11 Test Cases Enumeration 135

B.1 Parameters in Steps JSON objects - Static, General 204
B.2 Parameters in Steps JSON objects - Frequency 205
B.3 Parameters in Materials JSON objects 207
B.4 General Parameters Input Sheet 209
B.5 Deck Properties Input Sheet 210

B.6 Tower Properties Input Sheet - Main Profiles, Mesh and Constraints 210
B.7 Tower Properties Input Sheet - Coordinates and Braces 211
B.8 Cable Properties Input Sheet 212
B.9 Hanger Coordinates Input Sheet 213
B.10 Deck Spans Input Sheet 213
B.11 Main Cable Restraints Input Sheet 214
B.12 Hanger Beams Input Sheet 214
B.13 Cable Pretensions Input Sheet 215
B.14 Cable Damage Input Sheet 216
B.15 Wind Loading Input Sheet 216

C.1 Pretension Values in the Hangers and the Main Cables 218

H.1 Cable Damage Equivalency 236

List of Algorithms

3.1 Wind Signal Generator . . . 100
4.1 Abaqus Launch Command . . . 110
4.2 Blank Model Creation Commands . . . 113
4.3 Assembly Setup Commands . . . 114
4.4 Profile Creation Command . . . 115
4.5 Truss Section Creation Commands . . . 116
4.6 Beam Section Creation Commands . . . 116
4.7 Wire Part Creation Commands . . . 117
4.8 Instance Creation Command . . . 118
4.9 Instance Translation Command . . . 118
4.10 Gravity Loading Creation Command . . . 119
4.11 Concentrated Load Creation Command . . . 119
4.12 Line Load Creation Command . . . 120
4.13 Moment Load Creation Command . . . 120
4.14 Mode Identification Algorithm . . . 140
A.1 Calculate Temperature Difference Functions . . . 202
B.1 Steps JSON File . . . 206
B.2 Materials JSON File . . . 208
D.1 Elastic Material Creation Commands . . . 219
D.2 Static General Step Creation Commands . . . 220
D.3 Frequency Step Creation Commands . . . 221
D.4 Vertex Set Creation Commands . . . 221
D.5 Mesh By Factor Creation Commands . . . 222
D.6 Mesh By Value Creation Commands . . . 222
D.7 Section Assignment Command . . . 222
D.8 Beam Orientation Assignment Command . . . 223

D.9 Coupling Constraint Creation Command 223
D.10 Multipoint Constraint Creation Command 224
D.11 Pinned Boundary Condition Creation Command 224
D.12 Encastre Boundary Condition Creation Command 224
D.13 Displacement Boundary Condition Creation Command 225
D.14 Predefined Field Creation Command for Pretension 225
D.15 Job Creation and Submission Command 226
E.1 Polynomial Root Finding Function 227
F.1 Genetic Algorithm Next Individual Creation 228
F.2 Genetic Algorithm Functions 229

Chapter 1

Introduction

1.1 Background

From the beginning of the 19th century, the emergence of suspension bridges marked the start of a new era in infrastructure development. This new concept would set the stage for two centuries of revolutionary progress in transportation. With its ability to span much longer unsupported distances and requiring fewer construction materials than the arch bridge, its alternative at the time, the suspension bridge became an indispensable infrastructure for civil engineers. As bridges are often used to cross bodies of water, the longer spans provided unparalleled advantages, not to mention the practicality and adaptability of this new type of construction. Technological advancements for many bridge types were important as the field continued advancing, but the suspension bridge maintained its advantage in material efficiency, slender configuration, and span capability. Figure 1.1 illustrates the evolution with time of the longest unsupported span for different bridge types [14, 15, 16, 17, 18, 19, 20], with the suspension bridge being virtually undefeated for the last 200 years.

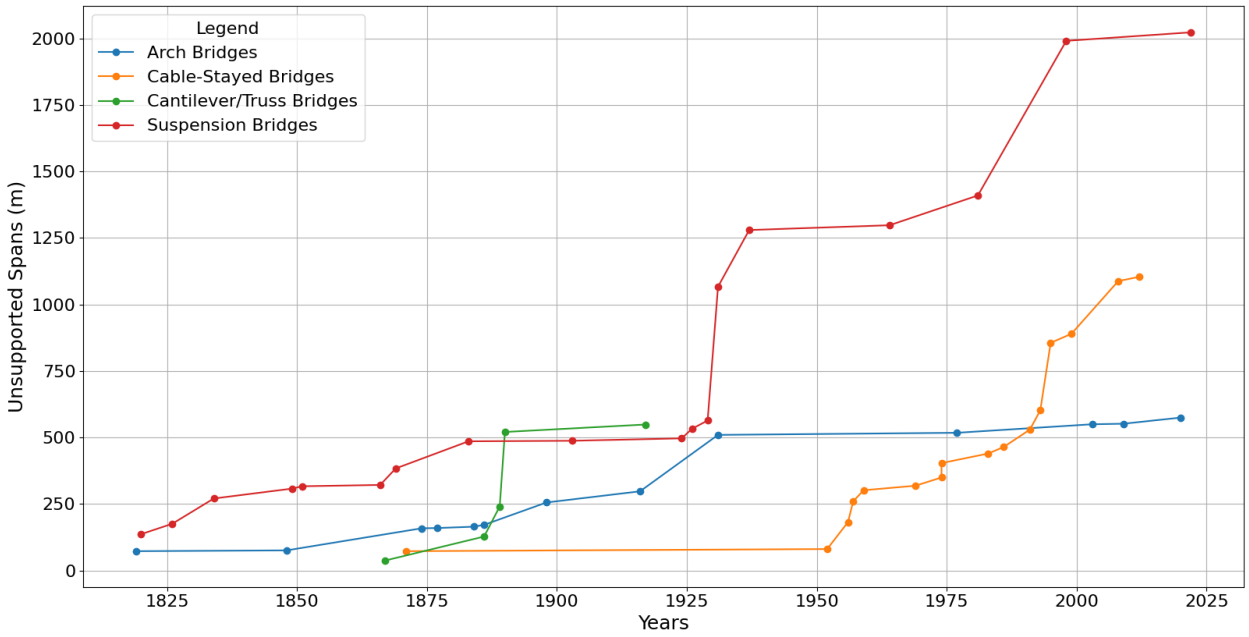


Figure 1.1: Longest Unsupported Bridge Span Through the Years

Today, projects such as the 1915 Çanakkale Bridge in Turkey, completed in 2022 and currently the world's longest suspension bridge with a main span of 2,023 m, are pushing civil engineering toward longer and more slender span geometries that were once considered unachievable. However, this bridge concept demands a significant requirement: specialized expertise. Indeed, these infrastructures are highly complex systems, requiring cables, towers, and the deck to work together from the design phase to construction. One may recall the infamous Tacoma Narrows Bridge collapse in 1940, which failed just four months after its opening due to a structural aerodynamics phenomenon that was unknown at the time. This serves as an example of the impact that a lack of expertise can have on such structures. Moreover, in addition to the complex design and high construction costs, suspension bridges require cost-intensive maintenance for the cables and deck, necessitating regular inspections and monitoring. These challenges are among the reasons why no suspension bridges have been constructed in Canada since 1970. However, as of 2025, there are still 17 active vehicular suspension bridges in Canada, most of which are nearing the end of their expected lifespans. This is illustrated in Figure 1.2, which summarizes the years of activity for all Canadian vehicular suspension bridges [21]. As these infrastructures age, the cost of maintaining and repairing them increases. However, these interventions are necessary to extend the lifespan of a bridge and prevent structural collapse. Therefore, we can expect that suspension bridges in Canada will require increasing repairs and replacements in the near future.

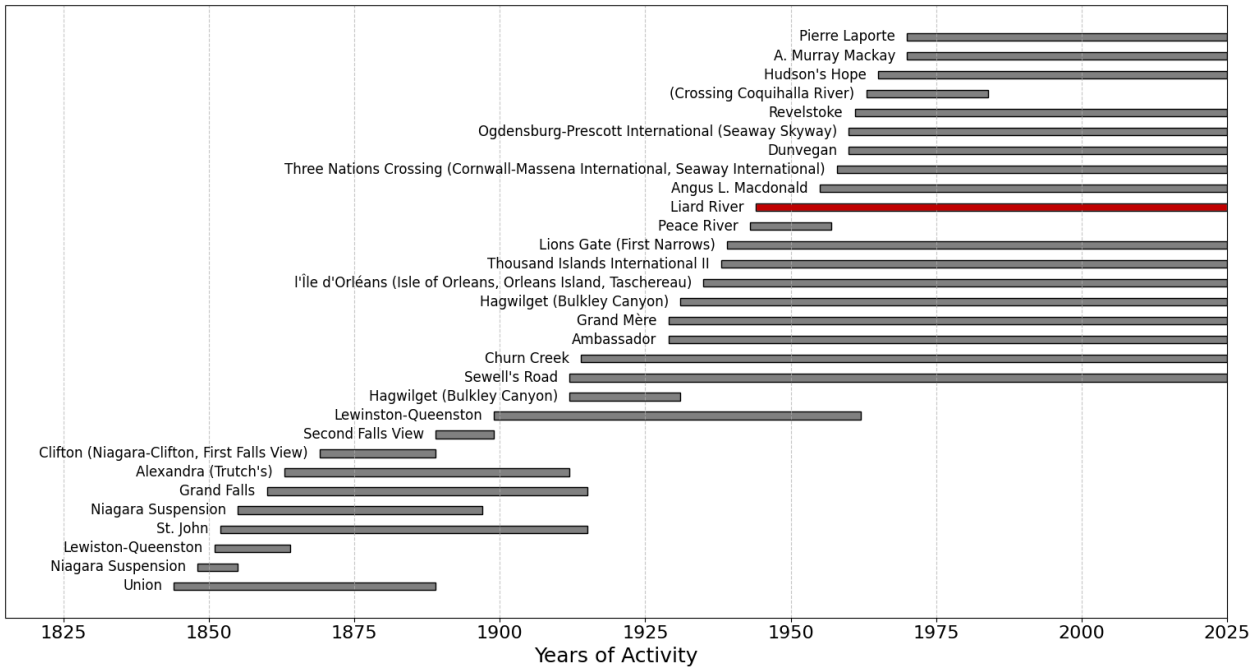


Figure 1.2: Suspension Bridges of Canada - Years of Activity

The Lower Liard River Bridge (LLRB) is a good example of a suspension bridge in Canada that is at the end of its lifespan. In 1943, this suspension bridge was constructed in British Columbia as part of the Alaska Highway. The objective of the construction of the highway was to connect Alaska to the rest of the United States in the event of an invasion from North America's west coast following the attack on Pearl Harbor during World War II. The LLRB was constructed using salvaged materials from the Tacoma Narrows Bridge after its failure due to steel shortages during the war [11]. This bridge has a span of more than 300 meters and has a standard two-tower design. Further technical details are provided in future sections, but it is important to note that this bridge is over 80 years old and displays signs of corrosion damage in the cables and the deck. The drawings were provided as a contribution to the development of this project. All provided documents, as well as the name of the owner, may not be shared in this publication due to a non-disclosure agreement.

One of the particularity of this bridge is that at the time of construction, the technology of constructing the main cable was not well developed. Moreover, it involved special compressing equipment which was not available in the remote areas such as the LLRB site. Therefore, the main cables were built as an assembly of 12 individual steel cables, bundled together to form a rectangle cross-section (3x4 pattern) and held in place with steel and wood restraints. This type of cable assemblage makes the strands more vulnerable to environmental factors, thus it is not used nowadays for suspension bridges. However, it is very challenging and structurally risky to

dismantle and replace the main cable of a suspension bridge. Therefore, the bridges which were built with this concept are preserved by inspections and repairs of the steel wires.

1.2 Research Motivation

Many suspension bridges in Canada are approaching the end of their lifespan, which causes an increase in their cost of maintenance. However, critical repairs are often performed too early or too late due to a lack of visibility of the state of the structure. Maintenance performed too early incurs high unnecessary costs, whereas maintenance carried out too late can cause even more serious issues or structural failure in extreme cases. This limited visibility is a considerable problem caused by the complexity of monitoring older structures, and existing methods for assessing the structural health of suspension bridges are lacking compared to the level of available technologies. In many cases, the primary monitoring method is through visual inspections, an inefficient method prone to human errors. The difficulty of accessing the bridges located in remote areas, such as the LLRB, can hinder even more the restoring process of the bridge. A real-time tracking alternative with damage identification capabilities could prevent excessive damage from going unnoticed. This is particularly accurate for the periods between visual inspections, which may be separated by up to two years. In the case of the LLRB, one of the ongoing issues is the corrosion in the main cable wires. Visual inspections are made every few years, but if corrosion in the cables was to get worse in between, it would go unnoticed. The LLRB is a good candidate for this research since it is a bridge in a remote area which would particularely from a real-time remote monitoring system. Furthermore, while shorter suspension bridges are more rarely the focal point of this type of research since they are less subject to aerodynamic effects, it is expected that the method and the conclusions obtained from this bridge are applicable to structures with longer spans. Moreover, the impact of cable corrosion damage on the overall structural performance is not readily available. This means that even if damage is observed, its effect may not necessarily be known, which may again lead to repairs being done too early or too late. Therefore, the impediments reported for the processes of structural health monitoring and visual inspections contributed to the motivation of the current research project, which considers alternative methodologies of investigating change in structural properties of the LLRB.

1.3 Research Objectives

The main goal of the current research is to develop an automated tool which will generate the LLRB finite element model in Abaqys via Python scripting and would further enable the generation of other suspension bridges of similar configurations. Through the same automated tool, the current research investigates the change in different properties of the LLRB, such as the natural frequencies of the deck's vibration modes, when corrosion damage is introduced at various locations along the main cable wires. Finally, this research aims at setting the basis for creating a cable damage database to be further used for training a machine learning algorithm to identify damage intensities and locations in components of a suspension bridge. To achieve these research goals, the following intermediate objectives are envisioned for the current project.

1. Develop a program that automatically generates numerical models of suspension bridges based on detailed input of the bridge-specific properties.

2. Create a numerical model of the LLRB with the developed program that considers the particularities of the infrastructure, such as separating the main cables into multiple elements.

3. Calibrate the LLRB finite element model by setting the deck geometry to its final configuration and adjusting pretension in the cables to maintain such position.

4. Determine if there are quantifiable correlations between the damage of the main cables and changes in the properties of the LLRB. This should mainly cover the shifts in the natural frequencies but could also include the bridge response under gravity and wind loading.

1.4 Research Methodology

Subsequently, the methodology adopted in this research contains numerical work and is divided in several phases. The first part of the process is to create a program that generates a numerical model that can be analyzed automatically. The automation part is critical for this project to be expandable to any future projects investigating suspension bridges. Moreover, the computation process for a single model iteration should be efficient to allow a significant number of cases to be tested. Based on these requirements, it is determined that the only viable option consists in developing the model of the infrastructure with finite element theory, simplifying the actual structure into a more manageable model. The model generation automation should be prioritized when selecting the software for this project, which is the reason for choosing Abaqus. While Abaqus

offers a complete graphical user interface, a key feature is that it can also perform modelling tasks based on received Python commands. Therefore, the first step of this research must be to write a Python program that sends a series of precise commands to Abaqus to model the required structure. This process must be performed to allow the user to run multiple scenarios in a queue without any interactions with the program after launching it. As such, this program must not only include the generation of the model but also the job submission in Abaqus, the sampling of the appropriate results, the preliminary analysis of these results, the saving of the model as well as all of its auxiliary files, and the closing of the model.

Regarding the parameters used in generating the model, a simple method to provide inputs should be available for users to specify all required parameters. For ease of use, this is to be performed in an Excel spreadsheet organized to make it trivial for users to describe the structure to model. After appropriate testing of the created program, the LLRB structure should be created. This step includes determining the proper structural parameters to use as inputs based on the structural drawings of the bridge. The model's results should be evaluated and modified accordingly to calibrate the model. Once the model is representative of the original structure, corrosion damage cases can be defined based on the location and intensity of the damage. These should encompass a diverse range of patterns to facilitate the identification of patterns in the outputs. A wind signal generator is created for the wind analysis, and the program applies the appropriate loading to the model. Furthermore, an analysis of the shifts in the natural frequencies can be made to establish dependence on the applied damage parameters. Moreover, the response to gravity and wind loading may also be studied in correlation to the applied damage.

1.5 Conceptual Vision of the Project Framework

Before covering the research conducted in this study, it is crucial to understand the long-term objective of pursuing the larger project. This section briefly explains the vision that guided the development of this research. To reach a developed real-time monitoring system for a suspension bridge within the context of this research, accelerometers must be installed on the deck at a few locations along the length of the bridge. These sensors must sample displacements of each location remotely in real time. The required sampling rate of these sensors would need to be sufficiently high to capture the vibration modes of the deck. For the LLRB, the results in this research seem to indicate that a sampling rate of 6-10 Hz would be ideal based on the natural frequencies obtained, but longer structures would not require rates as high. Through the real-time high-frequency deck displacement signals, the system developed would detect changes in the natural

frequencies over time through Fast Fourier Transform (FFT), which can be associated with damage in the bridge's main cables. Having constant monitoring of the bridge would make it possible to eliminate environmental effects from the results over time through a moving window analysis with long windows to detect dominant natural frequencies. The real-time system on the bridge's deck would allow for the identification of corrosion propagation over time. In contrast, a single analysis in time does not permit comparison with previous states of the structure, making it impossible to identify changes in the state of the main cables.

An important consideration at this stage is that the analysis of the deck's natural frequencies is considerably more feasible than the analysis of the main cables' natural frequencies. This is because monitoring each component of the main cable in every segment necessitates an immense number of sensors, significantly surpassing the quantity required for the deck.

Furthermore, while live loads are not considered in this bridge due to the low amount of traffic it experiences, it is necessary to determine how traffic interferes with the natural frequencies of the deck vibration modes to generalize the proposed solution.

Assuming that the monitoring system feeds natural frequency states to an algorithm, such as a genetic algorithm or any machine learning alternative, estimation of the changes in the damage of the bridge can be produced in real time. The algorithm would require a connection to a database containing previously computed cases for the same bridge. Moreover, it would need to be able to compute numerical models with damage scenarios to test its next best guesses, which keeps building the database over time. This implies that the longer the monitoring system is in place, the faster damage estimation can be produced. As a reference, the shorter simplified cases in this research require approximately 10 minutes to compute, but this value can increase drastically based on the assumptions made and the size of the bridge. With further investigation of the impact of damage on other components of suspension bridges through this program, the future goal is to accurately predict changes in the damage intensity of structural components in real time. The program's creation in this research is a considerable step toward reaching this goal, which forms the basis of the entire analysis process for future development in this project.

1.6 Thesis Layout

The upcoming chapters of this thesis each contribute to completing the set objectives; they are briefly introduced below.

Chapter 2 The second chapter covers the literature review, touching on subjects relevant to the

rest of the study. These subjects include introducing suspension bridge general theory and the finite element modelling of suspension bridges. Some general theory concerning the corrosion processes in steel wires and the existing health monitoring methods for suspension bridges are presented. Furthermore, a section covering wind loading and wind signal generation is provided. Finally, the last section is a short theoretical introduction to genetic algorithms.

Chapter 3 The third chapter is the one in which the LLRB model is defined. A presentation of the LLRB is made, along with the key critical features for the modelling process. Then, the modelling steps are explained, including the creation of the parts, the assembly, the steps, the constraints, the boundary conditions, the loads, and the predefined fields. The loads section also includes an explanation related to the created wind signal generator.

Chapter 4 The fourth chapter describes the program's development and how this was used to automate the generation and analysis of the models with various damage scenarios. A thorough input description is provided for the model creation, followed by the model definition steps. This chapter also includes the iterations used to calibrate the model, specifically regarding the adjustment of the tensions in the hangers. Finally, the test cases are presented for each test series performed, along with the method used to store and analyze outputs from the program automatically. The latter includes the presentation of the created mode identification algorithm, which is required for case automation.

Chapter 5 The fifth chapter presents different results obtained for the defined test cases, along with explanations of the implications of these results. This chapter is separated into three main sections. The first section covers the study of the variations in the natural frequencies of the deck's vibration modes under different damage scenarios. The second section is a complementary analysis of the changes in the vertical displacement of the deck through the damage scenarios. The third section is the lateral and torsional response of the deck under wind loading. Relevant points of discussion are approached in this chapter.

Chapter 6 The sixth and last chapter summarizes the process used to accomplish the set objectives, along with a reminder of the key findings made in Chapter 5. Suggestions for future work and limitations of this current study are also provided in this chapter.

As a note to the reader, references to code snippets use the term “Algorithm” regardless of whether the snippet’s content is an algorithm. Additionally, the term “wire” is used to describe a bundle of “strands”, where strands are the smaller element. This choice of terminology is made to avoid confusion with the term “wire” in Abaqus for 1D elements, even if the terms are in conflict with typical structural engineering terminology. Another term that does not follow typical terminology is “beam”, which designates what structural engineering theory refers to as “frame” starting in Chapter 3. This notation is also used to match Abaqus’ terminology, in which the distinction between beams and frames is different than for structural engineering theory.

Chapter 2

Literature Review

A comprehensive literature review was conducted in this chapter to gather information on subjects relevant to the content in the following chapters, which will later be used as references in this study. The first section introduces the general theory of suspension bridges (Section 2.1), followed by an overview of the basic concepts of finite element modelling (Section 2.2), with a focus on suspension bridges. Sections 2.3 and 2.4 explore the corrosion process of steel wires and existing health monitoring techniques in suspension bridges. Section 2.5 presents some important context related to wind, specifically for the wind loads acting on a structure and the theory behind wind signal generation. Thereafter, Section 2.6 presents a simplified overview of genetic algorithms.

2.1 Suspension Bridge General Theory

2.1.1 Introduction to Suspension Bridge Theory

Over the last century, a significant increase in the amount of suspension bridges built took place worldwide, with bridges extending longer spans and presenting more challenging geometries. As the structure is very light and efficient, this style of bridge is a favoured construction style, especially for bridges requiring a longer unsupported span. The main elements that compose suspension bridges are the towers, the main cables, the hangers, and the deck. The most popular model is composed of two towers and one main span model, in which the main cables are anchored at their extremity and supported at the tower tops. The role of the cables is to support the dead and live loads from the deck and transfer them to the towers. This style of design, presented in Figure 2.1 [1], makes suspension bridges a relatively flexible structure under dynamic loading. Combined with its ability to span long unsupported distances, it remains a key infrastructure in modern engineering.

This section explains how this type of design allows for efficient load distribution. Then, some details specific to this research are provided, such as cable slipping at the towers and the dynamic behaviours of the deck.

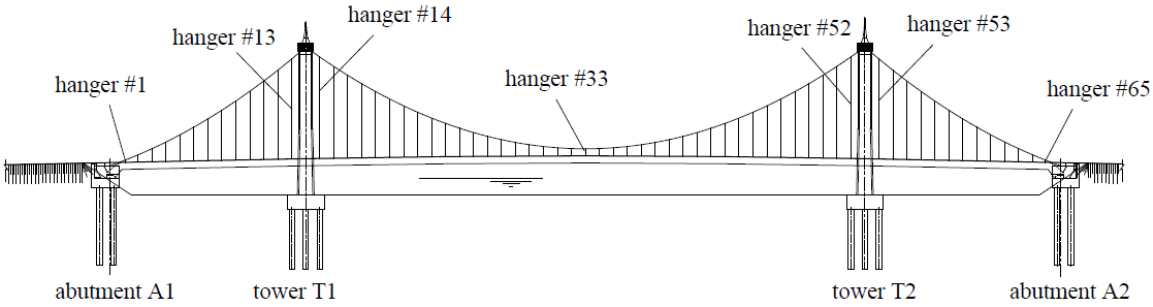


Figure 2.1: Typical Suspension Bridge Sketch [1]

2.1.2 Cable System and Load Distribution

The first important part of this section is the role of the cables within the design of the suspension bridge system. First, the hangers, or suspenders, connect vertically different points along the bridge deck to the main cables above. The intervals between consecutive hangers are typically a regular distance, ensuring the deck is supported throughout. This also allows for efficient load distribution between each hanger. The deck transfers the weight to the hangers, which then transmit it to the main cables, ultimately distributing the load to the towers and anchorages [22]. While the basic theory of this system is simple, it is essential to control the shape of the main cable such that the loaded cable's natural shape is maintained. This ensures that the bending moments within the cables are minimized. This can be better demonstrated by solving each cable connection individually, as presented in Figure 2.2.

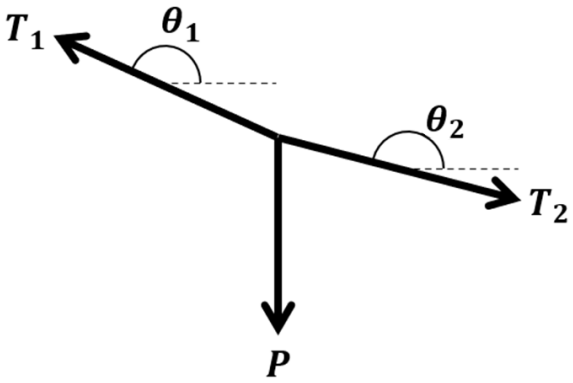


Figure 2.2: Cable Connection Forces

If the assumption is that steel wires have no bending stiffness and that the system is static,

solving for equilibrium in the forces along x and y yields the relation presented in Equation 2.1.

$$\sum F_x = 0 \rightarrow \cos(\theta_1) T_1 - \cos(\theta_2) T_2 = 0$$
$$\sum F_y = 0 \rightarrow \sin(\theta_1) T_1 - \sin(\theta_2) T_2 - P = 0$$

$$\theta_1 = \arctan\left(\frac{P}{T_x} + \tan(\theta_2)\right) \tag{2.1}$$

Where T_x is the horizontal component of tension in the main cable ($\cos(\theta_1) T_1$ or $\cos(\theta_2) T_2$). The expression for the angles is very restrictive if no bending moments are allowed within the cables. However, while the bending stiffness of the main cables and the hangers is very low when compared to structural steel elements, it provides some flexibility to Equation 2.1 by allowing moments at the joint. Overall, the angles of the main cable spans have an important role in maintaining the stability of the cable system and ensuring that loads are properly distributed throughout components [23]. When applied to actual cases, the main cable shape for the main span is a parabola when it can be assumed that the weight of the cables is negligible compared to the deck's weight. This shape is slightly different than the catenary shape of a uniformly loaded cable since most of the weight is distributed uniformly along the horizontal axis of the cable and not along the length of the cable. Ultimately, this shape allows most of the load to be transferred through tension in the cables. In turn, the main cables transfer the loads towards the towers and the anchorages. The load path for suspension bridges begins at the deck, which supports the weight dead and the live loads, and is supported by the hangers. Then, the hangers transfer this load to the main cables, which in turn transfer the loads to the towers. Finally, the towers propagate the loads to the ground. The other load transfer points for the main cables are the anchorages, which secure the cables to the ground [22].

2.1.3 Dynamic Behaviour of Suspension Bridges

The dynamic behaviour of suspension bridges is part of a complex branch of engineering. Due to their flexible nature, these infrastructures may be vulnerable to wind loading when the frequencies of the wind align with the bridge's natural frequencies. This can cause resonance even at moderate wind speeds, which can induce a great response, leading to increased fatigue and premature failure of the bridge [24, 25]. Different behaviours can be observed when certain conditions are met, such as vortex-induced vibration, galloping, flutter, and buffeting. It is critical to consider these

phenomena when designing a suspension bridge. Otherwise, some extreme cases like the Tacoma Narrows Bridge may arise [24]. Even if these aerodynamic behaviours are critical to design and analyze suspension bridges, they are not the focus of this research, they bring up the subject of natural frequencies. Natural frequencies are a critical component of bridge design, since the aforementioned behaviours are dependent on these. A comparison could be made between the natural frequencies of a suspension bridge and DNA, in the sense that it is a unique characterization of the structure. They provide a close to complete description of the infrastructure and its behaviour while not being a visible characteristic. Wind-induced behaviours of a bridge are closely related to its natural frequencies, which makes it essential to understand what they are [26].

The most significant modes of vibration of a suspension bridge are associated with the deck because that is where most of the mass of the structure is located. The vibration modes can be separated into three categories: vertical, lateral, and torsional vibration [2]. Vertical modes are recognized by the vertical movements of the deck, while lateral modes are characterized by the side-to-side motion of the deck, and torsional modes produce a twisting behaviour along the length of the deck, as presented in Figure 2.3 [2]. These modes can be categorized further into symmetric and asymmetric modes, which exist for all three types of mentioned vibration. Symmetric modes occur when displacements are mirrored on both sides of the deck about the midspan, and asymmetric modes are the opposite. To analyze the changes in behaviour of a suspension bridge, this classification is critical due to the effects of some mode combinations. Therefore, having these vibration modes available for a suspension bridge becomes vital to ensure that the aerodynamic stability of the infrastructure is not problematic.

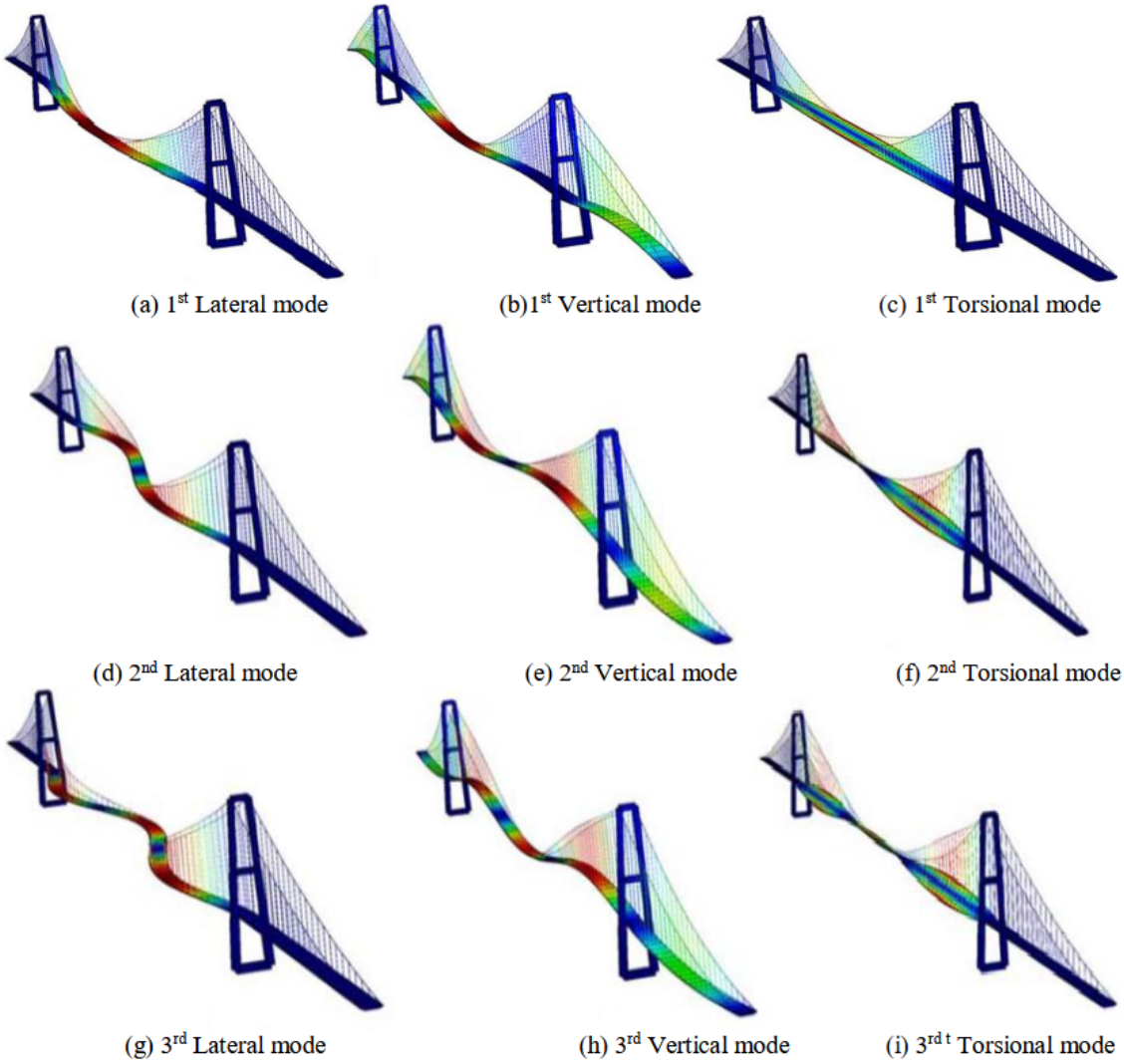


Figure 2.3: Typical Deck Modes of Vibration [2]

Many studies typically focus on reporting primarily the first modes of vibration, as they provide significant insights into the dynamic properties of a bridge. For context within this study, Figure 2.4 [3] presents first natural frequencies associated to the different modes of vibration for different suspension bridge with varying main span length of the deck.

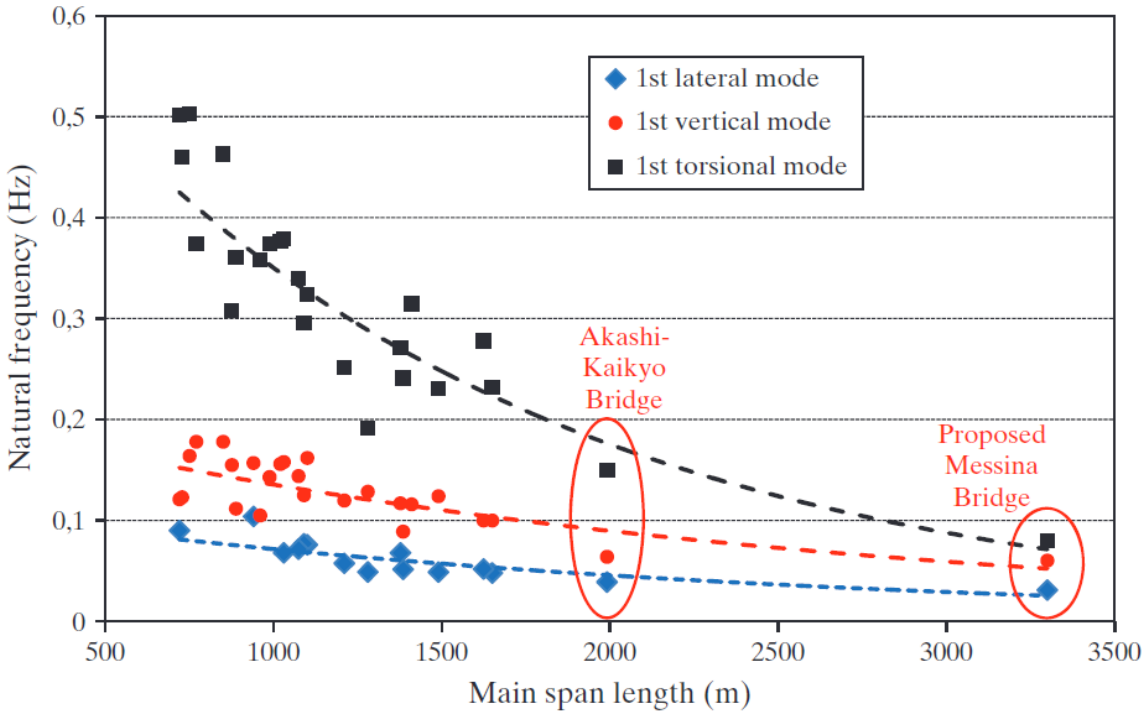


Figure 2.4: First Natural Frequencies of Deck's Vibration Modes Against Main Span Lengths [3]

2.1.4 Cable Slipping at Tower Saddles

In suspension bridges, cable slipping at the saddles installed at the top of the towers is the relative movement between the saddle and the main cables. The assumption of no cable slipping is typically justified for non-seismic loading conditions, as the friction at the saddle-to-cable contact prevents movement. This was demonstrated in a study by Zhao et al., which examined the effects of frictional slippage at the saddle-cable interface during seismic loading [27]. Through nonlinear dynamic analysis, the research compared two models, differentiated by whether the effect of slippage was considered. By comparing the results from different loadings, this study determined that significant slipping occurs only under extreme seismic loading and is otherwise negligible in the bridge model's response. Therefore, the assumption that there is no cable slipping is considered appropriate for a research that does not involve seismic loading scenarios.

2.2 Finite Element Modelling of Suspension Bridges

2.2.1 Basic Theory of Finite Element Method

Finite Element Method (FEM) is a computational technique used to simplify complex problems through discretization of a continuous domain for structural elements and systems. Such methods

produce approximate solutions to situations where exact solutions may not be achieved, at least not efficiently. Overall, FEM is a numerical method to solve partial differential equations by converting them into more manageable forms that may be solved through algebraic manipulations.

Traditional FEM theory typically considers small finite elements within an element or a structure. Differential equations may be derived from these, which form partial differential equations that describe the system. However, solving these equations becomes increasingly challenging as complexity increases. Due to the associated complex geometries, it can become impossible for real-life scenarios to get an exact solution. A simple example can be made when dealing with structure or element shape irregularities. In such cases, the partial differential equations forming the system are highly complex and may not be solved analytically.

Therefore, FEM becomes an effective alternative by discretizing the domain into finite elements instead of infinitesimal elements, which allows the partial differential equations to be approximated with algebraic equations. These equations are much easier to solve and more practical to develop and automate, a key point for using Computer Aided Engineering (CAE). However, since the domain is separated into finite elements, the solution obtained approximates the exact solution with a precision dependent on the type of the element and on the size of the defined elements. The finite elements composing the domain each represent a portion of the structure. These elements are simple shapes bound by nodes, edges and surfaces based on their dimension. Depending on the geometry type, these may be of different dimensions, such as a line in 1D, a triangle or a quadrilateral in 2D, and a tetrahedron or a hexahedron in 3D. These shapes, as presented in Figure 2.5 [4], are selected based on the specific requirements of a problem. It should also be mentioned that these shapes can be used in spaces with higher dimensions, such as a line element, which can be used in a 3D space. This is due to the fact that elements can have different degrees of freedom based on the modelling space. The network of elements created with these shapes is called meshing [4].

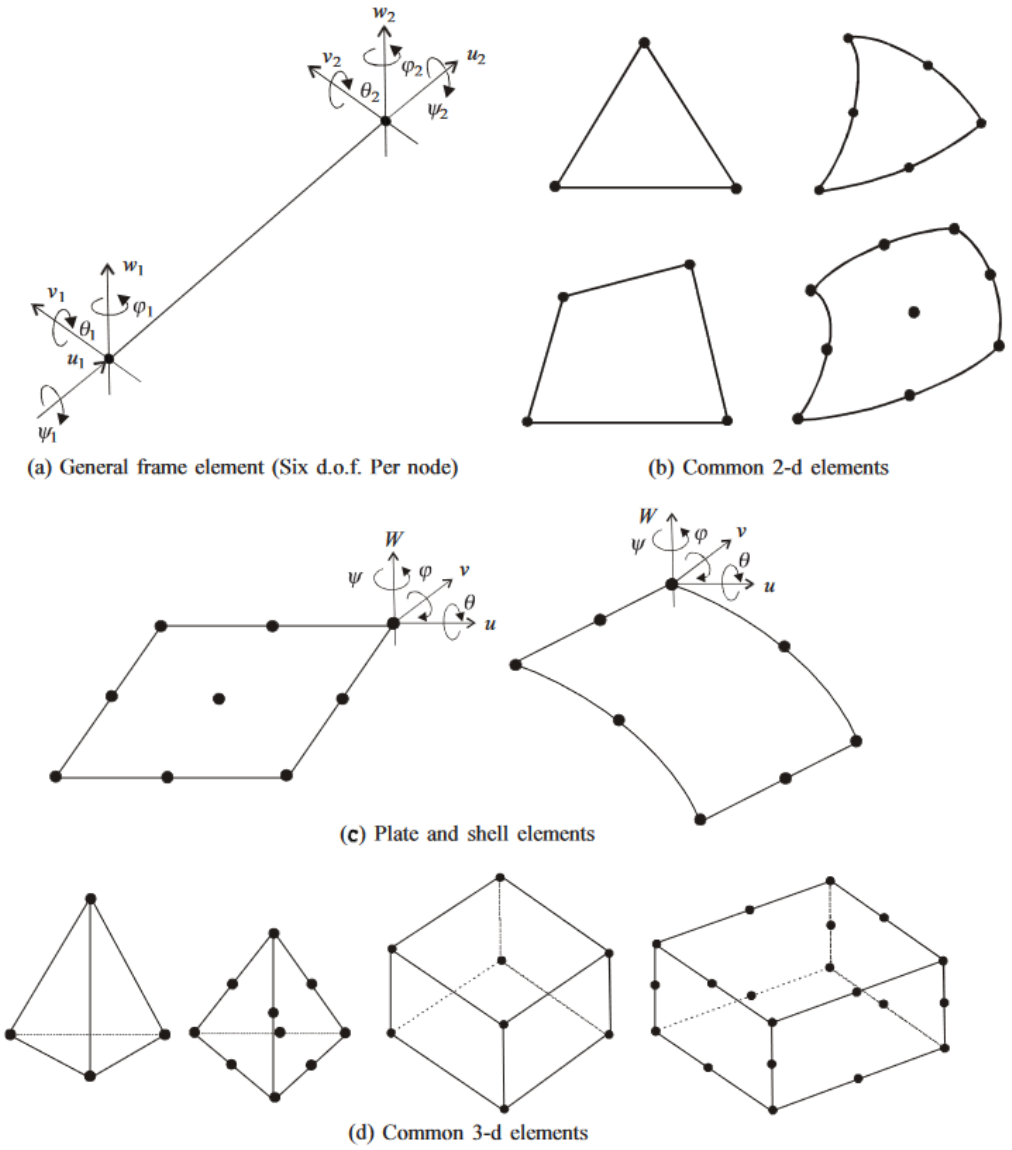


Figure 2.5: Finite Element Shapes [4]

Degrees of freedom in finite element methods can be defined as the possible displacement of a node, more specifically the translation and the rotation along the different axes of the space. The main element shape used for this work is the line, which has two nodes at each end, and can be categorized as a truss, a beam element, or a frame elment. A truss element in 3D supports only translation along its longitudinal axis, meaning it does not resist bending, torsion, and shear [28]. These elements are defined by three translational degrees of freedom per node, one along each axis in space.

On the other hand, a frame element in 3D is designed to resist axial, bending, torsion and shear stress [28]. This is due to the additional degrees of freedom in rotation about each axis, resulting in six degrees per node. The difference between these elements is presented in Figure 2.6, which

depicts the difference between truss and frame elements.

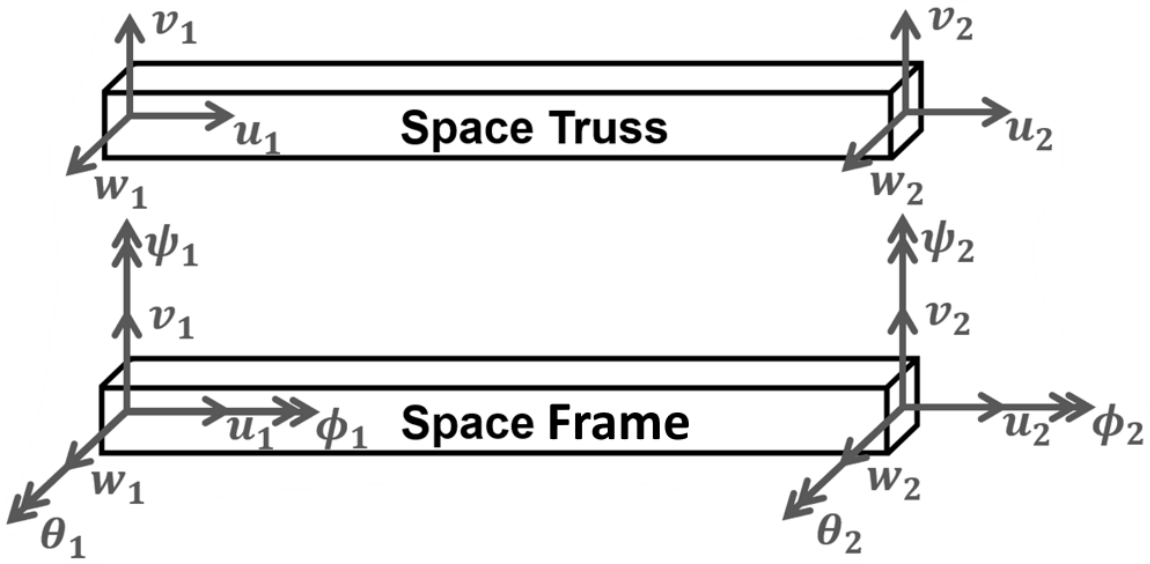


Figure 2.6: Degrees of Freedom in a Truss and a Frame Element

Each element in FEM is associated to its stiffness matrix, which is a representation of its stiffness with respect to each degree of freedom. Therefore, the stiffness matrix for a truss member is a 6x6 matrix (2 nodes of 3 DOFs each), whereas a frame member is represented by a 12x12 matrix (2 nodes of 6 DOFs each). This 12x12 matrix in a local coordinate system is presented in Equation 2.2 [29], with the following associated DOFs with reference to the DOFs from Figure 2.6: u_1, v_1, w_1, ϕ_1, ψ_1, θ_1, u_2, v_2, w_2, ϕ_2, ψ_2, and θ_2.

$$K_{frame,local} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 & 0 & 0 & \frac{-12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 \\ 0 & 0 & \frac{-6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\ \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} \\ 0 & 0 & \frac{-12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & 0 & \frac{-6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{bmatrix} \tag{2.2}$$

The stiffness matrix for a truss is not presented here but can be obtained as a subset of the frame matrix by removing the DOFs associated with rotations and considering zero values for I_y, I_z, and J, which do not exist in a truss. While these matrices are valid for local coordinate systems, they must be adjusted for their location in space in a global system to be compatible with other elements. This implies using a 12x12 transformation matrix T that considers the element's orientation in space, resulting in the following Equation 2.3 for the global stiffness matrix of a frame.

$$k_{frame,global} = T^T k_{frame,local} T \tag{2.3}$$

With the transformation matrix defined in Equation 2.6, which is derived from the multiplication of the rotation matrices for yaw (α), pitch (β) and roll (γ) in Equation 2.4 [30]. Therefore, the transformation matrix is a simple combination of this rotation matrix for each set of three degrees of freedom (x, y, z).

$$
\begin{aligned}
&R = R_z(\alpha)\, R_y(\beta)\, R_x(\gamma) \\
&R = \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\gamma) & -sin(\gamma) \\ 0 & sin(\gamma) & cos(\gamma) \end{bmatrix} \\
&R = \begin{bmatrix} cos(\alpha)cos(\beta) & cos(\alpha)sin(\beta)sin(\gamma) - sin(\alpha)cos(\gamma) & cos(\alpha)sin(\beta)cos(\gamma) + sin(\alpha)sin(\gamma) \\ sin(\alpha)cos(\beta) & sin(\alpha)sin(\beta)sin(\gamma) + cos(\alpha)cos(\gamma) & sin(\alpha)sin(\beta)cos(\gamma) - cos(\alpha)sin(\gamma) \\ -sin(\beta) & cos(\beta)sin(\gamma) & cos(\beta)cos(\gamma) \end{bmatrix}
\end{aligned} \tag{2.4}
$$

The yaw, pitch and roll angles required to rotate the orientation vector of the element within the global system of coordinates are equivalent to the Direction Cosine Matrix presented in Equation 2.5 [31]. In this representation, the x, y, and z axes are in the global system, and the x', y', and z' axes are in the local system.

$$
R = \begin{bmatrix} cos\left(\theta_{xx'}\right) & cos\left(\theta_{yx'}\right) & cos\left(\theta_{zx'}\right) \\ cos\left(\theta_{xy'}\right) & cos\left(\theta_{yy'}\right) & cos\left(\theta_{zy'}\right) \\ cos\left(\theta_{xz'}\right) & cos\left(\theta_{yz'}\right) & cos\left(\theta_{zz'}\right) \end{bmatrix} \tag{2.5}
$$

$$
T = \begin{bmatrix} [R]_{3x3} & & & \\ & [R]_{3x3} & & \\ & & [R]_{3x3} & \\ & & & [R]_{3x3} \end{bmatrix} \tag{2.6}
$$

Each element in the domain can be defined this way to form a global stiffness matrix of the system with a global coordinate system. The matrix is assembled by having a single row/column per DOF in the system, forming a square matrix of size n, the number of DOFs. Then, a vector of size n with the same sequence of DOFs can be assembled for the forces and moments applied. Through Hooke's law, a relation is established between the loads and the displacements by using the global stiffness matrix. This simple relation is presented in Equation 2.7, and forms the basis of the finite element principle [32].

$$
\overrightarrow{F} = K\,\overrightarrow{d} \tag{2.7}
$$

Where $\vec{F}$ is the load vector of size n, $\vec{d}$ is the displacement vector of size n, and K is the global

stiffness matrix previously defined. It should be mentioned that the loads can be either forces or moments, with respectively associated translations and rotations. Then, the stiffness matrix must be inverted to solve for the displacements of a known set of loads. It should also be mentioned that boundary conditions should be considered by fixing some degrees of freedom and preventing displacement for certain DOFs.

In FEM, displacements are calculated at the nodes and approximated within elements using displacement functions. These shape functions define the displacement of the element between nodes. They are generally polynomial functions that allow the element's shape to be obtained at any point. These functions are crucial, as they ensure the continuity of displacements in the model across all elements.

While this introduction presents the simplified method to solve a system when subjected to loading, it quickly becomes complex to solve a system as the number of degrees of freedom increases. This is because the time complexity of a matrix inversion is $O(n^2 m)$, with m the rank of the matrix, which should be equivalent to $O(n^3)$ for a stiffness matrix. This is also the case for a naive multiplication of a vector with a matrix, with a time complexity of $O(n^2)$. Solving larger systems by hand becomes impossible since FEM was developed to take advantage of computers. FEM software uses advanced methods to solve FEM equations more efficiently. These software, such as Abaqus, allow engineers to solve systems with many DOFs. Post-processing of the system is also performed in these software, which provides stress and strain in the elements.

2.2.2 Introduction to Structural Dynamics in FEM

With the basic theory of FEM presented, it is then necessary to discuss how dynamic behaviour is studied within this context. This part is crucial to obtain the natural frequencies of a structure under free vibration. First, let's consider the equation of motion for a simple system, which can be presented in Equation 2.8 for the typical single degree of freedom mass-spring-damper system [33, 34].

$$m\ddot{v}(t) + c\dot{v}(t) + kv(t) = F(t) \tag{2.8}$$

Where m is the mass (kg), c is the damping (Ns/m), k is the stiffness (N/m), $v(t)$ is the displacement function with respect to time (m), and $F(t)$ the applied force function with respect to time (N). Then, a free vibration implies that there are no applied forces, simplifying to Equation 2.9.

$$m\ddot{v}(t) + c\dot{v}(t) + kv(t) = 0 \tag{2.9}$$

By assuming a solution for the displacement function of the harmonic form $v(t) = \phi e^{\alpha i t}$ (where $\alpha, \phi \in \mathbb{R}$) for a oscillating response [35], the frequency can be obtained by solving the presented differential equation (2.9). The functions of velocity ($\dot{v}$) and acceleration ($\ddot{v}$) must be derived from the displacement function.

$$\begin{aligned} v(t) &= \phi e^{\alpha i t} \\ \dot{v}(t) &= \alpha i \phi e^{\alpha i t} \\ \ddot{v}(t) &= -\alpha^2 \phi e^{\alpha i t} \end{aligned}$$

These functions can be used in the defined equation of motion.

$$-m\alpha^2 \phi e^{\alpha i t} + c\alpha i \phi e^{\alpha i t} + k\phi e^{\alpha i t} = 0$$

Since $\phi e^{\alpha i t} \neq 0$, the equation can be divided by that term to create the characteristic equation (2.10).

$$-m\alpha^2 + c\alpha i + k = 0 \tag{2.10}$$

Then, solving for α through the quadratic equation:

$$\begin{aligned} \alpha &= \frac{-(ci) \pm \sqrt{(ci)^2 - 4(-m)(k)}}{2(-m)} \\ \alpha &= \frac{c}{2m} i \pm \sqrt{\frac{k}{m} - \left(\frac{c}{2m}\right)^2} \\ \alpha &= \frac{c}{2\sqrt{mk}} \sqrt{\frac{k}{m}} i \pm \sqrt{\frac{k}{m}} \sqrt{1 - \left(\frac{c}{2\sqrt{mk}}\right)^2} \\ \alpha &= \xi \omega_n i \pm \omega_n \sqrt{1 - \xi^2} \end{aligned}$$

Where:

$$\xi = \frac{c}{2\sqrt{mk}}$$
$$\omega_n = \sqrt{\frac{k}{m}}$$

Replacing into the displacement function and applying Euler's identity:

$$v(t) = e^{\left(\xi\omega_n i \pm \omega_n\sqrt{1-\xi^2}\right)it}$$
$$v(t) = e^{-\xi\omega_n t}\cos\left(\omega_n\sqrt{1-\xi^2}t\right) \pm e^{-\xi\omega_n t}\sin\left(\omega_n\sqrt{1-\xi^2}t\right)$$

From this result, the frequency of the oscillation is determined to be $\omega_n\sqrt{1-\xi^2}$, which is a combination of the undamped natural frequency ($\omega_n = \sqrt{\frac{k}{m}}$) and the damping ratio ($\xi = \frac{c}{2\sqrt{mk}}$). This solution is valid for underdamped cases in which the damping ratio is less than 1 [36].

While this demonstration applies to a single degree of freedom system, a similar procedure can be used for multiple degrees. Different displacement functions should be assumed for each DOF, and then differential equations can be assembled for each DOF, considering the interactions with adjacent DOFs. By solving these equations for non-trivial solutions, the eigenvalues/frequencies of the system can be determined with one frequency per DOF. This manual process quickly becomes obsolete with the addition of more DOFs, which is where FEM comes in.

With FEM, the property matrices (mass, stiffness, and damping) can be used to solve for the natural frequencies of the system. This is accomplished with the multiple degrees of freedom equation (2.11) for free vibration.

$$M\ddot{v}\left(t\right) + C\dot{v}\left(t\right) + Kv\left(t\right) = 0 \tag{2.11}$$

Where M is the mass matrix, C is the damping matrix, K is the stiffness matrix, and $v(t)$ is the displacement vector. The size of the matrices and the vectors are $n \times n$ and $n \times 1$ respectively, with n the number of natural frequencies. As a simplification for this demonstration, an undamped system is considered, with $C = 0_{n\times n}$. A harmonic solution in Equation 2.12 is assumed for each degree of freedom, which can be derived twice to obtain the acceleration solution in Equation 2.13.

$$v(t) = \Phi e^{i\omega t} \tag{2.12}$$

$$\ddot{v}(t) = -\omega^2 \Phi e^{i\omega t} \tag{2.13}$$

Where Φ is the mode shape vector of size n. Substituting in the equation of motion, yielding the generalized eigenvalue Equation (2.14):

$$\begin{aligned} -M\omega^2\Phi e^{i\omega t} + K\Phi e^{i\omega t} &= 0 \\ \left(K - M\omega^2\right)\Phi &= 0 \end{aligned} \tag{2.14}$$

This represents a generalized eigenvalue problem in which non-trivial solutions exist when the determinant of $K - M\omega^2$ is not zero, as presented in Equation 2.15.

$$\det\left(K - M\omega^2\right) = 0 \tag{2.15}$$

Solutions to this equation yield eigenvalues ω_i^2 for the system, where i is the mode (from 1 to n). The natural frequencies (ω_i) are then obtained from the eigenvalues. With the eigenvalues determined, the associated mode shape vectors can be computed in order to evaluate the participation factors and the effective masses for each mode.

Participation factors are a measure of the contribution of each mode to the system's response in a direction (per DOF). This is essential information to determine which mode shapes contribute to the system's displacement in a direction, which is critical for vibration analyses. For a multiple degree of freedom system, Equation 2.16 can be used to determine the participation factor of a mode [37].

$$\Gamma_i = \frac{\Phi_i^T M r}{\Phi_i^T M \Phi_i} \tag{2.16}$$

Where Γ_i is the participation factor for mode i, Φ_i is the mode shape vector for mode i, M is the mass matrix, and r is the direction vector. The direction vector should be defined with respect to the orientation of the DOF in the global coordinates system. The modal mass $\Phi_i^T M \Phi_i$ can be identified as a different representation of the mass matrix, where the mass is associated with each mode in the modal system, obtained from the mode shapes [38]. This representation is a decoupled version of the matrix into independent modes, meaning each mode can be analyzed individually.

Back to the participation factor, a high participation factor for a mode indicates that this mode has an important contribution to the system's motion in the specified direction vector [37].

On the other hand, the effective masses presented in Equation 2.17 are a quantification of the contribution of each mode to the total mass movement in response to a dynamic excitation in a direction [37].

$$M_{eff_i} = \frac{\left(\Phi_i^T M r\right)^2}{\Phi_i^T M \Phi_i} = \Gamma_i^2 \Phi_i^T M \Phi_i \tag{2.17}$$

Where M_{eff_i} is the effective mass (kg) of mode i for the direction r. This metric represents how much of the system's mass is influenced by each mode in each direction, which is another critical point for vibration analyses.

2.2.3 FEM-Based Fishbone Models for Suspension Bridges

Fishbone models are a simplified representation of suspension bridges, particularly suited for this type of model due to the flexible nature of such bridges. This type of model reduces the complexity of suspension bridges through the use of one-dimensional frame and truss elements to represent the real components [39]. In this representation, the deck can be taken as the "spine" of the bridge and the hangers as branching "ribs", as presented in Figure 2.7 [5] for an idealization of this concept. The simplicity of the fishbone model then comes from a geometric simplification of its components as an interconnected system of line elements. This makes fishbone models computationally efficient while still capturing the essential behaviours of the original structure. This approach is beneficial for preliminary design studies, large-scale parametric analyses, and modal analyses, in which computational efficiency is critical.

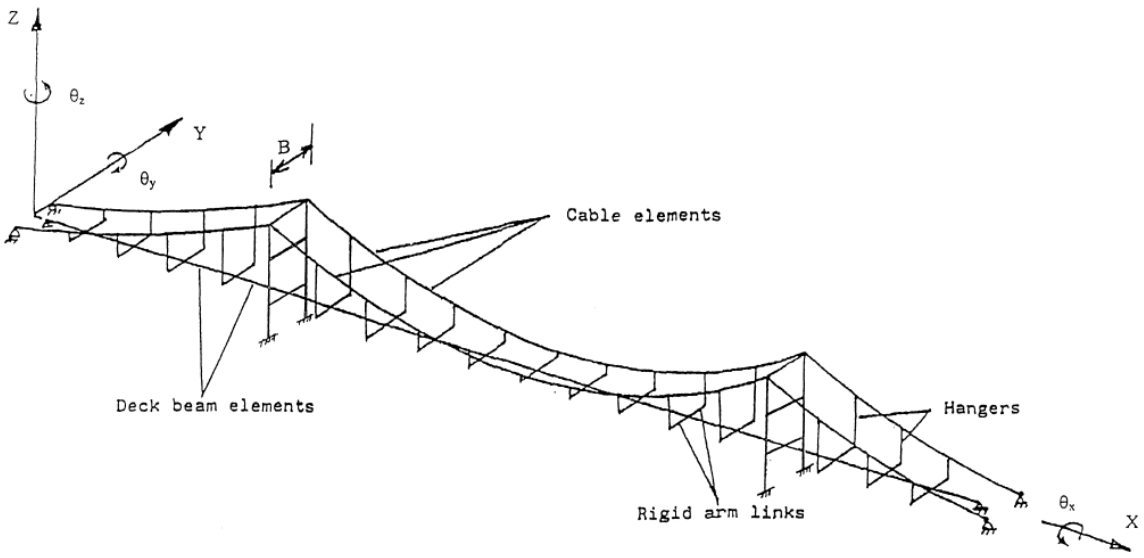


Figure 2.7: Fishbone Model Idealization [5]

This type of model has been used for most FEM analyses of suspension bridges due to the advantages it provides. Multiple examples of this model usage are found in recent literature, such as the study on contact and slip behaviours of suspension bridge's main cables [40], the study of the effects of bending stiffness of main cables on flexural and torsional vibrations of suspension bridges [41], the study of the nonlinear dynamic response under sudden hanger breakage [1], or the study of the performance of double-deck plate-truss composite steel girders of a multi-span (4) suspension bridge [42]. Overall, these recent examples only demonstrate a glimpse of the various uses of this very popular model.

The computational efficiency of fishbone models lies in using line elements instead of more realistic and detailed three-dimensional elements. Therefore, the number of elements is greatly reduced, significantly reducing the number of nodes per meshing element. This makes the size of the properties matrices that describe the system multiple orders of magnitude smaller. As per the theory presented in Section 2.2.1, this significantly accelerates the computation of results by substantially reducing the number of required operations.

However, while computationally efficient, fishbone models do present some limitations. The most significant drawback is that since components are modelled as line elements, most localized behaviours and stresses are lost in the simplification process. This is especially true for elements that are more complex in their three-dimensional form, such as deck elements. Most local behaviour is missed by reducing these components to line elements, which is a trade-off between accuracy and efficiency. As such, fishbone models should be utilized only in appropriate situations where detailed local response evaluation is not the main focus.

An additional aspect to evaluate is the choice to use truss or frame elements to represent the cables in the fishbone model. Since cables have a low resistance to bending, they are typically modelled as truss elements in larger models to reduce the degrees of freedom required to compute the model. However, this bending resistance is not negligible and can be considered with a small value of the second moments of area in frame elements [43]. While this increases the total number of DOFs in the model and the computation time considerably, it should generally produce a more stable model with an improved representation of the cables.

2.3 Corrosion Process in Steel Wires

2.3.1 Mechanisms of Corrosion in Steel Wires of Suspension Bridges

Corrosion is a common phenomenon in which a material is subjected to degradation due to reactions with corrosive agents in its environment. In general, the material in question is a metal, which is transformed into a more stable oxide, altering the properties of the metal in the process. When iron is the metal subject to corrosion, the resulting byproduct of the process is rust. This rusting process is caused by the contact of iron with oxygen and water, which forms oxides, as presented in Figure 2.8 [6], which demonstrates the electrochemical reactions.

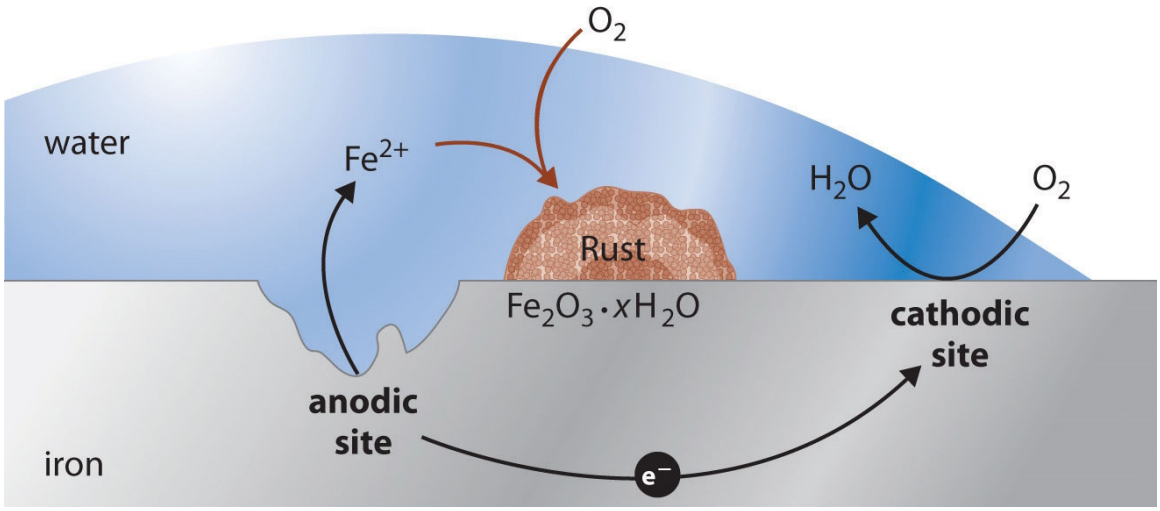


Figure 2.8: Iron Corrosion Process [6]

The anodic site reaction is the process in which the iron is oxidized and loses electrons to form iron ions, which are then dissolved into the water (Equation 2.18).

$$\mathrm{Fe(s)} \longrightarrow \mathrm{Fe^{2+}(aq)} + 2\,\mathrm{e^-} \tag{2.18}$$

The electrons from the anodic site travel through the iron member to the cathodic site to reduce the oxygen (Equation 2.19).

$$\mathrm{O_2(g)} + 4\,\mathrm{H^+(aq)} + 4\,\mathrm{e^-} \longrightarrow 2\,\mathrm{H_2O(l)} \tag{2.19}$$

The iron ions further react with water and oxygen to form the hydrated iron oxide (Equation 2.20), which is rust in this context.

$$4\,\mathrm{Fe^{2+}(s)} + 3\,\mathrm{O_2(g)} + 2\,\mathrm{nH_2O(aq)} \longrightarrow 2\,\mathrm{Fe_2O_3 \cdot nH_2O(aq)} \tag{2.20}$$

The resulting iron oxide forms a flaky and porous layer that provides little to no protection to the underlying iron, allowing the corrosion process to continue. As expected, the iron oxide byproduct does not contribute to the structural integrity of the member. Therefore, as the iron is transformed, it reduces the member's capabilities. Overall, steel also follows this rusting process, as it is mainly composed of iron. However, several methods exist to prevent iron from rusting, such as galvanizing the steel. Typically, galvanizing consists of coating a steel element with a layer of zinc, which is both a barrier and a sacrificial method of iron protection. The zinc layer forms a protective coating around the component, preventing oxygen and water from reaching the iron, which is necessary for the corrosion process to occur. Moreover, if the zinc layer is damaged and the iron is exposed, the zinc is more reactive with oxygen and should be oxidized instead of the iron. This prolongates the underlying material's lifespan proportionally to the zinc coating's thickness and the layer's deterioration rate [44].

While galvanized coatings are adaptive to multiple environments due to the corrosion-resistant layers that they create, they are often subjected to localized corrosion and pitting. Persson et al. [45] reinforce that these types of damage are of particular concern, as they may lead to damage that penetrates the protective zinc layer, exposing the steel to direct environmental attacks. While pitting is usually more critical with high exposure to chlorides and sulphates, it is expected to happen over long periods on steel cables in environments with high moisture and varying temperatures. In these conditions, the zinc layer eventually does not fully protect the steel, leading to the development of anodic and cathodic sites. These sites allow for the corrosion of the steel, enhancing the pitting in the member. In harsh environments, the structural integrity of the members can be maintained longer by simply increasing the thickness of the zinc coating. Alternatively, some

methods are used nowadays to increase the lifetime of the coating, such as adding small amounts of aluminum or magnesium, which form more stable corrosion products that control water infiltration into the steel [45].

A study by Li et al. [46] demonstrates that high humidity and temperature fluctuations can affect the corrosion rates of galvanized steel. This research focuses on galvanized steel wires in bridge cables, making it highly relevant to the current study. The paper demonstrates that temperature increases accelerate the corrosion reaction by decreasing the required activation energy. At such temperatures, other factors, such as the intensified molecular movement and accelerated ion migration rate, made the zinc leave the corrosion area without forming a protective layer. When combined with moisture exposure, localized degradation progressively occurs over time.

Expanding on the previously mentioned aluminum and magnesium additions to coatings, a recent study by Zhang et al. [47] examined the effects of the coating composition on stress corrosion resistance. Stress corrosion occurs when a metal undergoes cracking from the combination of a corrosive environment and stress. More specifically, this can happen in environments with high moisture or fluctuating temperatures where a steel member is under tensile stress. These combined factors accelerate degradation and pose a significant risk to steel cables, where small stress-induced cracks can lead to failure under continuous loading. The aforementioned article demonstrates that zinc coatings protect steel cables but are far from being the most effective solution against stress corrosion. Zinc alone has a porous nature as it starts to corrode, which can flake easily and expose the steel it should be protecting.

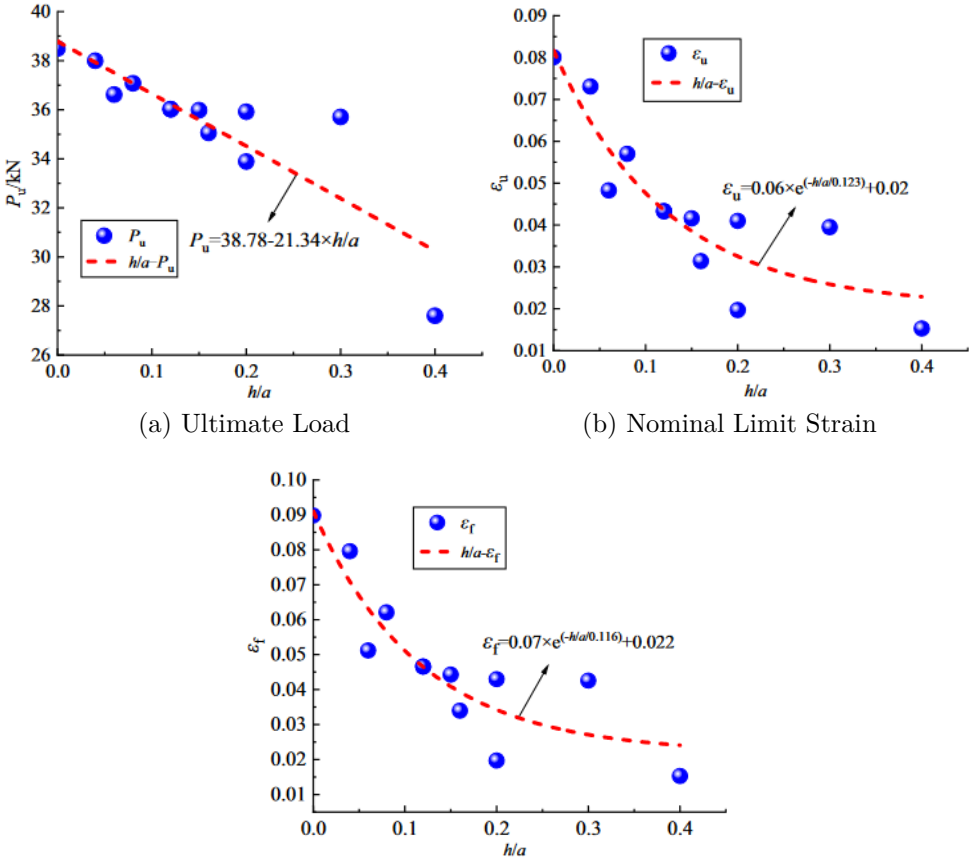
On the other hand, other compositions such as zinc-aluminum (ZnAl) and zinc-aluminum-magnesium (ZnAlMg) coatings have a much better performance as aluminum creates a stronger and more stable protective layer as it starts to corrode [47]. This layer is also made of denser oxides that are less subject to cracking and flaking. Moreover, magnesium does not produce soluble or porous corrosion products, which are prone to water penetration. The corrosion products, therefore, protect the member for better long-term performance [47].

These articles are particularly relevant to this research, since the Lower Liard River Bridge cables are composed of galvanized steel wires and most likely use a pure zinc coating based on the year of erection of the bridge. Therefore, this 80 years old bridge has been exposed to high moisture levels and considerable temperature fluctuations since it crosses a river in northern Canada. It is expected that such conditions encounter some corrosion damage in the steel wires forming the hangers and the main cables of the bridge. According to the explanations from this section, two expected scenarios for wire damage can be identified. The first type is pitting in some strands on

the wires outside at locations where the protective zinc layer has been breached. The second type is where some strands in the wires failed and snapped completely. In both cases, corrosion can be regarded as highly localized.

2.3.2 Impact of Corrosion on Structural Properties of Steel Cables

Corrosion has a significant impact on the mechanical properties of steel cables. As the article by Kang et al. [7] explains, the presence of corrosion pits on steel cables leads to a localized stress concentration. In this study, stress levels were more significant at the bottom of the pit, increasing the crack propagation under tensile loading. Moreover, as steel is affected by corrosion, its ductility is reduced, which causes a shift from ductile to brittle fracture mode. Another point demonstrated in this article is that as the depth-to-width ratio of the pit increases, the internal stresses increase as well, causing accentuated crack growth. Then, as that ratio increases, a reduction in the ultimate load-bearing capacity and the fracture strain are noticed. As Figure 2.9 [7] presents, the limit and plastic strains are reduced according to an exponentially decaying function as the depth-to-width ratio of the pit increases. On the other hand, it is interesting that the ultimate load for the damaged cables decreases linearly with the depth-to-width ratio.



(a) Ultimate Load (b) Nominal Limit Strain

(c) Nominal Fracture Strain

Figure 2.9: Corrosion Pit Mechanical Properties Changes For Depth-To-Width Ratio [7]

While the changes in mechanical properties are presented under different damages in the cables, they are still limited to the pit's shape and have significant variations. For example, many values of depth-to-width ratio are associated with a fracture strain value of 0.043-0.047 or a limit strain value of 0.040-0.043. This might indicate that another factor could be involved or that the depth and the width should be considered as separate measurements instead of a ratio.

Nakamura and Suzumura's [48] study demonstrates that the tensile strength of the remaining area of corroded wires does not significantly decrease. However, corrosion damage substantially decreases the ductility, torsional strength, and fatigue strength. This implies that an affected wire cannot sustain the same strain levels before failure. Such findings agree with those reported by Kang et al. [7], with the loss of ductility in the remaining steel and the linear decrease in the ultimate load that can be attributed to the cross-sectional area reduction. This study also states that the initial mass loss is due to the degradation of the zinc layer; reductions in the other mechanical properties become critical as the corrosion gets to the core of the wire, the steel core. As such, the combined studies demonstrate that the steel core is exposed and becomes less ductile

as the zinc layer is penetrated. This loss of ductility increases the strand's susceptibility to failure under cyclic loading.

The study by Jeon et al. [8] further reinforces the previously mentioned relations. However, as presented in Figure 2.10 [8], this study includes different corrosion patterns in the results.

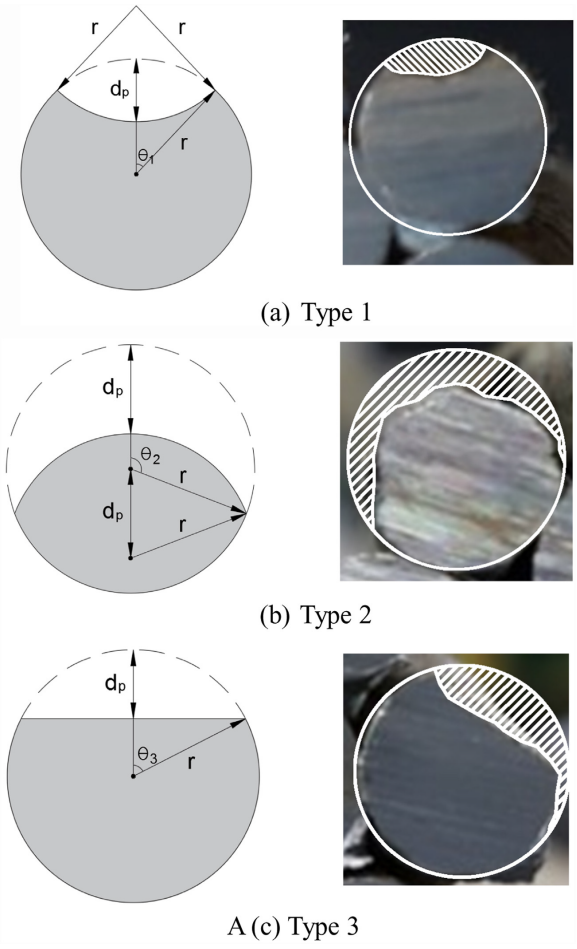


Figure 2.10: Cable Corrosion Damage Pattern Types [8]

By simulating different pit depths and patterns with finite element modelling, this study was able to establish correlations between the extent of a loss of area and the tensile strength and fracture strain. These values obtained are presented in Figure 2.11 [8] below. For the fracture strain specifically, it is observed that the double-fitting equation is more convincing than the depth-to-width ratio relations from Figure 2.9 [7]. Furthermore, the ultimate stress is also noticed to vary linearly with the section loss, which matches some relations from different other studies. However, it is also noticed that the linearly forecasted values of ultimate stress at a section loss of 100% are less than zero. This indicates that an additional fit might be necessary at higher section losses to properly describe the ultimate stress.

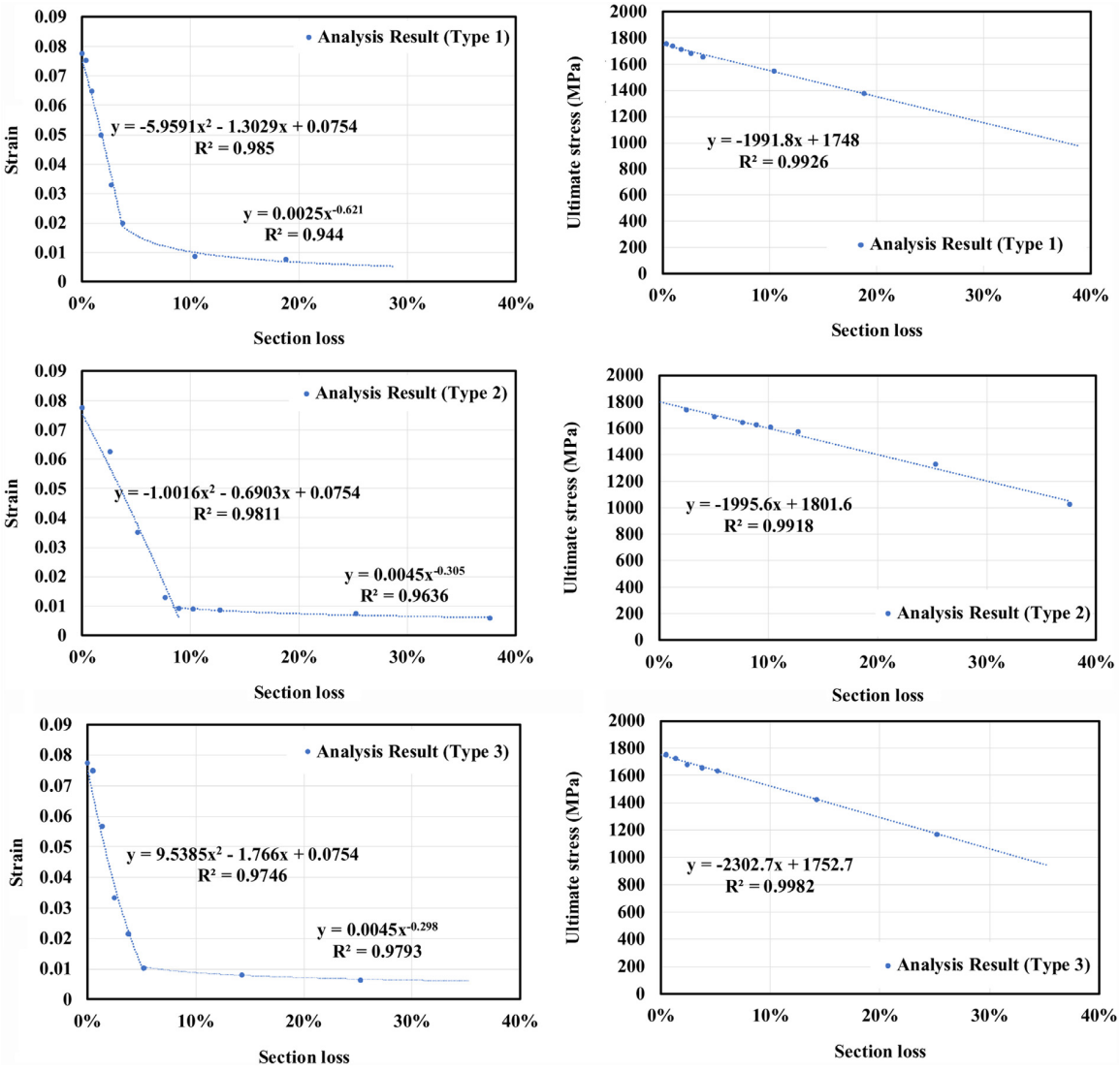


Figure 2.11: Corrosion Pit Mechanical Properties Changes For Section Loss Per Damage Pattern Type [8]

Ultimately, this study found that the ultimate stress in a corroded wire is not the most affected property. On the other hand, ductility is reduced critically, even at relatively low levels of corrosion or area loss from corrosion in a wire. This decreases the energy a wire can absorb before moving to a plastic behaviour or fracture point. This is an indication that a small section loss causes that individual member to break at considerably lower loads, which is a justification for the modelling of corrosion damage as cross-sectional area reduction for this type of analysis.

However, these studies apply to a single strand with corrosion-induced damage (using the alternative terminology mentioned in Chapter 1 with wires being composed of strands). In a larger cable wire composed of many strands, some strands on the outside of the wire often get their zinc layer damaged. Once the protective layer is breached and the steel is exposed, it undergoes changes

in mechanical properties discussed in this subsection, such as the change in ductility. Under cyclic loading, these strands eventually break completely, which cancels their structural contribution to the wire. When this occurs and it is the primary source of corrosion damage in a cable, it may be reasonable to simplify the damage as a reduction in the wire's cross-sectional area. This modelling assumption replicates broken strands that no longer contribute structurally while disregarding damaged wires that still provide some structural support. This simplification is more appropriate when the number of broken strands exceeds the number of damaged but intact strands. However, it may not be suitable for studying localized damage behaviour.

2.3.3 Higher Risk Locations For Main Cable Corrosion in Suspension Bridges

While it is critical to understand how corrosion forms and its impact on the properties of a steel wire, it is also essential to know which locations in a suspension bridge are typically more prone to corrosion in the main cable. A report from Hopwood II and Havens [49] details this topic, identifying vulnerabilities to corrosion in the regions near the towers and the anchorages. According to this report, areas near the towers are typically more exposed to moisture retention and stress concentration, causing an increased risk of corrosion. The other vulnerable location is at the anchor because it is usually more exposed to ground-level moisture and runoff. Also, if there is inadequate drainage, water pooling at the cables can become a considerable issue that causes severe corrosion if unaddressed. Another location that can be proportionally more affected by corrosion is the underside of the wires, which tends to retain moisture. However, the report also presents a damage location analysis, revealing that many corrosion sites appear to be randomly distributed. On average, some locations should experience more critical corrosion, but localized damage can occur anywhere along a cable. This occurs because imperfections in the zinc layer or external damage to it are randomly distributed and do not follow a predictable pattern over time.

This is especially true for the Lower Liard River Bridge as per the 2011 visual inspection [9]. This inspection demonstrates that damage in the main cables is located all along the length of the bridge, with an exception at the anchor, which seems to be unaffected. Table 2.1 [9] presents in detail the damage distribution on the bridge, which is more concentrated in the bridge's main span. However, no other patterns can be recognized directly from these results, which reinforces that a considerable part of the damage location can be stochastic.

Table 2.1: Lower Liard River Bridge Main Cable Corrosion and Damage Locations [9]

Element	2009			2011			Observations	Photo References
	MCR	PCR	Priority	MCR	PCR	Priority		
Main Suspension Cables – East								
Note: Naming convention identical to as-built drawings and previous reports								
South Abutment								
0S-2S	5	5	D	5	5	D	No visible defects.	
2S-4S	5	5	D	5	5	D	No visible defects.	
4S-6S	5	5	D	5	5	D	No visible defects.	
6S-8S	5	5	D	5	5	D	1 #7 and 3 #8 broken wires clamped – 3 clamps on #8, rust under clamps	56,57
8S-10S	5	5	D	5	5	D	1 #5 broken wire clamped	58
10S-12S	5	5	D	5	5	D	3 #1 broken wires double clamped	59
South Tower								
12S-14S	5	5	D	5	5	D	No visible defects.	
14S-16S	5	5	D	5	5	D	1 #5, 2 #9 and 2 #8 broken wires clamped	67,68,69
16S-18S	5	5	D	5	5	D	1 #8, 1 #10 and 1 #11 broken wires clamped	70,71,72
18S-20S	5	5	D	5	5	D	1 #1 and 1 #10 broken wires clamped	73
20S-22S	5	5	D	5	5	D	2 #1, 1 #7, 1 #8, and 1 #10 broken wires, 2 clamps on #1, 1 clamp on #7, 1 clamp on #8, and 1 clamp on #10	74,75,76
22S-24S	5	5	D	5	5	D	1 #7 broken wire clamped	77
24S-26S	5	5	D	5	5	D	1 clamp on #8	78
Midspan								
26N-24N	5	5	D	5	5	D	2 #1 broken wires clamped	84,85
24N-22N	5	5	D	5	5	D	1 #7 broken wire clamped	83
22N-20N	5	5	D	5	5	D	3 #10 broken wires clamped – 2 clamps in 1 location and 1 clamp in another	81,82
20N-18N	5	5	D	5	5	D	No visible defects.	
18N-16N	5	5	D	5	5	D	1 #5 and 2 #7 broken wires clamped	79,80
16N-14N	5	5	D	5	5	D	No visible defects.	
14N-12N	5	5	D	5	5	D	No visible defects.	
North Tower								
12N-10N	5	5	D	5	5	D	No visible defects.	
10N-8N	5	5	D	5	5	D	1 #11 broken wire clamped	88
8N-6N	5	5	D	5	5	D	No visible defects.	
6N-4N	5	5	D	5	5	D	No visible defects.	
4N-2N	5	5	D	5	5	D	No visible defects.	
2N-0N	5	5	D	5	5	D	No visible defects.	
North Abutment								

Moreover, to enforce the previous point, most of the damage reported is broken strands that have been clamped. Figure 2.12 [9] is a clear example of such broken wires, in which clamps hold the fractured strands in place. However, there is no significant discolouration, pitting, or visible rust in the pictures from the inspection, which would have indicated advanced corrosion. This suggests that some of the strands had their zinc layer damaged, which exposed the steel below and reduced ductility from corrosion. Since the fracture strain of a corroded wire is reduced, it is therefore prone to breaking. The fact that no other corrosion is noticeable indicates that it is only localized damage in some strands. As such, the remaining strands are not damaged or affected by the corrosion in the broken strands, meaning that this damage is simply a cross-sectional area reduction in the wire.



Figure 2.12: Lower Liard River Bridge Cable Damage Picture From Visual Inspection [9]

2.4 Health Monitoring in Suspension Bridges

2.4.1 Introduction to Structural Health Monitoring

Structural Health Monitoring (SHM) is an essential component in the maintenance of suspension bridges. It aims to detect and evaluate damage early to ensure the integrity of the infrastructure over its lifespan. Depending on the method used, this can be achieved through either continuous or periodic assessment of the components. Effective SHM techniques seek to detect damage before it threatens the structure's structural integrity. By identifying damage in the bridge, appropriate interventions are possible to prevent failure in the components [50]. This subject is particularly relevant to this research since the long-term objective is to automate a structural health monitoring system.

One of the critical components often at the core of such assessment is the main cables, which are vulnerable to corrosion and can significantly affect the bridge's structural integrity. As presented in Section 2.3.2, when strands of the main cables are affected by corrosion, ductility in the remaining steel is considerably affected, making them prone to failure. Therefore, even relatively minor damage to the steel wires can lead to disproportionately large impacts on the overall structural integrity over time. Different levels of damage evaluation exist, which describe the level of precision

of an analysis [51, 52, 53, 54]:

1. Level 1: Determination of the existence of damage
2. Level 2: Determination of the location of the damage
3. Level 3: Determination of the type of damage
4. Level 4: Determination of the extent of the damage
5. Level 5: Prediction of the remaining service life of the structure

One commonly used SHM technique is vibration-based monitoring, which leverages the fact that a structure's dynamic response changes when its components sustain damage. Within vibration-based monitoring, natural frequency-based methods are a recognized technique that offers some advantages but also some limitations to studying structural changes.

This section first presents some of the SHM techniques currently in use. It then dives deeper into natural frequency-based methods, which are the focus of this research. Subsequently, the advantages and limitations of such methods are presented, followed by some application examples in the suspension bridge field.

2.4.2 SHM Techniques for Suspension Bridges

The health monitoring techniques for suspension bridges have evolved to incorporate traditional and modern technologies in damage detection. Most of these techniques can traditionally be separated into four main categories: visual inspections, electromagnetic methods, acoustic emission monitoring, and vibration-based methods.

Visual inspections are the most traditional and widely used method in SHM of suspension bridges. These involve human inspections and observations to identify visible damage, such as cracks, deformation, or corrosion in components, as is the case for the LLRB [9]. This method has the advantage of being highly simple, but visual inspections are labour-intensive, time-consuming, and prone to human error. Moreover, they are limited to the detection of damage at the surface of components, as the material below the surface is not exposed. Some modern approaches incorporate more advanced tools, such as drones with high-resolution cameras and sensors [55], but this remains a limited solution for the aforementioned reasons.

On the other hand, electromagnetic methods are more effective at detecting internal defects and corrosion. By identifying variations in the magnetic properties of the structural components,

these techniques detect anomalies in cables, which in turn is generally a good indicator of damage. Magnetic flux leakage and eddy current testing are commonly used among these methods. For example, Dongfeng He [56] demonstrated the use of a portage eddy current system to estimate levels of corrosion in galvanized steel wires. Overall, these studies demonstrate the potential of detecting early-stage corrosion in cables.

Another branch of techniques is acoustic emission monitoring, which detects stress waves from wire breaks or crack propagation. This real-time monitoring method is effective as a warning system for suspension bridges. As such, continuous acoustic monitoring allows for real-time identification and location of damaged areas. A notable early application of this method was presented by Paulson [57], who explored the possibility of monitoring large suspension cables to detect individual wire breaks.

Finally, vibration-based methods are one of the most advanced and widely studied SHM branches. These methods analyze changes in the dynamic properties of components to detect possible damage. These properties may include natural frequencies, mode shapes, and damping ratios. Overall, vibration-based approaches result in an assessment of the global stiffness change, which can indicate corrosion. For example, Fan and Qiao [58] presented the effectiveness of natural frequency analysis to in assessing global damage. In contrast, mode shape curvature demonstrated its capability at localizing stiffness reductions in individual components [59]. Despite the numerous advantages of vibration-based methods, they generally lack precision at localizing damage due to a lack of sensitivity.

2.4.3 Natural Frequency-Based Methods

For this research, the main objective is related to the vibration-based method, specifically the detection of changes in natural frequencies in suspension bridges. Natural frequency-based methods leverage the fact that a system's natural frequencies depend on the stiffness of the components forming the system. If a component is damaged and suffers a change in stiffness, this impacts the natural frequencies of the entire system (Section 2.2.2 for more details on natural frequencies theory). Therefore, the fundamental principle of natural frequency-based methods uses the shifts in natural frequencies to retrospectively determine the changes in stiffness, which can be linked to damage.

Over the past decades, the literature has demonstrated that natural frequency-based methods for damage detection within the context of structural health monitoring is a viable option. This was observed in Kim and Stubbs' work [60], where it is clear that frequency shifts can pinpoint

crack locations by correlating them with stiffness reductions in single beam-like structures. Moreover, Zhang et al. [61] proposed an approach that uses the curvature of frequency shift surfaces for plate structures, which were noticed to be sensitive to local damage. Notably, Salawu [62] presents a comprehensive review that demonstrates how various structural damage detection can be achieved through the use of natural frequencies. This review demonstrated that changes in natural frequencies can localize damage in beams with a precision of 3% of their length in experimental conditions [63]. Similarly, the ratio of frequency changes in different modes was used to identify stiffness variations and locate damage [64]. However, Salawu [62] also reported that these approaches are computationally intensive and depend on accurate finite element models to produce accurate results.

Despite their effectiveness, natural frequency-based methods are not without their limitations. Localized and subtle damage in a structure is not easily detectable with such methods, as the shifts in natural frequencies primarily reflect changes in the global stiffness of a system. Another consideration that is often dismissed is that the dynamic response of a structure is also dependent on environmental factors, such as temperature, humidity, and air pressure. These factors can, therefore, affect the measured frequencies, as demonstrated by Askegaard and Mossing, who stated that natural frequency measurements of an undamaged structure remain consistent when taken at the same time of the year [62, 65]. This indicates that while considerable fluctuations in the environmental factors are significant, smaller changes in these factors are not the leading cause of shifts in natural frequencies. This study also concluded that variations during short-term tests are negligible. However, the influence of the environment can be noticed over extended periods of analysis while still having less of an impact than deterioration in the structure [65]. Finally, one main limitation is the requirement of a baseline data set for the undamaged structure. Without such data, it becomes a great challenge to distinguish structural damage from environmental or operational variations. However, in most cases, these values are unavailable, which means that further calibrations are required to infer damage on a structure [62].

Overall, natural frequency-based methods are viable for monitoring structural state and behaviour when periodic vibration testing is performed. However, Salawu also stated that computational intensity is a considerable drawback for large civil engineering structures with this type of method. For this reason, it was concluded that natural frequency changes alone are not sufficient to identify damage locations in a structure. Finally, Salawu [62] concluded his review by asserting that assessing the integrity of civil engineering structures using natural frequency-based methods would need to be based only on the first few critical vibration modes and incorporate simplifying

assumptions about how the structure behaves.

In recent years, Wickramasinghe et al.'s [66] study covered the effectiveness of vibration-based damage indicators under specific damage scenarios. They used a laboratory model of a suspension bridge to simulate damage conditions and examine the resulting changes in natural frequencies. Their findings are a demonstration that shifts in natural frequencies of the bridge are observed when the cables of the bridge are subjected to damage, which is a strong indication of the potential that vibration-based methods present for damage identification in suspension bridges.

An independent study [67] also demonstrated that a correlation can be established between measured changes in a bridge's natural frequencies and simulated damage to the deck components. This research used the interpolation damage detection method of a numerical model to obtain level 2 damage identifications. This study also conducted seismic excitation analyses, but solely focused on damage to the deck, not the cables.

Xiang et al. [68] performed an analysis on a cable-stayed bridge by applying different damage patterns to the cables and analyzing the first 30 natural frequencies. By comparing the damaged bridge's frequency shifts from an undamaged model, some correlations were observed with the applied damage. However, this study only analyzed the direct natural frequencies of the bridge without considering the mode shapes, making it difficult to track changes between the different models.

In the end, natural frequency-based methods have proven their efficiency as a tool for structural health monitoring. With the analysis of shifts in natural frequencies, more recent studies have concluded that detecting changes in the stiffness is possible. Such methods are well-supported in the literature, from precise damage location identification in singular beams to analysis of complex systems such as suspension bridges. As presented, some limitations remain for the more advanced systems, such as the sensitivity to localized damage, environmental factors, and high computational costs, to name a few. However, when combined with complementary SHM techniques, these studies affirm the viability of natural frequency-based methods for monitoring of structural integrity of suspension bridges. Moreover, these methods are typically non-invasive as they can function with minimal operational disruption during data acquisition, and the sensors can be simple and cost-efficient.

2.5 Wind Loading and Signal Generation

2.5.1 Introduction to Wind Loading on Suspension Bridges

Due to their flexible nature, wind loading is a critical consideration in suspension bridge design. Since these infrastructures are relatively light, they are particularly susceptible to significant deformations and aerodynamic effects discussed in Section 2.1.3. For this purpose, two wind loading analyses can be performed to evaluate the capacity of a bridge to resist wind loading: static and dynamic.

The first analysis is for static wind loading, representing wind as a steady horizontal force on the bridge. This method is used to determine if the structure's displacement from consistent wind conditions is reasonable to maintain general stability. If the bridge is unable to resist uniform lateral forces without excessive deformation, fatigue becomes a risk over time, leading to increased damage.

The second analysis is with dynamic loading, in which a fluctuating wind signal is applied to the structure, creating oscillations and vibrations. This type of analysis is more complex and leads to more dangerous responses from the bridge as it may cause aerodynamic behaviours such as resonance, buffeting, flutter, and galloping. These phenomena must be carefully considered and mitigated in the design process to ensure long-term safety for the bridge.

The first step in performing these analyses is generating the wind loads acting on the bridge's components. This involves understanding the pressure, lift, drag, and moment effects on the structure, which can be derived using Bernoulli's equation (2.21) [69].

$$P + \frac{1}{2}\rho V^2 + \rho g h = constant \tag{2.21}$$

Where P is the static fluid pressure acting on the surface (N), ρ is the fluid density (kg/m^3), V is the fluid velocity along the streamline (m/s), g is the gravitational acceleration (m/s^2), and h is the elevation (m). The dynamic pressure term (kinetic energy - second term) is essential for wind loading since it controls the pressure acting on the structure and integrating the pressure across a surface results in net forces and a moment, as presented in Figure 2.13 [5].

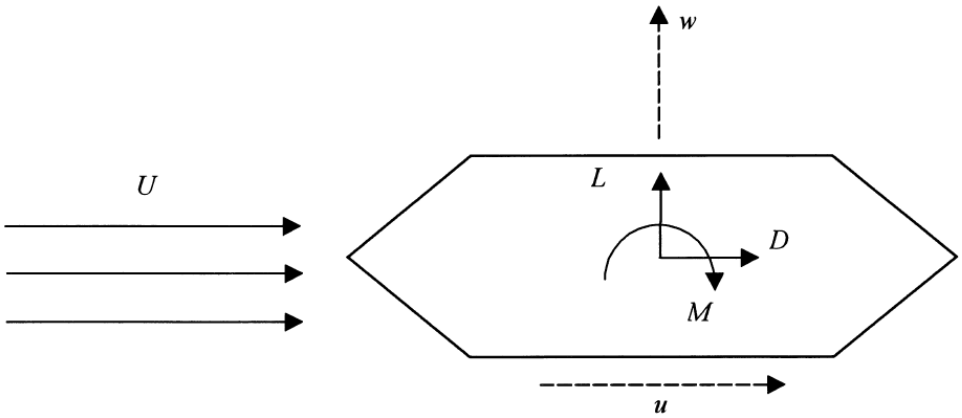


Figure 2.13: Wind Loads on a Body: Drag, Lift, and Moment [5]

To simplify the analysis, dimensionless coefficients are used to describe drag, lift and moment. These coefficients are based on the geometry of the object and the flow conditions to determine the load acting on the object. Moreover, these coefficients can be obtained in a wind tunnel experiment, usually done for the most critical components. For example, the deck of a bridge is typically tested in a tunnel since it is a critical part of a bridge design [5].

First, the drag coefficient is used to obtain the drag force on a body, which is the force along the direction of the wind flow. This drag force equation is presented as Equation 2.22 [70].

$$F_D = \frac{C_D \rho U^2 H}{2} \tag{2.22}$$

Where F_D is the linear drag force acting on the body (N/m), C_D is the drag coefficient, ρ is the air density (kg/m^3), U is the wind velocity (m/s), and H is the height of the object (m), perpendicular to the wind flow.

Second, the lift coefficient is related to the lift force, which acts perpendicular to the wind flow, as per Equation 2.23 [70].

$$F_L = \frac{C_L \rho U^2 B}{2} \tag{2.23}$$

Where F_L is the linear lift force acting on the body (N/m), C_L is the lift coefficient, and B is the width of the object (m) parallel to the wind flow.

Third, the moment coefficient is used in the moment determination, as per Equation 2.24 [70].

$$M = \frac{C_M \rho U^2 B^2}{2} \tag{2.24}$$

Where M is the linear moment acting on the body (Nm) and C_M is the moment coefficient.

These equations are critical to evaluate wind effects on different components of a structure, such as the deck, cables, and towers of a suspension bridge. However, depending on the type of analysis, some of the components in a suspension bridge have more impact when loaded. The deck experiences drag, lift, and moment forces, making it the primary focus of static and dynamic analysis. Under static conditions, drag forces dominate and must be controlled to avoid excessive displacements. In dynamic analyses, lift and moment become more significant since they cause some of the bridge's aerodynamic behaviour. On the other hand, the cables are subjected to drag forces and less to lift and moment due to their shape. Finally, the towers are also more influenced by drag loading, especially in static loading analyses.

2.5.2 Wind Velocity Signal Generation

Performing a dynamic wind loading analysis on a structure involves a realistic wind velocity signal that encapsulates the steady and fluctuating components of the wind. However, wind signals with high sampling frequencies are rarely readily available, making it difficult to simulate wind loading. To bypass this problem, this research proposes a wind signal generator script, which produces a wind signal with specific provided parameters. Details on this are available in Section 3.7, whereas this section introduces some of the key concepts used to create the generator. Subjects presented in this section highlight the importance of turbulence models, spectral analysis, and frequency-domain techniques in generating wind signals for such analyses.

Turbulence in wind signals is characterized by random fluctuations added to the mean wind velocity. To simulate said fluctuations, a turbulence spectra can be used to define the energy distribution across a range of frequencies. The Kaimal spectrum is commonly used to capture the observed characteristics of wind turbulence closer to the ground [71, 72, 73]. Low frequencies have the most significant energy in this spectrum compared to higher frequencies, a key characteristic of wind turbulence. The Kaimal spectrum is presented in Equation 2.25, which describes the variance of wind velocity fluctuations throughout the frequency range [72].

$$S(f) = \frac{4\sigma_u^2 L}{U}\left(1 + 6f\frac{L}{U}\right)^{-5/3} \tag{2.25}$$

Where $S(f)$ is the energy density $[(\mathrm{m/s})^2/\mathrm{Hz}]$ as a function of the frequency f (Hz), σ_u is the standard deviation of wind speed fluctuations (m/s), U is the mean wind velocity (m/s), and L is the integral length scale (m), which controls how quickly the energy decays with frequency (typically 200 m in open terrain). A random signal with a flat spectrum can be added to represent white noise typical of wind turbulence. This turbulence can be converted to a time-domain function

and added to a mean wind speed that varies randomly over time. Overall, combining a stable component with a turbulent component makes it possible to recreate a wind velocity signal.

The sampling frequency used for this process is controlled by the Nyquist frequency, which states that the sampling frequency must be at least twice the highest frequency of interest in a signal. This relation, presented in Equation 2.26 [74], ensures that undersampling is not an issue in the analysis. Undersampling, also known as aliasing, is an issue in which frequencies greater than $0.5f_s$ appear as lower frequencies, leading to distortions in the signal. Therefore, this condition should be met to prevent this phenomenon and preserve the signal's integrity [74].

$$f_i \leq 0.5f_s \tag{2.26}$$

Where f_i is the greatest frequency of interest (Hz) and f_s is the sampling frequency of the signal (Hz).

2.6 Genetic Algorithms

Genetic Algorithms (GAs) are a branch of optimization techniques that is inspired by the concept of natural selection and genetics. This evolutionary algorithm is a type of unsupervised machine learning, a subset of artificial intelligence. It is an effective method to solve complex, non-linear, and multi-modal optimization problems. The main process of the algorithm starts with an initial population of individuals that represents the solutions, each of them having an associated score based on their proximity to the target solution. These individuals have chromosomes, which represent the parameters that form an individual and that serve as inputs to the calculation process. The concept of the algorithm is that the chromosomes evolve within the population over multiple generations through the application of genetic selection, mating, and mutation [10]:

- Selection: Individuals are evaluated using a fitness function, which is a way to measure the accuracy of the results. This score can be interpreted as the inverse of the error of a solution.
- Mating: When creating new generations, selected individuals can "mate" to exchange portions of their chromosomes to promote diversity in the population, therefore exploring new regions of the solution space.
- Mutation: During the creation of the individuals for a new generation, some individuals may undergo a mutation, which is a small change in some chromosomes. This maintains some genetic diversity and prevents early convergence to a local optima.

Through different methods, the individuals of the starting population will be subjected to selection, mating, and mutation to form the individuals of the next generation. This process is repeated over multiple generations by allowing a selection of the best individuals to pass on their genes [10]. Over time and with adequate parameters, this should mimic natural selection and converge to stronger individuals (greater fitness score) that provide a better solution to a problem. Figure 2.14 [10] is a simple flowchart representation of this concept.

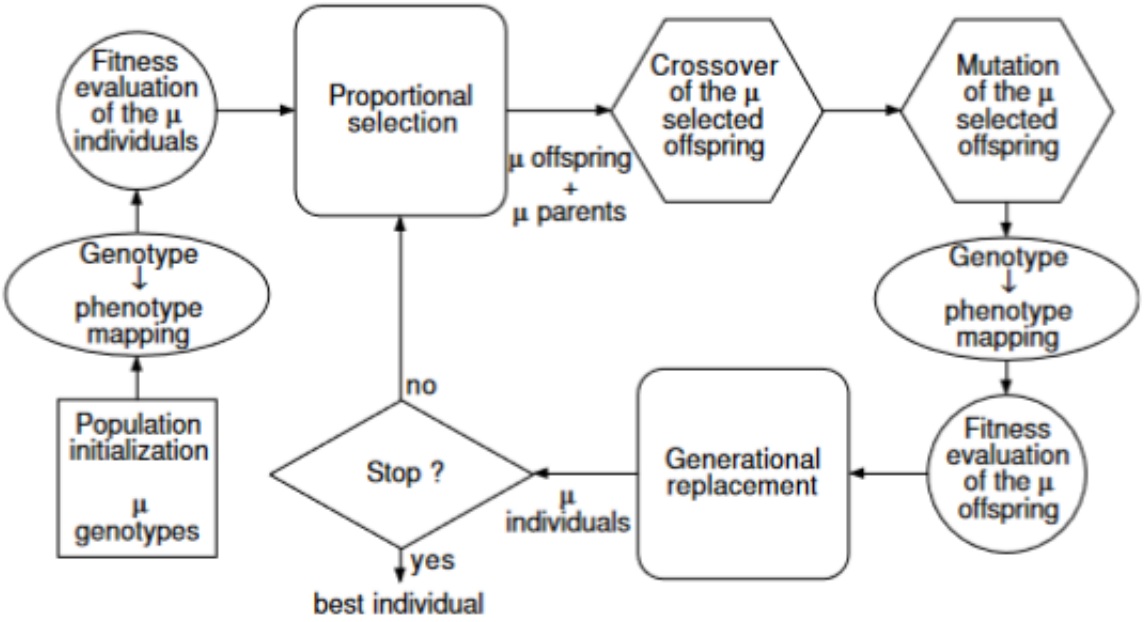


Figure 2.14: Simple Genetic Algorithm [10]

In the past decades, GAs have started to be applied in structural engineering with different objectives. For instance, Yan et al. [75] illustrate using GAs for vibration-based damage detection in structures by analyzing the dynamic response. Then, Greco et al. [76] demonstrate how GAs can be used to predict the seismic collapse of frame structures. In a similar branch, a different study developed a method to achieve an optimal plastic design of a frame structure using GAs to balance seismic safety and economic considerations [77]. These studies demonstrate how GAs are slowly incorporating the field of structural engineering while being applicable to diverse contexts.

In this study, genetic algorithm concepts are applied as a convergence tool for a complex structural system. Additionally, they can be utilized in future developments of this project to automate structural health monitoring through machine learning.

Chapter 3

LLRB Finite Element Modelling

3.1 Chapter Overview

This chapter covers the modelling process of the Lower Liard River Bridge (LLRB) as a fishbone model (FEM). Preliminary sections briefly discuss the rationale behind the selection of the software. Additionally, a short disclaimer regarding the structural drawings of the bridge is included to offer necessary insight into the bridge modelling process. A short general presentation of the LLRB structure is added to give context to the following modelling sections. From this point, the chapter is structured to exclude discussions on the scripting and automation aspects of the modelling process. Instead, explanations and justifications are presented as if the model was created using the Abaqus Graphical User Interface (GUI).

3.1.1 Software Selection

As previously stated, the finite element software Abaqus was used to develop the bridge model in this research. This software was selected over alternative options due to the available scripting capabilities, which allowed for the automation of the model generations and analysis through Python. Due to the nature of this project, this feature is required to complete this current research and future developments in this field using this program.

3.1.2 Structural Drawings Disclaimer

The design drawings and other technical details necessary for the modelling of the LLRB were obtained from the owner of the bridge. However, a non-disclosure agreement was signed and the owner's name will be omitted from this current and any further publications. No parts of the

structural drawings can be shared directly in the context of this research. Furthermore, the only version of the structural drawings made available for the pursuit of this project is missing key elements that have been lost from the poor scan quality of the original documents. Apart from the original 1943 drawings, no other physical or digital copy of the material was available. Therefore, a considerable part of the work not presented in this thesis has been identifying dimensions and properties from the drawings, often from assumptions or visual inspections available on the public domain.

3.1.3 Design Particularities of the LLRB

The Lower Liard River Bridge (Figure 3.1 [11]) is a suspension bridge located on the Alaska Highway in British Columbia that, as the name indicates, crosses the Liard River. The steel deck has a length of over 300 m, and the steel portion of the towers has a height of almost 28 m. A total of 23 hangers can be found along the bridge on each side, with 13 of them located in the bridge's main span. A small particularity relevant to the modelling of this bridge is that out of these 23 hangers, five are solid steel sections instead of typical wires. Three of them are located at the midspan of the main span, and the others are at the location closest to the anchor.



Figure 3.1: Aerial Picture of the Lower Liard River Bridge [11]

Furthermore, the main cables are separated into a bundle of twelve individual wires attached at each hanger location and at regular intervals between hangers. These form a 3x4 bundle pattern in which every wire has a notation presented in Figure 3.2. Most of the bridge's components are made of steel, apart from the deck surface, which is made of concrete. Finally, the towers are built atop

large concrete piers, whereas concrete abutments support the end of the decks. Further details related to the design of the LLRB are available in relevant sections throughout this chapter. As stated in the introduction, this thesis uses the term "wire" to designate bundles of "strands", where strands are the smaller individual elements. This terminology is used to avoid confusion with the use of the term "wire" in Abaqus. A similar distinction must be made for the term "beam", which references what is known as "frame" elements in structural engineering.

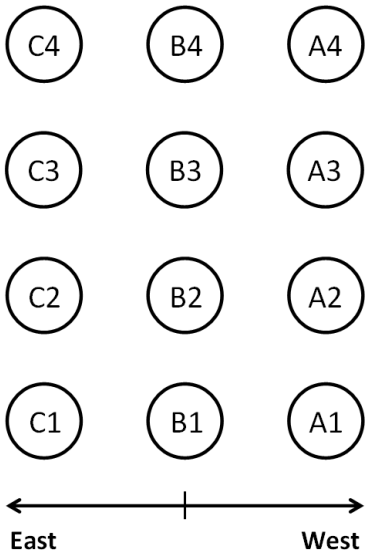


Figure 3.2: Main Cable Bundle Wire Notation

3.1.4 LLRB Bridge Model Terminology

In this chapter, references are made to the different bridge components. Therefore, a short explanation of the global nomenclature used is appropriate, which should facilitate reading the rest of the chapter. First, there are four groups of components, which are illustrated in Figure 3.3. In this figure, the groups are organized by colour: green deck, red towers, purple hangers, and blue main cables. Second, the coordinate system is available in the bottom left or top right corner of the figure. These axes remain consistent about the bridge orientation, such that the long bridge direction is the X axis, the cross-bridge direction is the Z axis, and the vertical direction is the Y axis. Third, the cardinal directions are identified in teal for the naming convention in the modelling. These directions are close to the actual bridge's, with a slight clockwise rotation in the top view. Specifically, the cardinal directions of the established coordinate system are defined as follows: north towards X positive, south towards X negative, east towards Z positive, and west towards Z negative. These naming conventions remain constant throughout this section.

As part of the general Abaqus terminology, the following definitions are established within the

context of this research:

- Model: The complete representation of the bridge in Abaqus, encompassing all components defined within the software.
- Parts: Independently defined components that typically correspond to physical components of the bridge. In this research, all parts are modelled as wires.
- Wires: One-dimensional parts embedded within a three-dimensional space, defined by nodes and elements.
- Elements: Finite Element Method (FEM) segments connecting nodes along a wire. In this research, elements are strictly one-dimensional.
- Nodes: Discrete points within the FEM mesh that define the geometry of parts and facilitate element connectivity.
- Assembly: The three-dimensional space where instances of parts are positioned, and where steps, constraints, boundary conditions, loads, and predefined fields are applied.
- Instances: Individual representations of parts within the assembly, which can be translated, rotated, or duplicated as needed.
- Sets: Collections of elements or nodes grouped together for the purpose of applying loads, boundary conditions, or for post-processing analysis.

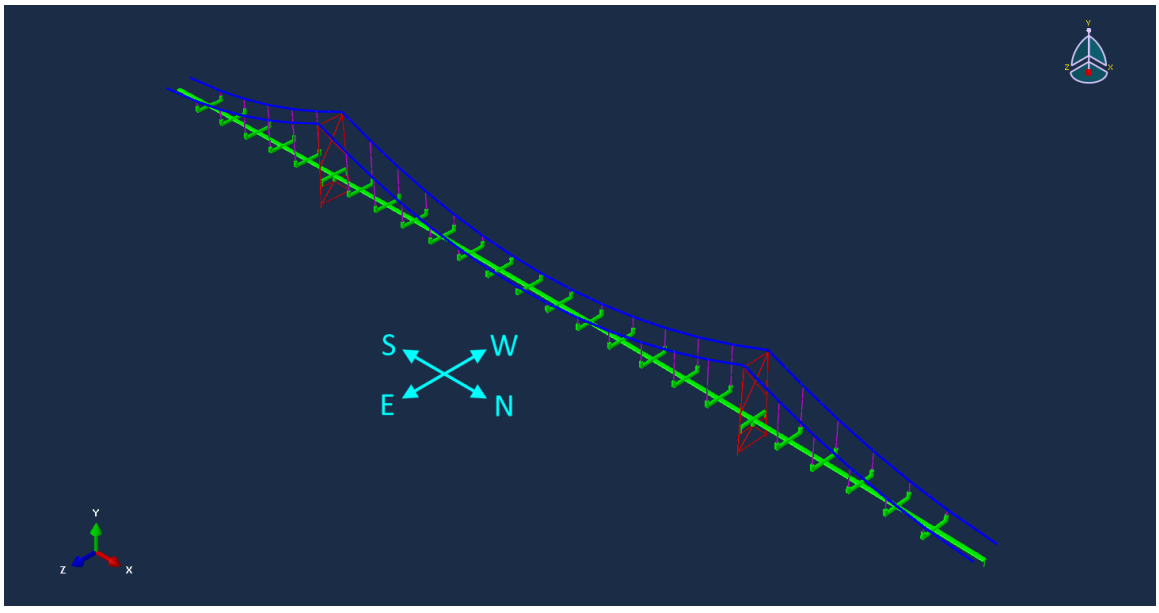


Figure 3.3: Nomenclature for the Complete Bridge Model

3.2 Bridge Parts Modelling

This section provides a comprehensive overview of the process used to model the various structural components of the LLRB through the creation of parts in Abaqus. In the context of this research, structural elements from the actual bridge are simplified into three-dimensional parts, modelled as deformable wires used in the model assembly construction. These parts serve as the fundamental elements forming the model, directly representing the physical components of the bridge. Except for the cable supports (Section 3.2.3), all the parts are unique in the assembly, meaning there is only a single instance of every part. This approach is unconventional in typical Abaqus usage, particularly given the double symmetry along two of the bridge's axes. In similar cases, many identical instances within the assembly could have been generated from a single part using translation, reducing the amount of parts required for the model. Many similar instances in the assembly originating from different parts could have been created with a single part through translation in the assembly. However, the approach of using duplicate parts is preferred for two main reasons. Firstly, it simplifies the automation process through Python scripting, as the parts are directly modelled with the correct coordinates, eliminating the need for additional manipulation within the assembly. Secondly, this approach enables different symmetric components to exhibit varying levels of damage. This advantage is particularly crucial for the damage modelling of the main cables, as they may exhibit asymmetric damage patterns both along and across the length of the bridge.

All parts described in this section are created using the same method: datum points (features of the parts) are defined and connected through a point-to-point wire feature. These parts are created using their final coordinates in the assembly, which eliminats the need for any translation or rotation once the instances are created. With the definition of the geometry of the part, some sets are also created to facilitate the modelling of connections, boundary conditions, and loading in the later stages of the model creation. "Sets are references to selected geometrical elements, specifically nodes in this case.

Once the geometry of a part is defined, the mesh is applied to the wire in three steps. First, the seeding of the edges is created by element size or number of elements. Most parts in this model are seeded by number of elements, except for the tower parts; more details are provided in Section 3.2.1. Second, the element type is defined as Truss (T3D2: 2-node linear 3-D truss) for parts representing cables or Beam (B31: 2-node linear beam in space) for other components. For these members, Figure 2.6 can be used as a visual representation of the available degrees of freedom. Third, the mesh is generated for the part. The mesh can also be defined through the

assembly instead of the parts. Defining the mesh through the assembly allows much better control over the interactions between parts, ensuring that contact points are meshed as nodes, for example. Also, an assembly mesh makes it possible to adjust transitions between parts in a more complex model. For this project, the mesh through parts is the better option, as it requires slightly less computational effort and is easily automated. The interactions between parts are not a concern, as the wire parts are defined with future connections in mind, ensuring that the members are already separated according to the required connections. This ensures a meshing node is located at all the required contact points. Since the model consists solely of one-dimensional elements, it does not require the same level of transitions as a solid part model. Therefore, the advantages of an assembly-level mesh do not apply, which justifies the use of a parts-level mesh.

Subsequently, the section is defined and applied to the relevant wires within the part. Parts modelled as Truss elements for meshing include a section with a specified material and cross-sectional area. On the other hand, the section Beam elements must first be defined through a profile. The profile is a cross-section definition, either as a set shape (box, pipe, rectangular, etc.) or as a generalized set of properties (Area, I_{xx}, I_{xy}, I_{yy}, and J). This model uses the latter as it simplifies the automation process and the standardized way of providing inputs. Then, the section can be defined through the profile and the material.

The last step of the parts modelling is the beam section orientation for Beam elements. This is simply to specify the orientation of the defined properties (I_{xx}, I_{xy}, and I_{yy}) in the section.

The following subsections present the different parts of the model under four main categories: towers, deck, hangers, and main cables. A breakdown of the component geometry, set definitions, mesh, and section assignments is provided for each collection of parts. For reference, the assembly's origin is at the southern end of the deck at the centre of the deck.

3.2.1 Towers Parts Modelling

The Lower Liard River Bridge has two towers, one on each side of the main span. The different tower parts can be separated into three categories: the vertical towers, the braces, and the cross-girder. These elements are represented in Figure 3.4 by colour: red for vertical towers, green for braces, and purple for the cross-girder. Each component has a specific role in the structure. The vertical towers are the prominent members and are connected to the brace members, the cross-girder, and the main cables. Their primary role is to transfer the load from the deck to the ground through the direct connection to the deck or the main cables. The cross-girder element supports the deck through the tower bearing (Section 3.2.2) and transfers the load to the vertical towers. Finally,

the braces have multiple purposes: increasing the lateral resistance of the structure, supporting the cross-girder at its midspan, and increasing the rigidity of the entire tower. This increased rigidity reduces the total deformation of the tower and, therefore, controls the maximum sag of the deck.

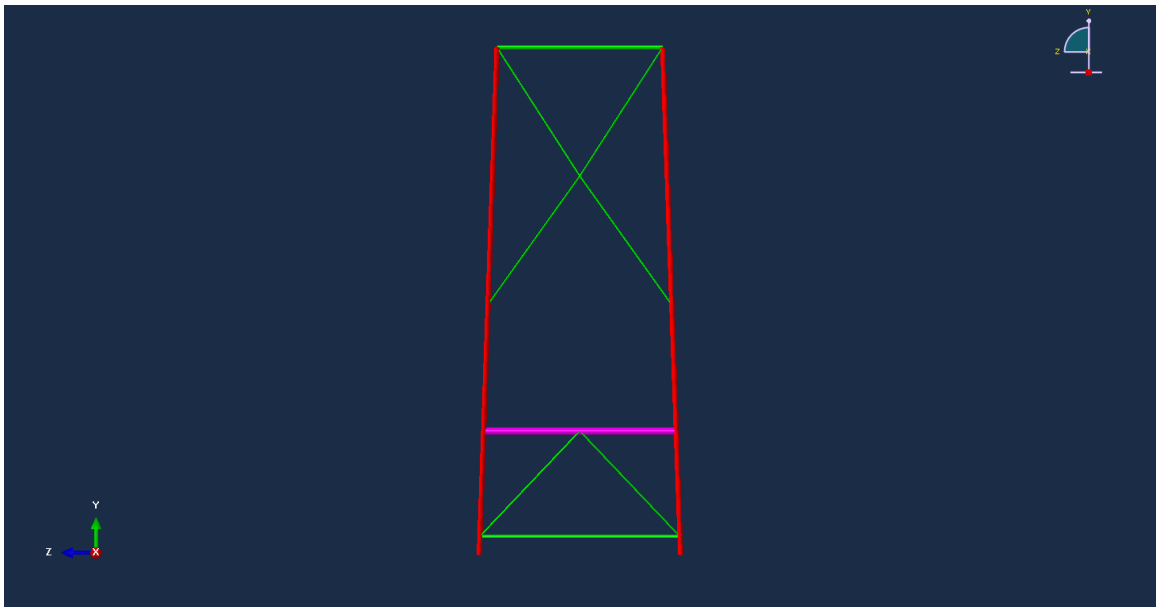


Figure 3.4: Nomenclature of the Main Components of the Tower

Due to their simple linear geometry, the tower components can easily be modelled as one-dimensional wires in 3D. However, many of the elements in the physical bridge are built-up steel sections, which means they are composed of multiple steel sections. It must be mentioned that all members are represented as one-dimensional beam elements in 3D. Therefore, the parts can be created individually by member based on their coordinates sequence. The coordinates used for the definition of the towers are approximated from the structural drawings, with some assumptions made to account for the one-dimensional simplification of the three-dimensional bridge components.

In essence, the wire parts are defined at the geometric center of the actual members, and the length of the adjacent members must be adjusted to ensure proper connection between them. The final drawing with dimensions used for creating the towers in the model is presented in Figure 3.5; coordinates of all members can be obtained from this drawing. Overall, 11 parts are defined per tower, for a total of 22 parts created for all towers. These parts and their name are presented in Table 3.1, with {E/W} representing either "E" or "W" for the east or west side of the bridge, {1/2} representing either "1" or "2" for the first (south) or second (north) tower, {Top/Bot} representing either "Top" or "Bot" bot top or bottom members of the cross brace, and {0/1} representing either "0" or "1" for the first (lower) or second (upper) horizontal brace.

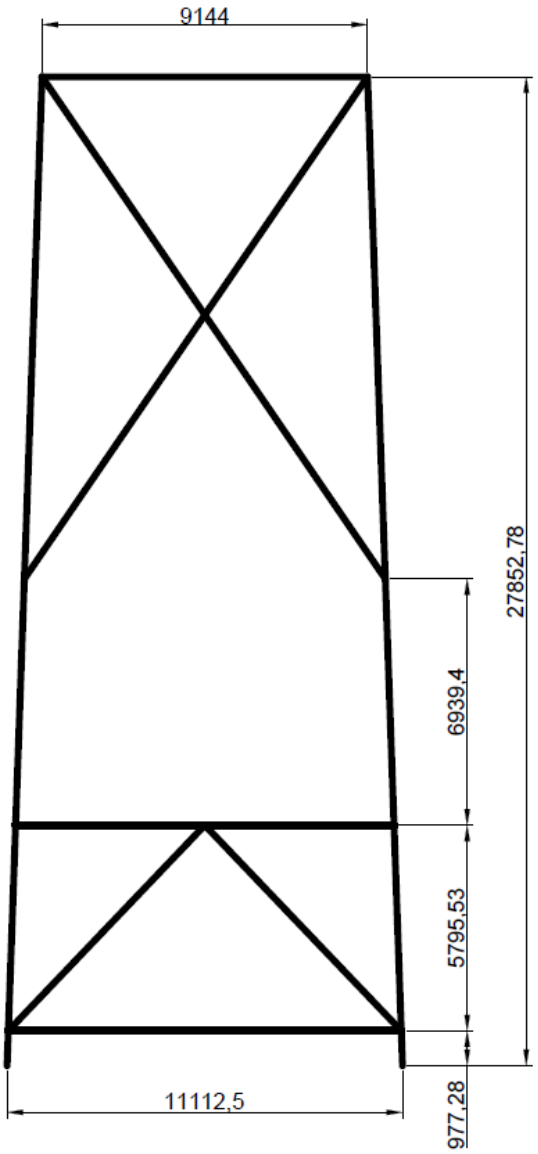


Figure 3.5: General Tower Dimensions

Table 3.1: Assigned Terminology for the LLRB Tower Parts

Type of Part	Part per Tower	Name
Vertical Tower	2	Tower[{E/W}][{1/2}]
Cross Brace	4	TowerBrace[Cross[0]:{Top/Bot}-{E/W}][{1/2}]
Chevron Braces	2	TowerBrace[Chevron[0]:{E/W}][{1/2}]
Horizontal Brace	2	TowerBrace[Horiz[{0/1}]][{1/2}]
Cross Girder	1	TowerCrossGirder[{1/2}]

The vertical towers can be identified as the outer members of the towers. Contrary to the name, these members are not perfectly vertical (along the Y axis) but have a slight angle (2.024°) visible in the YZ plane. Each vertical tower is created by defining a series of coordinates connected by a wire. The cross-girder then serves as the member connecting vertical towers to the deck bearing,

as defined in Section 3.2.2. The cross-girder is also supported at its mid-span by the chevron brace. While the cross-girder acts as a brace for the towers, the difference lies in the additional support to the deck. Four additional braces are distributed along the height of the tower, as presented in Figure 3.4. From bottom to top, the first brace is a horizontal single-member brace. The second is the chevron brace, starting at the ends of the first horizontal brace and ending at the midspan of the cross-girder. The third brace is the cross brace system, which begins approximately at the middle of the entire tower height and ends at the top of the vertical towers. The last brace is another horizontal brace at the top of the vertical towers, which also connects to the top of the cross brace.

A general naming system is used throughout the model to create the sets within the parts. The first term to be introduced is set_alledges, which is defined for all parts in the model and represents the entirety of the part (the wire element). Then, many parts have set_beg and set_end, denoting the first and the last node in a single one-dimensional element (wire). Naturally, first and last are insignificant terms for an element in a finite element model; they will be defined per part when relevant to the naming convention. For cross brace elements (four per tower), these three sets (set_alledges, set_beg and set_end) are defined. The external nodes (lower node for bottom elements and upper node for top elements) are identified as set_end, and the centre of the brace has all the last nodes (set_end). The chevron elements (two per tower) follow the same rule as the cross braces, with the bottom nodes set to set_beg and the connection between both elements as set_end for each part. The horizontal brace elements have the previously mentioned three sets, with the addition of a set_ctr, being the node at the centre of the part. For this type of brace, set_beg is on the west side (Z negative), and set_end is on the east side (Z positive). The cross-girder elements are similar to the horizontal braces but must also have a set with the purpose of connecting to the tower bearing, establishing the connection between the towers and the deck. These sets are named set_beg_mid and set_end_mid and are 9.144 m apart, centred at the set_ctr. Figure 3.6 demonstrates the layout of the sets in the cross girder.

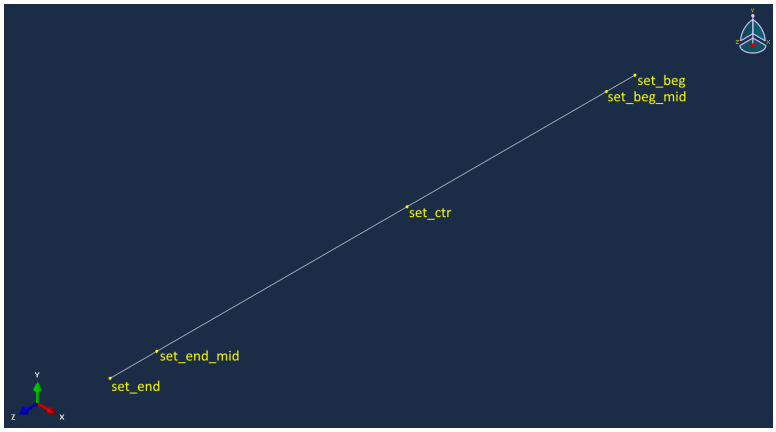


Figure 3.6: Cross Girder Sets Naming

While some parts use set_beg and set_end, others use set_bot and set_top to denote bottom and top nodes. There is no technical difference between the two naming conventions, but the bot/top version is preferred for vertical members. Therefore, the vertical tower parts use this convention, along with multiple set_mid{i}, with i varying from 0 to 2: set_mid0 is at the intersection with the lower horizontal brace and the bottom of the chevron brace, set_mid1 is at the intersection with the cross girder, and set_mid2 is at the intersection with the bottom of the cross brace. Figure 3.7 presents the described sets for a single vertical tower.

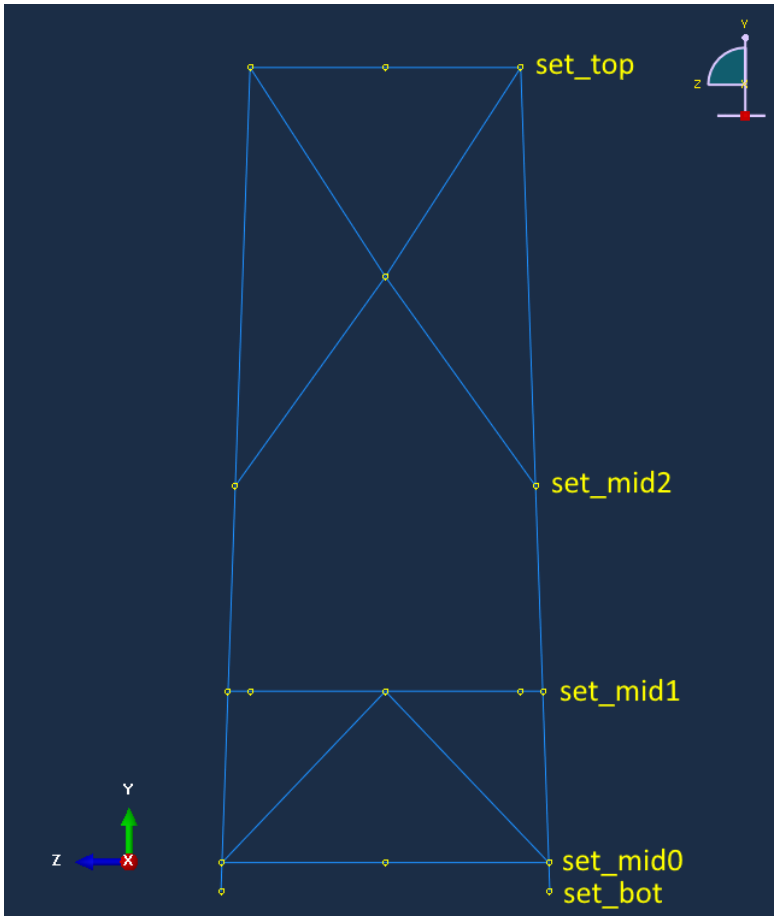


Figure 3.7: Vertical Tower Sets Naming

As mentioned previously, the meshing of the parts is done through the parts module and not the assembly. All tower parts are seeded by edges using the size method with the set set_alledges. The approximate element size used is 0.75 m, with a curvature control maximum deviation factor of 0.1 and a minimum size factor of 0.1. Larger mesh elements in the towers cause instabilities in the model, while smaller mesh elements do not significantly improve the precision of the results. The curvature control for the linear wire part uses default values and does not affect the seeding with this geometry. For set_alledges, the element types are assigned as B31 elements, which are beam elements from the standard element library that use Timoshenko beam theory instead of Bernoulli's. The beam type is shear-flexible, with a linear bulk viscosity of 1, the default value. Finally, the mesh is generated for the parts, resulting in the node pattern in Figure 3.8, with 38 nodes in vertical members and 15 nodes in horizontal members.

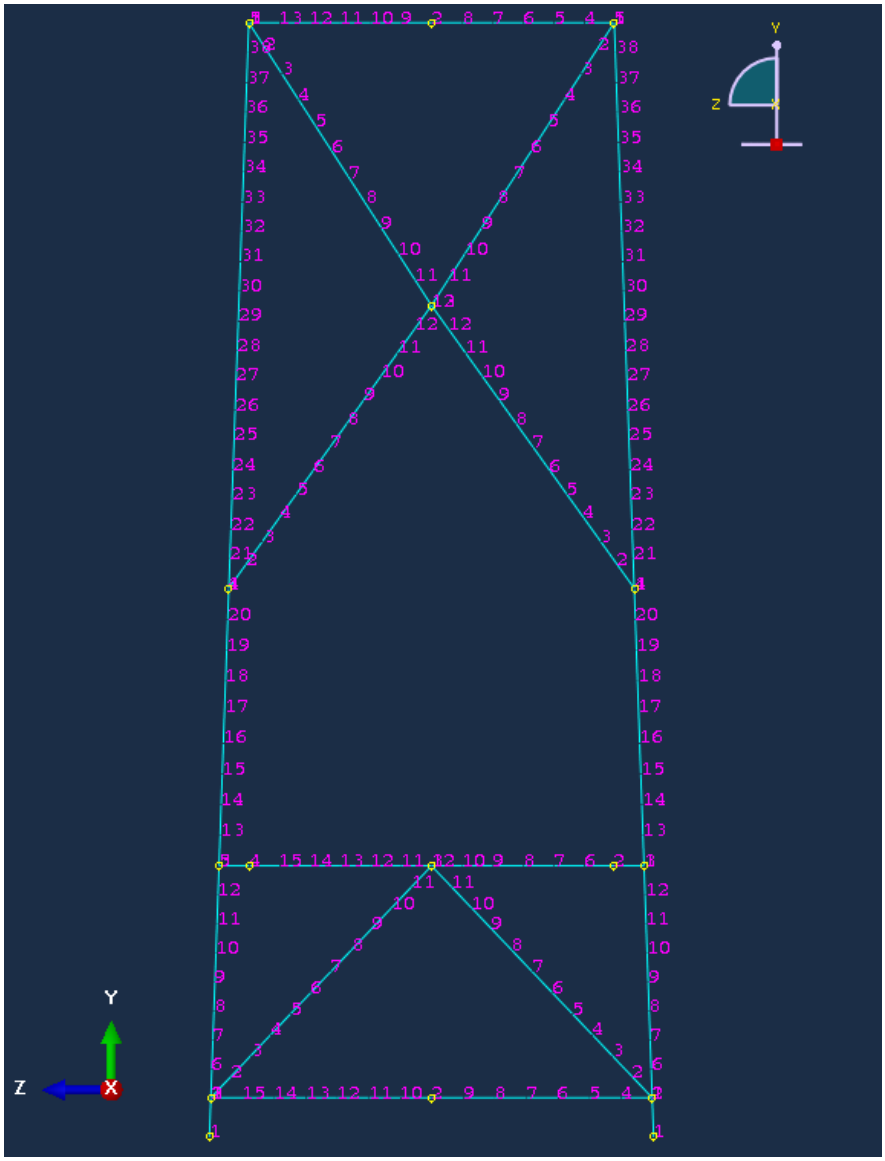


Figure 3.8: Tower Mesh

The next step in defining parts is the section assignment, which requires preliminary steps, including creating a material, profile, and section compatible with the element type (B31). The material for the structural steel (named "Structural Steel" in the model) is defined with a density of 7,850 kg/m^3, a Young Modulus of 200 GPa, and a Poisson's Ratio of 0.3 [78]. These values are selected as default steel properties since no details are available on the actual properties of the bridge.

Due to the complex geometry of the built-up steel beams, the generalized profile type of the beam must be used in conjunction with the estimation of profile properties for each member. The goal is to simplify an arrangement of steel beams (3D elements in a 3D space) to a wire beam (1D element in a 3D space). The structural drawings of the LLRB were obtained from the owner;

however, due to the pour scan quality of the drawings, the properties of most tower elements are missing. Therefore, most details are obtained by matching visual inspections from a 2011 field report [9] and from Google Maps Street View [79] to known dimensions in the drawings and cross-checking all available steel sections from the steel handbook [13]. Naturally, this method can produce approximations of the actual geometry, but it is anticipated to yield values with minimal error. Since the confidentiality agreement prohibits the sharing of any part of the structural drawings, this lengthy process is not described or presented in this thesis. Results from this investigation are presented below for profiles used for the tower parts. A table of the properties of sections used in the steel handbook [13] is presented in Table 3.2 below, and a table of the built-up beam layout is provided in Table 3.3. In these tables, A is the area, I_x is the second moment of area about the X-X axis, I_y is the second moment of area about the Y-Y axis, and J is the torsional constant.

Table 3.2: Individual Steel Beam Properties [13]

Sections	A (mm^2)	I_x ($10^6 mm^4$)	I_y ($10^6 mm^4$)	J ($10^6 mm^4$)
W530x165	21,100	1,110	114	2.83
W530x196	25,000	1,340	139	4.7
C380x74	9,480	168	4.6	1.1
L102x76x9.5	1,600	1.67	0.8	0.0487

The brace elements in the cross and chevron braces are not built-up steel sections but single steel beams identified as W530x165, while the horizontal brace elements are a built-up section consisting of two channels and two plates arranged in a rectangular configuration. The channels are C380x74 sections, whereas the plates are 20 in (0.5080 m) in length and 11/16 in (0.0175 m) in thickness; the layout can be observed in Figure 3.10 [12], and a picture from the 2011 visual inspection is provided in Figure 3.9 [9] in which the saddle is also included.



Figure 3.9: Horizontal Brace From the 2011 Visual Inspection [9]

The cross girder member is a complex built-up steel beam composed of two horizontal plates, four vertical plates and eight angles. Two vertical plates and two horizontal plates form a rectangular shell for the beam, which is then reinforced with the other two vertical plates as additional vertical members. Since the system uses only plates, as presented in Figure 3.11 [12], the connections are made possible by the angles at each corner of contact. The horizontal plates are 42 in (1.0668 m) in length and 3/4 in (0.0191 m) in thickness, while the vertical plates are 36 in (0.9144 m) in length and 3/8 in (0.0095 m) in thickness. The angles are all L102x76x9.5, with the longer side always in contact with the vertical plates.

The vertical tower members have a constant cross-section throughout the height of the towers, considerably simplifying the modelling process. The section is a built-up beam made from three W530x196 and two 17 in (0.4318 m) by 1 in (0.0254 m) plates, as presented in Figure 3.12 [12]. The layout is a single W530x196 with its flanges reinforced by two more of the same beam rotated 90°, such that the exterior part of the flange is in contact with the web of the rotated beams. Then, the web of the first W shape is reinforced with the two plates, one on each side.

To define profiles in Abaqus, the properties of an entire profile must be provided (Area, I_{xx}, I_{xy}, I_{yy}, and J). Therefore, the properties of the described built-up steel beams must be obtained. This

is mainly done using CSiBridge's Section Designer Module [12] for simplicity, but it has also been verified through manual calculations. As a result, Figure 3.10 presents the horizontal brace profile, Figure 3.11 details the cross girder profile, and Figure 3.12 displays the vertical tower profile [12]. Table 3.3 presents the summary properties for these profiles to be used in Abaqus. It is important to note that the I_{xy} properties of all profiles remain zero due to their symmetry. Overall, these profiles are directly defined in the model as TowerBraceProfile[Chevron], TowerBraceProfile[Cross], TowerBraceProfile[Horiz], CrossGirderProfile, and TowerProfile (for the vertical tower).

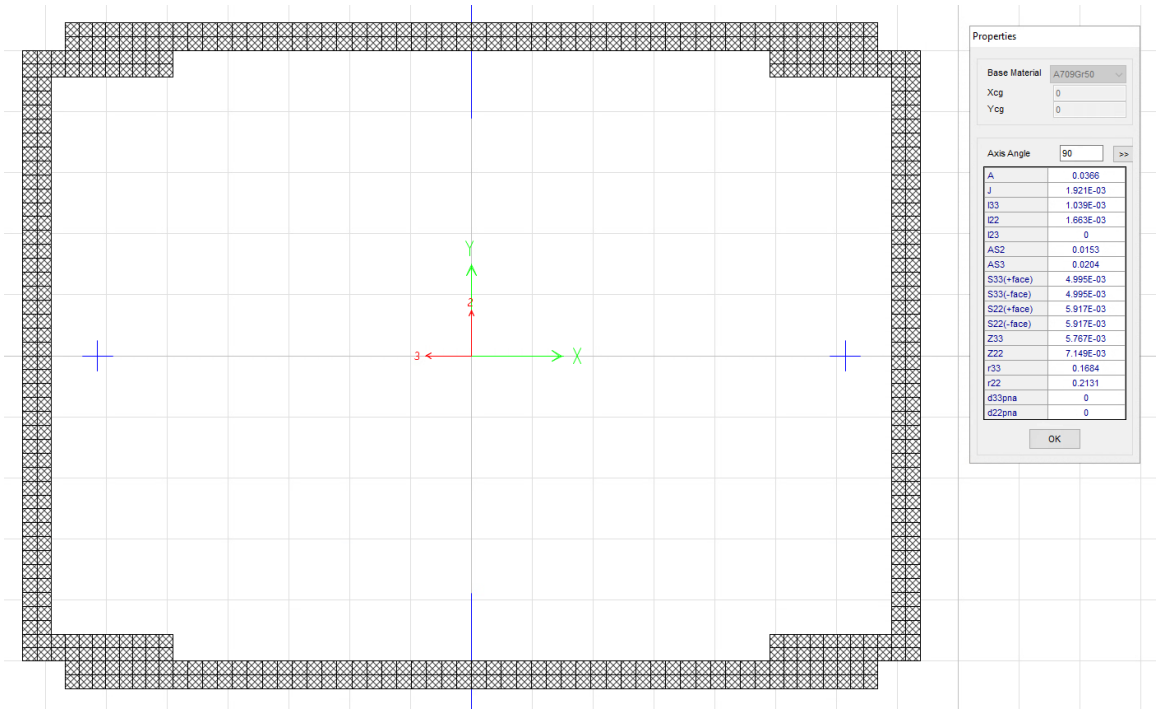


Figure 3.10: Horizontal Brace CSiBridge Section Designer [12]

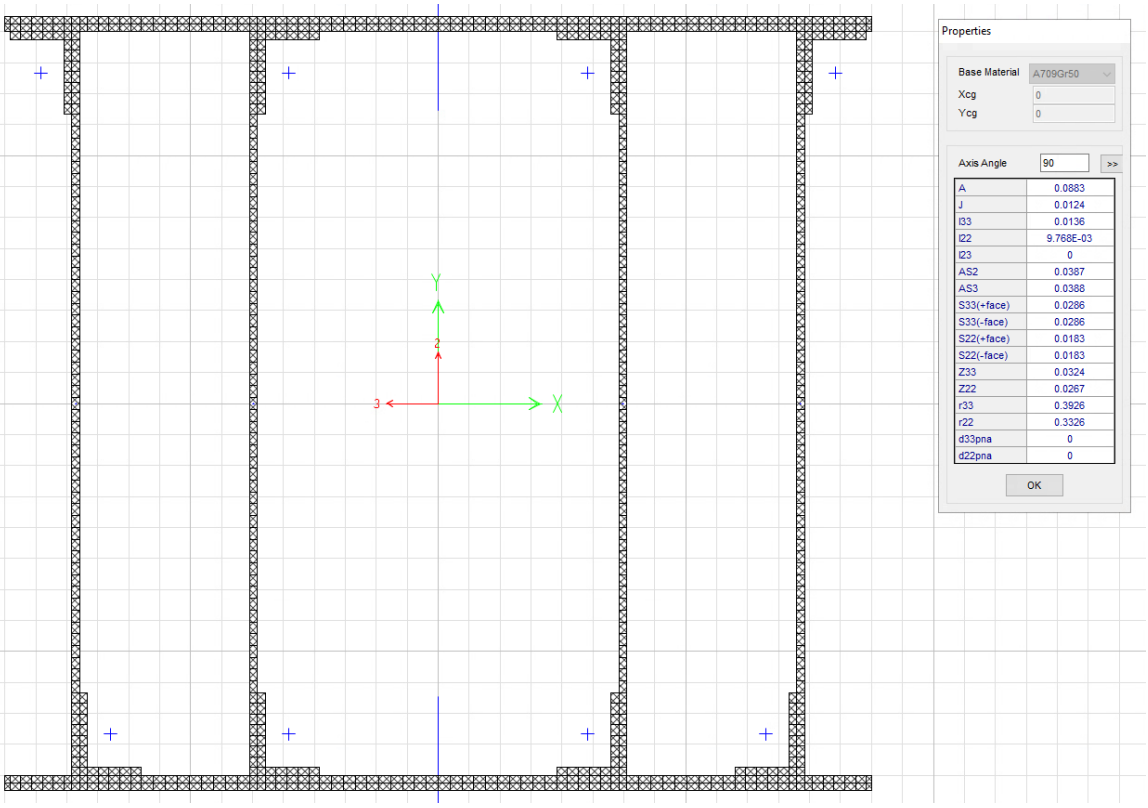


Figure 3.11: Cross Girder CSiBridge Section Designer [12]

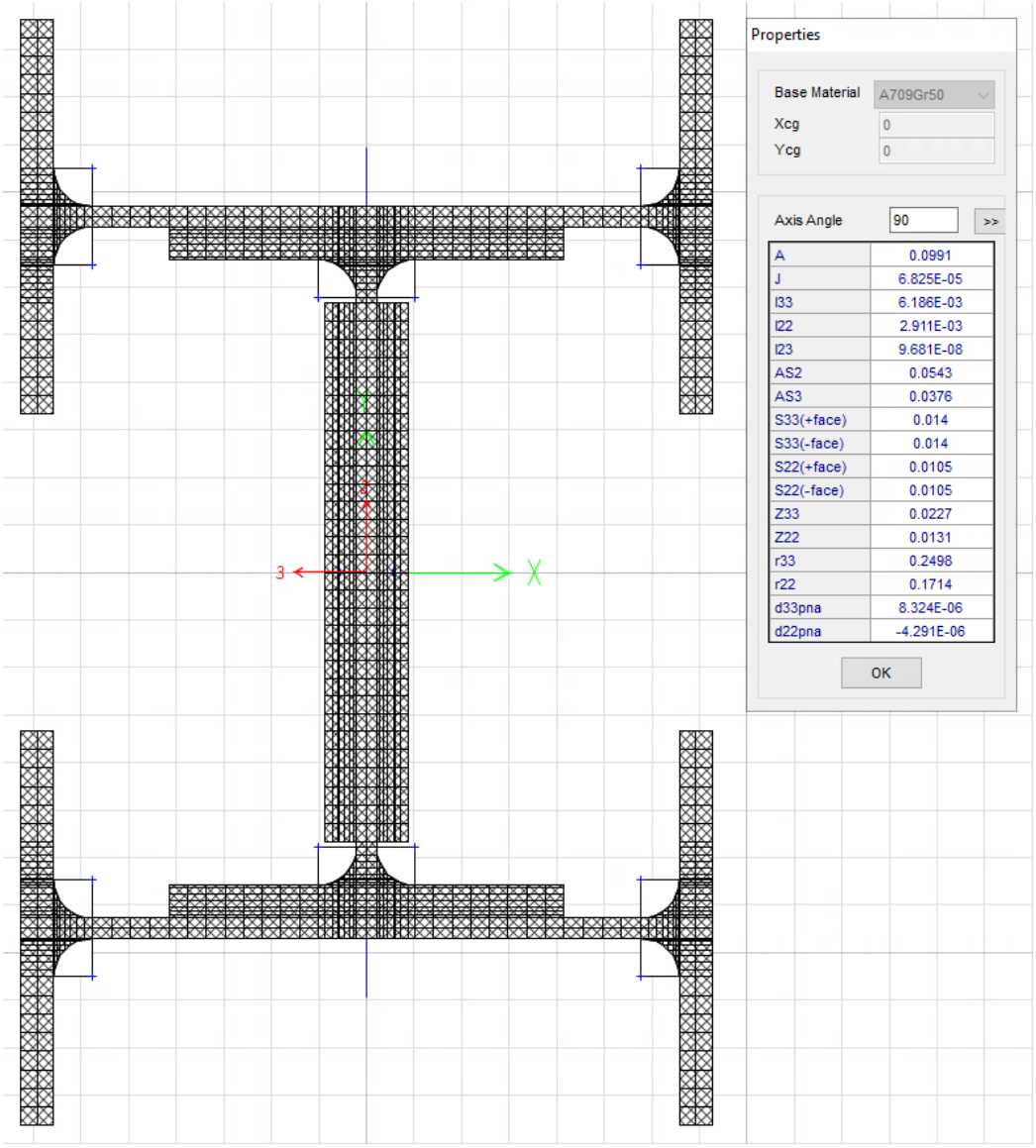


Figure 3.12: Vertical Tower CSiBridge Section Designer [12]

Table 3.3: Tower Profiles Final Properties

Profiles	A (mm^2)	I_x ($10^6 mm^4$)	I_y ($10^6 mm^4$)	J ($10^6 mm^4$)
Cross Brace	21,100	1,110	114.0	2.830
Chevron Brace	21,100	1,110	114.0	2.830
Horizontal Brace	36,670	1,041	1,679	1,921
Cross Girder	88,300	13,600	9,768	12,400
Vertical Tower	96,935	6,027	2,830	68.25

For profiles defined with the Generalized option in Abaqus (final property input instead of shape definition), the section must be defined as a beam with a section integration before analysis instead of during analysis. This is because the properties are provided as constants throughout the analysis and should not be reevaluated during the analysis. It also means that the computational

cost is reduced. General structural steel properties are used as material properties due to the lack of additional information. The defined sections are named TowerBraceSection[Chevron], TowerBraceSection[Cross], TowerBraceSection[Horiz], CrossGirderSection, and TowerSection, which are associated with the corresponding profiles.

Finally, the beam orientation is defined so that the orientation of the defined profiles aligns with the actual bridge.

3.2.2 Deck Parts Modelling

For fishbone models of suspension bridges with truss-style decks, the deck is considerably more complicated to simplify than the towers. Modelling a box girder-style deck is much simpler, as the section of the deck is continuous along the length of the bridge. Once a single cross-section of the box girder deck is defined and properties are extracted, these can be used directly on model members representing the deck. This is because the box girder style can be considered a constant linear member in space, which is easily simplified to a one-dimensional element. On the other hand, a truss-style deck presents a supplementary challenge for fisbone models, because it does not have a constant section along the length of the bridge. Depending on the location along the length of the deck, the layout of the steel members in the cross-section is not the same due to the truss pattern and the girders crossing the bridge deck at regular intervals. Therefore, the area, the second moments of area and the torsional constant must be approximated through a separate analysis of the deck, which is presented later in this section. For the general geometry of a fishbone bridge model, the deck is modelled as a continuous beam along the length of the bridge and girders to connect to the hangers. This structure gives the model a fishbone-like appearance, with the deck acting as the "spine" and the girders as "ribs". For this model, the properties of the actual deck are reflected in the deck parts, while the girders are modelled as rigid elements used only to connect the hangers to the deck while maintaining the three-dimensional aspect of the bridge.

The LLRB has a total of 26 segments between anchors, hangers and towers: six from the first anchor to the first tower, 14 from the first tower to the second tower, and six from the second tower to the second anchor. These segments are referred to as spans throughout this research. Figure 3.13 illustrates the general appearance of the deck, with the towers depicted to indicate the boundaries between the main span and the two side spans. The red members represent the deck elements, the green members denote the common girders, the teal members correspond to the girders at the towers, the yellow elements indicate the anchor bearings, and the purple members represent the tower bearings. The hangers, located at regular intervals, support the deck elements through the

girders and the towers through the girders at the towers and the tower bearings elements. Both ends of the deck are connected to anchor bearings, establishing the boundary condition to the ground.

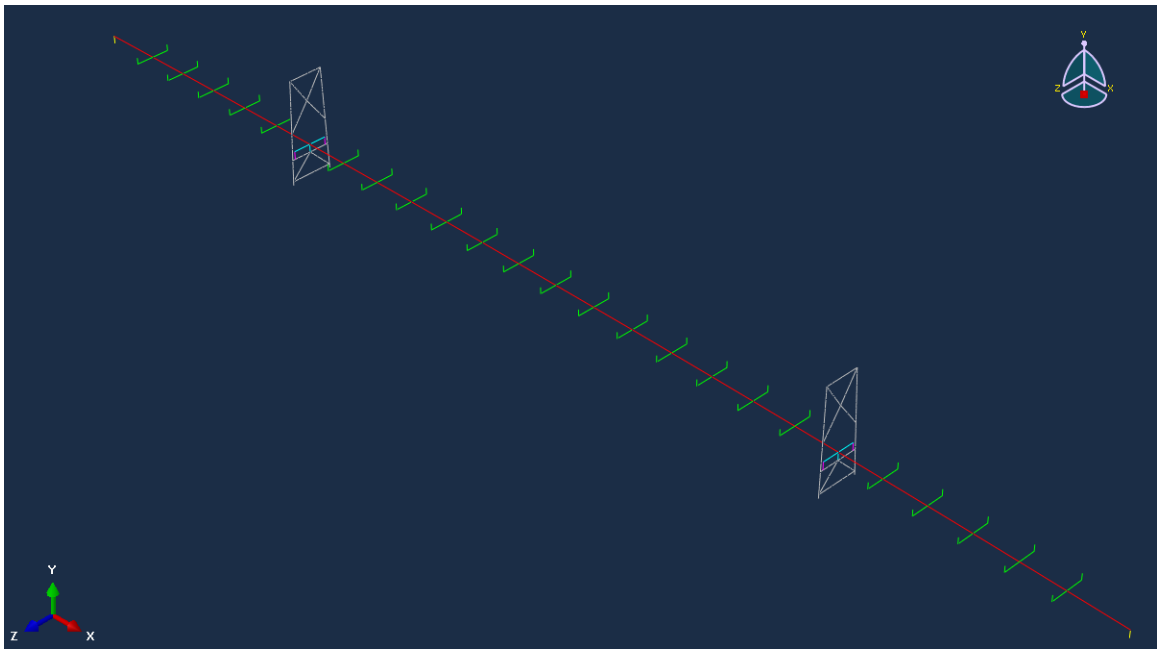


Figure 3.13: Deck Fishbone Model of the LLRB

There are 26 deck members, 23 common girders, two girders at towers, four tower bearings and two anchor bearings. All parts related to the deck follow the convention presented in Table 3.4, with {E/W} representing either "E" or "W" for the east or west side of the bridge, {i} representing the deck location index starting at 0 at the first anchor and increases by 1 at every end of deck span (up to 26), with index 6 and 20 being replaced by "T1" and "T2".

Table 3.4: Assigned Terminology for the LLRB Deck Parts

Type of Part	Name
Deck	Deck[{i}/{i+1}]
Girder	Girder[{i}]
Tower Bearing	TowerBearing[{E/W}][{i}]
Anchor Bearing	AnchorBearing[{i}]

In the LLRB model, all common girders have identical geometries, with a length of 9.144 m and exterior vertical arms of 1.657 m, with the centre being at the origin of the Z axis. The arms are created only to maintain the correct geometry of the deck by accounting for the deck's thickness from the section's centroid to the bottom of the hanger. While the length is obtained directly from the drawings as the distance between hanger pairs across the deck, the 1.657 m dimension was

obtained through the solid model section presented later in this chapter. All parts of the girders are parallel to the YZ plane, with the main length parallel to the Z axis and the exterior arms parallel to the Y axis. The first sets for this part are at the top of the arms with set_beg and set_end as Z negative and positive, respectively. These sets are used to connect the bottom of the hangers with the deck. Thereafter, the centre of the main length of the girders is identified with set_ctr, which is used to connect with the deck elements. A visual representation of these can be observed in Figure 3.14.

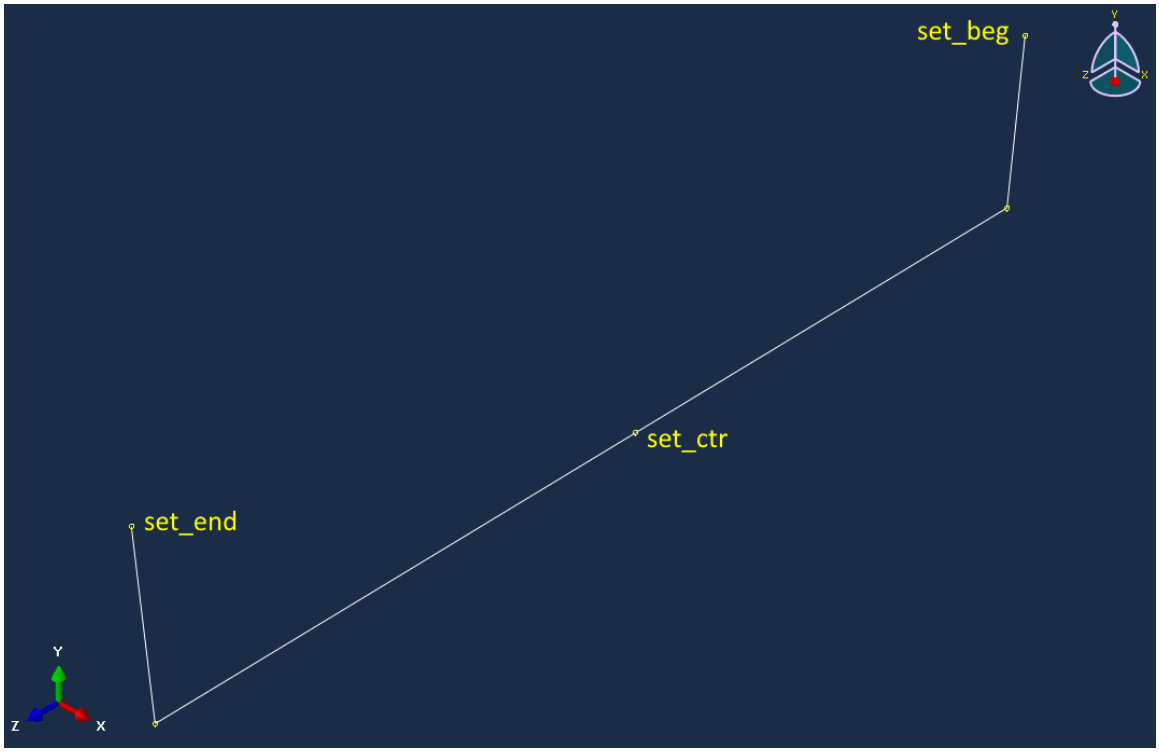


Figure 3.14: Assigned Terminology for the Girder Sets

The girders at the tower are slightly different, as they do not require vertical arms to attach to the hangers. Furthermore, a vertical wire component is added below the centre to reach the cross girder from the tower as part of the bearing system. This part has a length of 2.004 m, which was obtained through the elevation difference between the tower's and deck's known components coordinates. The set_beg and set_end are moved to the ends of the main wire, and the bottom of the vertical component is named set_ctr_bot. These sets are presented in Figure 3.15.

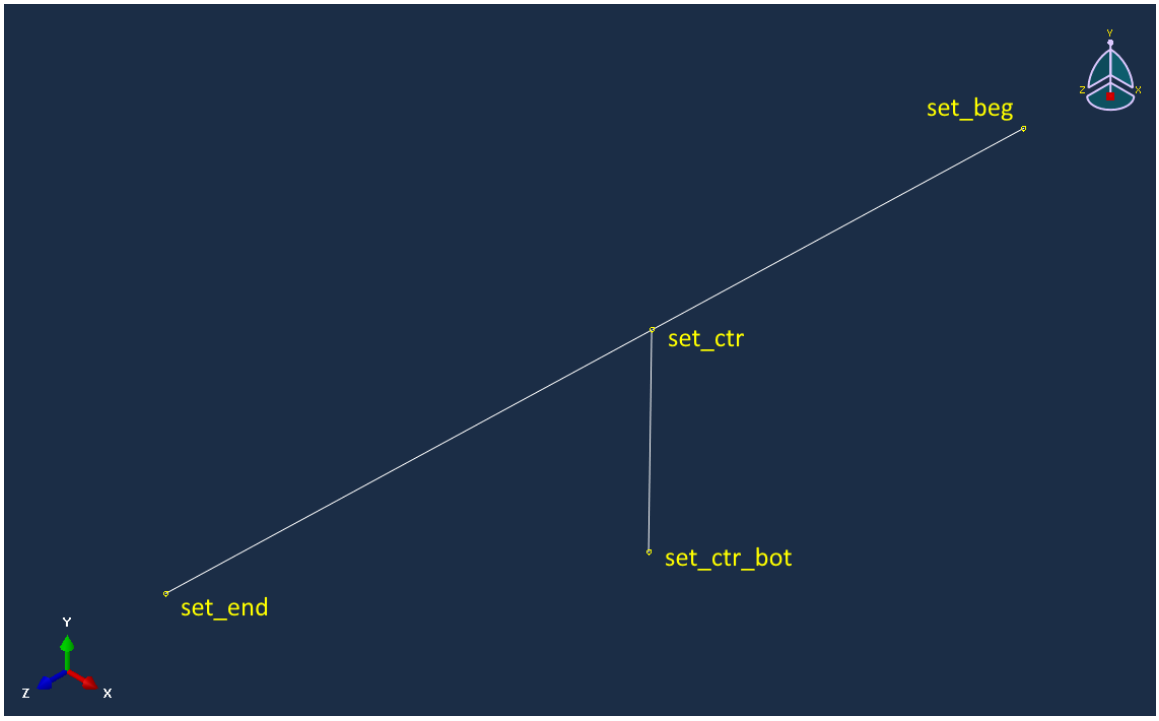


Figure 3.15: Assigned Terminology for the Girder at Tower Sets

As for the deck elements, they are created continuously to form a curve matching the deck's actual geometric configuration. The structural drawings do not directly display the deck's dimensions, making modelling challenging. Most values used for the deck coordinates are obtained as a deduction from available surveying measurements and hanger lengths. While these values are approximate, it is considered that they represent the real bridge well because the accuracy falls within the range of other assumptions made throughout the modelling process. The XY plane view of the LLRB is presented in Figure 3.16, and the coordinates of the start and end of the deck elements are provided in Table 3.5. Members are created from pairs of consecutive location index coordinates for a total of 26 elements. The length in the X direction is typically 11.786 m, with exceptions near the towers, where it is 12.014 m and at the anchors, where it is 15.399 m. The Y coordinates are set such that the top of the girder's arms are at a height of 0 m at the anchor if a girder was present at that location. This means that the first location index has a Y coordinate of -1.657 m, the length of the girders' arms. The maximum difference in elevation in the deck is 2.389 m, which is in line with the technical drawings of the LLRB. Every deck part has two edge sets: set_beg and set_end, with set_beg as the lower X end.

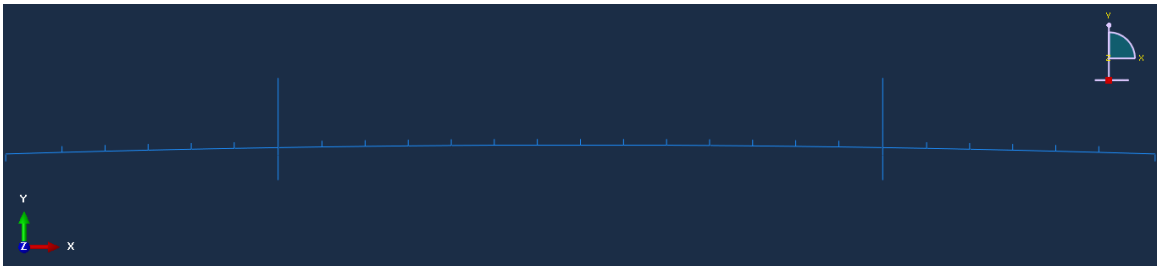


Figure 3.16: Deck Lateral View of the LLRB

Table 3.5: Start and End Coordinates of Deck Elements

Location Index	X (m)	Y (m)
0	0.000	-1.657
1	15.399	-1.211
2	27.185	-0.900
3	38.970	-0.615
4	50.756	-0.356
5	62.541	-0.123
T1	74.556	0.088
7	86.570	0.271
8	98.355	0.425
9	110.141	0.553
10	121.927	0.655
11	133.712	0.731
12	145.498	0.732
13	157.283	0.732
14	169.069	0.732
15	180.855	0.731
16	192.640	0.655
17	204.426	0.553
18	216.211	0.425
19	227.997	0.271
T2	240.011	0.088
21	252.025	-0.123
22	263.811	-0.356
23	275.597	-0.615
24	287.382	-0.900
25	299.168	-1.211
26	314.567	-1.657

For the bearing members, there is only a vertical component to consider. For the anchors, this component is set as 2 m, an arbitrary value due to the lack of details in the structural drawings. The bearings at the towers have a height of 2.004 m, the same as the tower bearings' height. The sets defined for these parts are set_bot and set_top, which are self-explanatory.

The meshing of deck parts is accomplished through the parts module and all parts also have a set named set_alledges that contains all the elements within a part. The parts forming the deck are seeded by edges by number, such that for each linear segment in the parts, there is a

single mesh element with a node at each end. This method of seeding is preferred since the stress complexity is much lower in the deck, which reduces the computational cost of the LLRB model. For example, trying to run the same bridge model with a single mesh element per segment in the towers creates a very unstable model that cannot arrive at a solution. Through set_alledges, the element type is assigned to all deck elements as B31 elements, described in Section 3.2.1 for the towers. The mesh is then generated for each part.

For the definition of the sections, the same material as the towers is used for the deck components: structural steel. An exception to this is for the girder parts, for which more details are provided later. The profile values are determined through a complete solid model of the deck due to the complexity of the truss system. Therefore, a 3D solid model of a typical 11.786 m long segment of the deck is created in Abaqus, independent from the fishbone bridge model. While some deck segments present slightly different steel members, their global properties are generally very similar. Thus, a typical deck segment is selected from the structural drawings and modelled with solid and shell parts. While the technical details for the deck parts used may not be provided as part of the confidentiality agreement, the resulting assembly created in Abaqus is presented in Figure 3.17, which replicates an actual deck segment of the LLRB with all dimensions and properties.

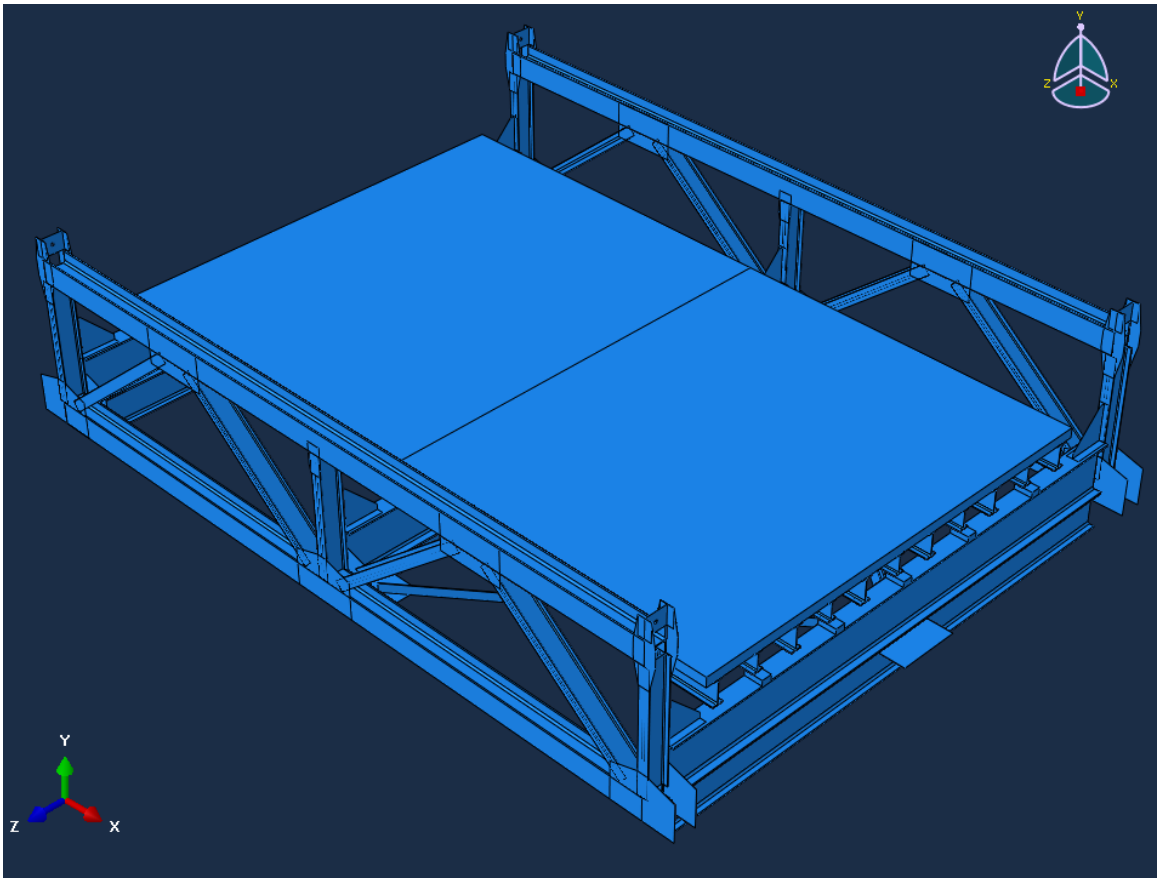


Figure 3.17: LLRB Deck Segment Solid Model

To determine the deck properties, one end of the deck is restricted with encastre (fixed for all DOFs) boundary conditions applied to the end of the stringers, the end of the concrete deck and the end of the top and bottom chords of the trusses. The other end of the deck is loaded with three successive scenarios: a vertical loading of 10 kN, a lateral loading of 10 kN and a torsional loading of 100 kNm. Due to the elastic nature of the materials defined, the magnitude of these values do not affect the final properties of the deck. The displacement or rotation in the appropriate dimension is obtained for each loading, and the deck properties are calculated with the beam deflection for a cantilever beam with a concentrated end force in Equation 3.1 [80] and the torsional constant formulation in Equation 3.2 [81].

$$I = \frac{PL^3}{3E\delta_{max}} \tag{3.1}$$

Where I is the second moment of area (m^4), E is the modulus of elasticity (Pa), δ_{max} is the maximum displacement in the axis of the applied force, P is the point force applied at the end of the beam (N), and L is the length of the beam (m).

$$J = \frac{TL}{G\theta_{max}} \tag{3.2}$$

Where J is the torsional constant (m^4), T is the torque applied at the end of the beam (Nm), L is the length of the beam (m), G is the shear modulus (Pa), and θ_{max} is the maximum rotation in the axis of the applied torque. The shear modulus can be obtained through Equation 3.3 below.

$$G = \frac{E}{2\,(1+v)} \tag{3.3}$$

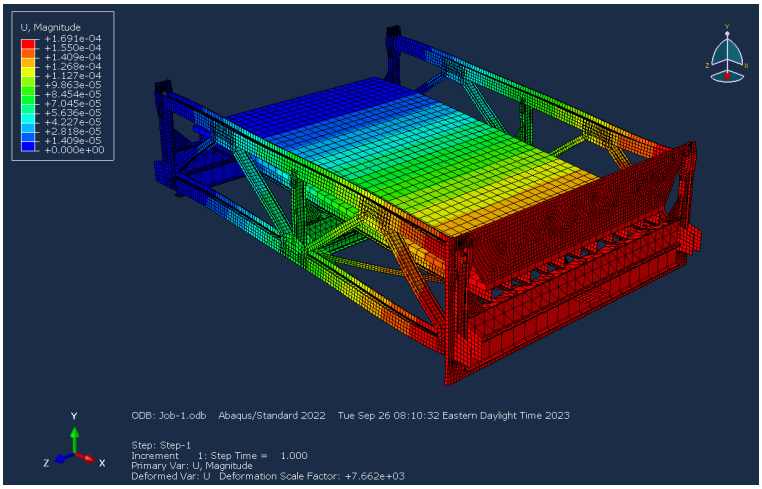
Where v is the Poisson's ratio.

A rigid loading plate is created to apply the loads uniformly on the deck end to equally engage all components in the deck. This plate has a very high modulus of elasticity (10^9 times greater than steel) to ensure that there can be virtually no deformations. Thereafter, tie constraints are applied at all contact points with the deck components to ensure loading continuity throughout the parts. Depending on the tested scenario, the plate has boundary conditions that allow it to move only vertically, laterally, or torsionally (based on the case). The loading used for each scenario is applied on the rigid loading plate as surface tension to amount to a total of 10 kN or 100 kNm. Finally, the maximum displacement or rotation of the plate is considered to be the maximum displacement or rotation at the end of the deck. The displacements and rotations obtained for the vertical, lateral and torsional cases are 0.1624 mm, 0.01644 mm, and 6.908e-5 rad, as observed in the displaced models presented in Figure 3.18. These values determine the mecanical properties calculated below.

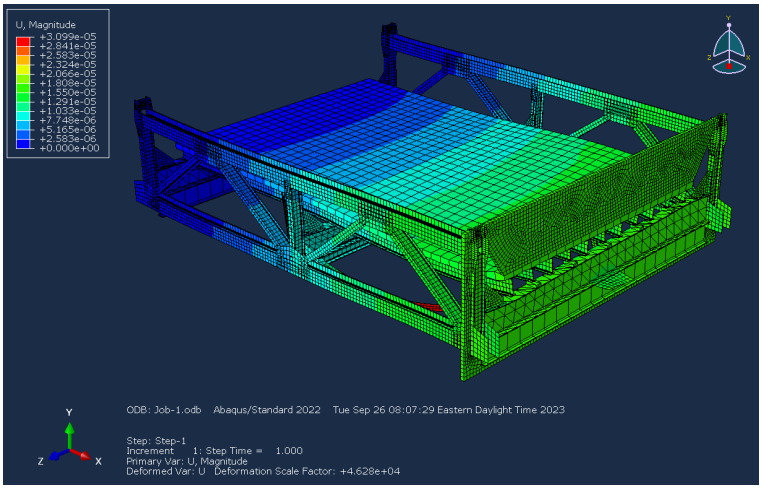
$$I_x = \frac{PL^3}{3E\delta_{max_y}} = \frac{(10,000\,N)\,(11.786\,m)^3}{3\,(200\times10^9\,Pa)\,(1.624\times10^{-4}\,m)} = 0.168\,m^4$$

$$I_y = \frac{PL^3}{3E\delta_{max_x}} = \frac{(10,000\,N)\,(11.786\,m)^3}{3\,(200\times10^9\,Pa)\,(1.644\times10^{-5}\,m)} = 1.66\,m^4$$

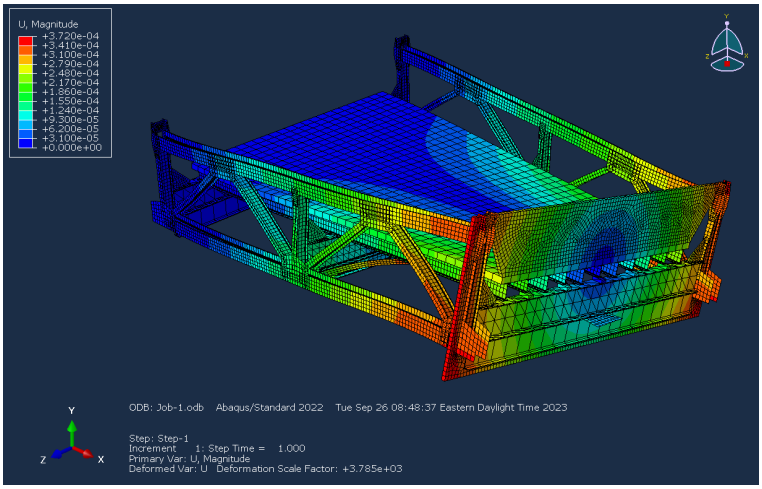
$$J = \frac{TL}{G\theta_{max}} = \frac{TL}{\frac{E}{2(1+v)}\theta_{max}} = \frac{(100,000\,Nm)\,(11.786\,m)}{\frac{200\times10^9\,Pa}{2(1+0.3)}\,(6.908\times10^{-5}\,rad)} = 0.22\,m^4$$



(a) Vertical Loading Case



(b) Lateral Loading Case



(c) Torsional Loading Case

Figure 3.18: Deck Solid Model Displacements

The area of the section is estimated through an average weight of a typical deck segment of

63,730 kg (140,500 lbs), which can be converted to an area through the density of the material (7,850 kg/m^3) and the length of a deck segment (11.786 m). This value also matches the one obtained in the solid deck model, which considers only the structural components of the deck and comes up to 0.667 m^2 as a comparison.

$$A = 63,730\,kg \times \frac{m^3}{7850\,kg} \times \frac{1}{11.786\,m} = 0.689\,m^2$$

The structural properties obtained above for one deck segment are summarized and applied to the fishbone model of the bridge as DeckProfile with a generalized profile definition.

The girder parts in the fishbone model are only created to maintain the actual 3D shape of the bridge, and they do not represent a component in the real bridge. As previously mentioned, the properties of the actual deck are reflected in the deck parts. Therefore, the girder parts are not expected to have a mass or to experience any deflection. To achieve this, the material for the girder parts is defined as "rigid steel" with a density of virtually 0 kg/m^3, a Poisson's ratio of 0 and a modulus of elasticity of 200 GPa. Ideally, the modulus of elasticity of this material would be much greater than the rest of the model (infinity in theory). However, in practice with this type of model in Abaqus, a high difference in the modulus of elasticity between adjacent elements causes instabilities in the model. This issue is solved by significantly increasing the profile properties of the girders to achieve the same effect. As such, an area of 1 m^2, a torsional constant of 10 m^4, and second moments of area about the x and the y axes of 100 m^4 are used. Through testing, it was determined that these arbitrary values are the greatest that can reasonably be used before the simulation encounters instability issues. Also, the stiffness values are much greater than the deck's, meaning that deflections within the girders are negligible, as intended within this context. The profile for the girder is named GirderProfile.

For the bearing members, a better representation is available in Figure 3.27 for the anchor bearing, located in Section 3.5. However, this system is also accurate for the anchor and the tower bearings. While the exact steel beams profiles could not be identified in the structural drawings, it is estimated through visual inspection that the vertical members at each end are W250x131 sections. The vertical bearings are doubled in the X direction for the tower system, thus the area and the second moment of area about the Y axis (using the profile's system, not the global system) are also doubled. Thereafter, the torsional constant is doubled as well, which is a conservative approach. Finally, the second moment of area about the X axis is adjusted using the parallel axis theorem as Equation 3.4, considering a spacing of 457.2 mm between both bearings. For the girders, the profiles are named AnchorBearingProfile and TowerBearingProfile.

$$I = \bar{I} + Ad^2 \tag{3.4}$$

Where I is the second moment of area (m^4) about the new axis, $\bar{I}$ is the second moment of area (m^4) about the axis going through the centroid of the section, A is the area of the section (m^2), and d is the distance between the two parallel axes (m). For the tower bearing profile that is a doubled W250x131 section, the second moment of area about the X axis is calculated with the parallel axis theorem and a distance of 457.2 mm between both members, as per the drawings.

$$I_x = 2\left(\bar{I}_x + Ad^2\right) = 2\left(221 \times 10^6\, mm^4 + \left(16,700\, mm^2\right)\left(\frac{457.2\, mm}{2}\right)^2\right) = 2,187.4 \times 10^6\, mm^4$$

For a more accurate representation of the anchor bearings, they should have been connected to a girder element at the anchor. However, due to initial design choices for the model, the first girder is located only after the first span, such that the anchor bearing is connected directly to the deck member and has not been doubled as it should have. While the expected impact of this change would be minimal in the results, it should still be stated as an inaccuracy in this model version. Overall, the final properties of the profiles are presented in Table 3.6. The values for the anchor bearings are taken directly from the steel handbook [13] for W250x131, and those for the tower bearings have been modified as per the previous explanation.

Table 3.6: Deck Profiles Final Properties

Profiles	A (mm^2)	I_x ($10^6 mm^4$)	I_y ($10^6 mm^4$)	J ($10^6 mm^4$)
Deck	689,000	168,000	1,660,000	220,000
Girder	1,000,000*	100,000,000*	100,000,000*	10,000,000*
Anchor Bearing	16,700	221.0	74.50	3.120
Tower Bearing	33,400	2,187.4	149.0	6.240

*: Arbitrary values used to create a rigid member in place of a change in the modulus of elasticity.

Similarly to the tower members, the profiles of the members forming the deck are defined through the Generalized option, meaning that the section integration of the beam section is performed before the analysis. The sections are named DeckSection, GirderSection, AnchorBearingSection, and TowerBearingSection, after which the beam orientation is defined.

3.2.3 Hangers Parts Modelling

With the main solid part of the infrastructure completed, the cables can be modelled through a set of similar steps. The key difference in the creation of the deck and the towers members compared to the cables is that trusses are used for the latter instead of beam members (as detailed in Section 2.2.3). The cable system can be separated into two types of components: the main cables and the hangers. This section describes the hangers definition, leading to the next section, which covers the main cables. The hangers follow the name notation in Table 3.7 and are defined in Abaqus as single vertical element wires in three dimensions. For the table above, {E/W} represents either "E" or "W" for the east or west side of the bridge, and {i} represents the deck location index, starting at 1 and going up to 25, with index 6 and 20 skipped to account for towers. A particularity of the hangers in the LLRB is that some of the hangers are replaced with structural steel members, which are modelled differently due to the change in the nature of the members. The changes in the definition of these specific hangers are presented towards the end of this section.

Table 3.7: Assigned Terminology for the LLRB Hanger Parts

Type of Part	Name
Hanger	Hanger[{E/W}][{i}]

The hanger parts are modelled as wire parts passing through two datum points. The top and bottom coordinates of all hangers are presented in Table 3.8 below, with the ± sign being used to differentiate the east and west hangers (east positive and west negative). One may note that the Z coordinate matches half of the girder length of 9.144 m (Section 3.2.1). In the presented Table 3.8, the tie column can be ignored for now; it will be used to determine the centre of the connection between the main cables and the hangers in the next section.

Table 3.8: Coordinates of Hanger Elements and Ties

Location Index	X (m)	Y_{bot} (m)	Y_{top} (m)	Z (m)	Tie (m)
0 - No hanger	0.000	0.446	1.949	±4.572	0.5990
1	15.399	0.446	1.949	±4.572	0.3302
2	27.185	0.757	4.012	±4.572	0.3556
3	38.970	1.042	6.795	±4.572	0.3556
4	50.756	1.301	10.273	±4.572	0.3556
5	62.541	1.534	14.447	±4.572	0.3556
T1 - No hanger	74.556	1.744	19.640	±4.572	0.0000
7	86.570	1.928	14.876	±4.572	0.3556
8	98.355	2.082	11.174	±4.572	0.3556
9	110.141	2.210	8.145	±4.572	0.3556
10	121.927	2.312	5.790	±4.572	0.3556
11	133.712	2.388	4.107	±4.572	0.3556
12	145.498	2.388	3.243	±4.572	0.2100
13	157.283	2.389	2.907	±4.572	0.2100
14	169.069	2.388	3.243	±4.572	0.2100
15	180.855	2.388	4.107	±4.572	0.3556
16	192.640	2.312	5.790	±4.572	0.3556
17	204.426	2.210	8.145	±4.572	0.3556
18	216.211	2.082	11.174	±4.572	0.3556
19	227.997	1.928	14.876	±4.572	0.3556
T2 - No hanger	240.011	1.744	19.640	±4.572	0.0000
21	252.025	1.534	14.447	±4.572	0.3556
22	263.811	1.301	10.273	±4.572	0.3556
23	275.597	1.042	6.795	±4.572	0.3556
24	287.382	0.757	4.012	±4.572	0.3556
25	299.168	0.446	1.949	±4.572	0.3302
26 - No hanger	314.567	0.000	0.000	±4.572	0.5990

For the hangers parts, sets are defined on the parts at the datum points locations as per the previously explained convention using set_bot and set_top. For the meshing, a set_alledges is also defined. The seeding is accomplished by edges by number, with a number of one, meaning that a single meshing element is created for each wire segment. The element type assignment is performed similarly to those in the towers and deck sections, with the key difference being that the element type for cables is a truss (T3D2) to match the part type. Thereafter, the mesh is generated based on the seeding defined.

Since the hangers are modelled as truss members, they do not need a profile; their section is defined only by a cross-sectional area and a material. For the LLRB, the cross-sectional area of the hangers is obtained through the structural drawings as $0.0014\,m^2$. For the material used, a Cable Steel material is defined with a density of $7{,}850\,kg/m^3$, a modulus of elasticity of 200 GPa, Poisson's ratio of 0.3, and a thermal expansion coefficient of 11.7 $\mu m/mK$ [82]. Moreover, the Cable Steel material is defined in Abaqus to prevent it from carrying any compressive load.

The structural steel members replacing hangers are situated at location index 1, 12, 13, 14,

and 25. For these members, the naming, the geometric definition, and the definition of the sets are identical to those of the cable hangers. For the meshing of these specific hangers, the primary difference is the use of the B31 beam element instead of truss elements, which results in a profile definition and a different section definition. These hangers are composed of two parallel plates of 38.1 mm (1.5 in) by 182.56 mm (7.1875 in) with a spacing of 215.9 mm (8.5 in). The second moments of area can be determined with Equation 3.5 and Equation 3.6 for rectangles and with the previously defined parallel axis theorem in Equation 3.4.

$$I_x = \frac{bh^3}{12} \tag{3.5}$$

$$I_y = \frac{b^3h}{12} \tag{3.6}$$

Where I_x and I_y and the second moments of area about the X and the Y axis of the profile (mm^4), h is the height of the rectangle (mm), and b is the width of the rectangle (mm). The second moments of area for the double plates are calculated below.

$$I_x = 2\left(\frac{bh^3}{12}\right) = 2\left(\frac{(38.1\,mm)\,(182.56\,mm)^3}{12}\right) = 38.636 \times 10^6\,mm^4$$

$$I_y = 2\left(\frac{b^3h}{12} + Ad^2\right)$$

$$I_y = 2\left(\frac{(38.1\,mm)^3\,(182.56\,mm)}{12} + ((38.1\,mm)\,(182.56\,mm))\left(\frac{215.9\,mm + 38.1\,mm}{2}\right)^2\right)$$

$$I_y = 226.054 \times 10^6\,mm^4$$

Then, the torsional constant of the hangers modelled as beams are obtained as 5.862x10^6 m^4 with the section designer from CSiBridge's Section Designer Module [12]. Finally, the area is obtained as 13,911 mm^2 through the calculation below.

$$A = 2\,(38.1\,mm)\,(182.56\,mm) = 13,911\,mm^2$$

Overall, a generalized profile with the properties defined in Table 3.9 is created with the name HangerBeamProfile, followed by a section of the name HangerBeamSection defined as a beam. Finally, the beam orientation is set to match the actual bridge.

Table 3.9: Hanger Beam Profile Properties

Profile	A (mm^2)	I_x ($10^6 mm^4$)	I_y ($10^6 mm^4$)	J ($10^6 mm^4$)
HangerBeam	13,900	38.64	226.1	5.862

3.2.4 Main Cables Parts Modelling

This section covers the creation of the main cables as wires in a three-dimensional model and the auxiliary components that come with it: the cable connectors, the main cable restraints and the cable supports. To simplify the application of cable damage in later stages of the research and to facilitate modelling the constraints within the cables, each cable segment (span between hangers) and each wire (of the set of 12) of the main cables are modelled as individual parts. The names for the parts mentioned in this section are presented in Table 3.10, with {E/W} representing either "E" or "W" for the east or west side of the bridge, {i} representing the deck location index starting at 0 at the first anchor and increases by 1 at every end of main cable span (up to 26), with index 6 and 20 being replaced by "T1" and "T2", {j} representing the index of the main cable restraint on its span (0 or 1), {Y/Z} representing either "Y" or "Z" for the direction of the support, {n} representing the column index of the wire in the parallel arrangement as A, B, or C, and {m} representing the row index of the wire in the parallel arrangement as 1, 2, 3, or 4.

Table 3.10: Assigned Terminology for the LLRB Main Cable Parts

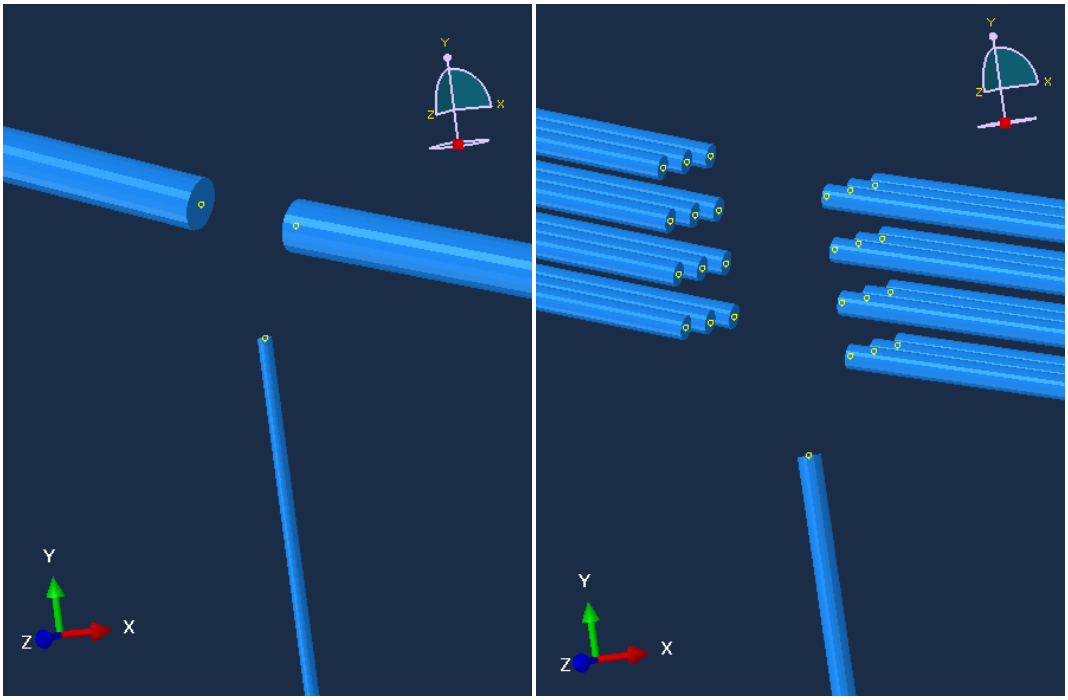
Type of Part	Name
Main Cable Wire	MCWire[{E/W}][{i}/{i+1}][{n}{m}]
Cable Connection	CableConnection[{E/W}][{i}]
Main Cable Restraint	MCRestraint[{E/W}][{i}/{i+1}][{j}]
Cable Support	[{Y/Z}]Support

To begin, datum points are created in Abaqus at a cable's start and end coordinates. The part is then modelled as a linear wire by connecting both points. As per the parts described in the previous sections, sets are created at each end of the wires as set_beg and set_end. Then, sets must also be made at every location where the cable is restrained through clamps with other parallel cables, referred to as main cable restraints. A set_mid{i} (i=0, 1, ...) is created for every location with a main cable restraint. However, since there are no datum points at those locations, the set(s) must be defined through meshing nodes instead.

For the meshing, the seeding is accomplished by edges by number (using the terminology in Abaqus), with the number of elements being increased by the number of restraints in the main cable span. The LLRB has one or two main cable restraints per span (cable segment), leading to

two or three meshing elements in the seeding. The element type assigned is a T3D2 truss, the same type as the hangers. Once the part meshing is completed, the aforementioned set_mid{i} may be created. Defining middle sets for main cable constraints through the mesh ensures that the main cable restraints are correctly located and separate the main cable spans uniformly, as in the actual bridge.

Regarding the global geometry of the main cables, each cable is designed to connect the top of two consecutive hangers (or the tower top/anchor), considering the height of the ties. In Table 3.8 is presented the top coordinates of the hangers on which the height of the ties may be added to obtain the cable connector centre coordinates. However, some slight modifications are made to account for the main cable connector and the physical connection between the main cables and the hangers. An example of this type of member can be noticed in Figure 3.20, which displays an image of the element in the actual bridge. To allow this member to be incorporated into the model, both ends of the main cable wires are moved closer together to shorten the apparent length of the cables. The cable connectors, described later in this section, should link the end of a cable with the beginning of another cable and fit within the created gap in the model. The other modification applied is a shift for the main cable wire pattern. As mentioned in Section 3.1.3, the main cables in the LLRB are composed of 12 individual cables set in a parallel formation. These 12 cables form a 3x4 (three wires along the width and four along the height) rectangular pattern along the entire length of the main cables, as presented in Figure 3.20. This section focuses on creating the main cable wires as 12 individual cables; however, two versions of the model are developed for this research. The second version simplifies the main cable bundle into a single equivalent wire, significantly reducing the number of DOFs and the computation time for the model. The explanations in this section are also applicable to the 1x1 main cable wire version, with a simple reduction of the column and row index of the wires to only “A” and “1”. It follows that the cross-sectional area of the single cable must equal the sum of the areas of the 12 wires. The overall differences between the two models employed in the current study can be observed in Figure 3.19.



(a) 1x1 Version (b) 3x4 Version

Figure 3.19: Main Cable Bundle Variations

To accurately model this geometry for any model version, each cable defined in Abaqus must have a coordinate offset to account for its position in the pattern (1x1, 3x4, or any other combination). As such, a shift in the Y and Z directions is considered based on the specific wire to model. Equations 3.7 and 3.8 are used to determine the total coordinate shift required to account for the cable connector.

$$\left(x_{b_{ij}}; y_{b_{ij}}; z_{b_{ij}}\right) = \left(x_{b_c} + \frac{L}{2}; y_{b_c} + H\left(i - \frac{n+1}{2}\right) + \frac{L}{2}tan(\alpha_b); z_{b_c} + W\left(j - \frac{m+1}{2}\right)\right) \tag{3.7}$$

$$\left(x_{e_{ij}}; y_{e_{ij}}; z_{e_{ij}}\right) = \left(x_{e_c} - \frac{L}{2}; y_{e_c} + H\left(i - \frac{n+1}{2}\right) - \frac{L}{2}tan(\alpha_e); z_{e_c} + W\left(j - \frac{m+1}{2}\right)\right) \tag{3.8}$$

Where $\left(x_{b_{ij}}; y_{b_{ij}}; z_{b_{ij}}\right)$ are the corrected coordinates for the beginning of the cable ij (m), $\left(x_{e_{ij}}; y_{e_{ij}}; z_{e_{ij}}\right)$ are the corrected coordinates for the end of the cable ij (m), $(x_{b_c}; y_{b_c}; z_{b_c})$ are the coordinates of the centre of the cable connector, or top of tie, at the beginning location (m), $(x_{e_c}; y_{e_c}; z_{e_c})$ are the coordinates of the centre of the cable connector, or top of tie, at the end location (m), α is the angle of the cable connector (°), L is the length of restraint of the cable

connector (m), H is the vertical spacing between two main cable wires (m), W is the lateral spacing between two main cable wires (m), n is the number of main cable wires in the vertical direction (4), m is the number of main cable wires in the lateral direction (3), i is the vertical index of the main cable wire (1, 2, 3, or 4) and j is the lateral index of the main cable wire (1, 2, or 3). The length of restraint can be interpreted as the distance along the X axis for the main cable, for which the connector restricts the main cable. Then, the angle of the cable connector is the angle of the main cable wire passing through the cable connector about the X axis. This value is selected as the average angle between two adjacent main cable spans. The angle of the cable connector at the anchor and the tower is taken as 0°. The rationale for the specific geometry described above is that the supported length of the main cable wires forms an unknown angle that must join the angles of the two adjacent wires, as observed in Figure 3.20 [9]. It is, therefore, assumed that the orientation of the connection tends to minimize the maximum angle between the free and fixed part of the cable. The notation used for the 12 wires of the main cables follows the one defined in Figure 3.2.



Figure 3.20: Cable Connector From the 2011 Visual Inspection [9]

As per the structural drawings, the cross-sectional area of the main cable wires is 0.00159236 m^2 per wire, for a total of 0.0191083 m^2 for the 12 wires. The wire section in Abaqus is named MainCableSection and the same material as the hangers is used in the section definition: Cable Steel. However, in some of the analysis cases (defined in Chapter 4) where some main cable wires would be damaged, the damage is applied as a reduction of the cross-sectional area of the main cable section, named MainCableSection{d}, where d is the proportion of the area reduction with a precision of six decimals (i.e. 0.000000 for the undamaged cables and 0.100000 for a cable having only 90% of its original cross-sectional area). Section 2.3.2 describes how a cross-sectional area reduction is an appropriate modelling simplification of corrosion in some strands, which forms the basis of the approach used here. This assumption essentially comes from the fact that as small

section loss quickly results in the failure of the strand, which has no effect on the remaining strand forming the wire. Moreover, Appendix H introduces more details regarding these damage levels and their interpretation in the context of the LLRB by considering the cable restraints on the main cables and the effective length of stiffness reduction. Thereafter, the material is modified through density to account for the mass loss from the cross-sectional area reduction. In reality, a cable with broken strands loses effective area, but the mass of the material does not change proportionally to the loss of elastic properties. As such, the density of the material (Cable Steel {d}) is increased as per Equation 3.9 to counter the loss of mass that should not occur.

$$\rho_{corr} = \frac{\rho}{1-d} \tag{3.9}$$

Where ρ_{corr} is the corrected density value (kg/m^3), ρ is the original density value (kg/m^3), and d is the damage proportion of the cable as a loss of area. This modification to the model increases the accuracy of the mass of the damaged cables, making the response more reliable. This modification could also be applied to the hangers if damage was to be simulated on them in a further study.

The connection between the hangers and the main cable parts in the fishbone bridge model must be established. While the applied connections are defined in Section 3.5, the cable connection is not a simple constraint between a hanger and the main cable, partly due to the complexity of separating the main cable into 12 wires. To accurately model the connections and the relations between all the parallel wires, near-rigid parts are created to represent the actual cable connectors (Figure 3.20). These parts are composed of multiple wire beams connecting every main cable wire end to the cable connector centre and the cable connector centre to the top of the hanger, such that a cable connector has 25 wires for the LLRB with 26 datum points (24 main cable wire ends, the centre point and the hanger top), as presented in Figure 3.21. The length of the vertical component connecting the top of the hanger to the centre point of the part is the length of the tie, presented in Table 3.8. There are two exceptions to the creation of the cable connector part. First, the anchor location does not have a cable connector since it is assumed that the cables are anchored at that point and that boundary conditions are applied at the same location (Section 3.6). In reality, the cables are prolonged further to reach the anchor, which is lower than the bridge deck. This detail does not impact the accuracy of the LLRB modelling, allowing the simplification described. Second, the tower location has a cable connector similar to the top of the anchor, except that the vertical component is calculated based on the provided Y coordinate of the tower top and the Y_{top} value provided at the tower location in Table 3.8. The top of the tower has a Y coordinate of

19.076 m, which can be obtained from a combination of Figure 3.5 and the tower bearing height of 2.004 m. Meanwhile, the specified Y_{top} value at the tower locations is 19.640 m, meaning that the vertical component measures 0.564 m along the Y axis.

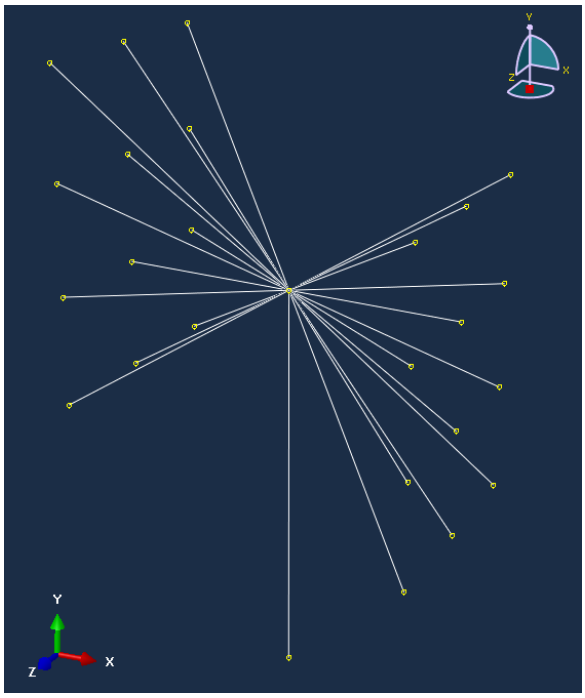


Figure 3.21: Cable Connector Part

The sets within the cable connector part can be separated into three categories. First, the set that connects to the top of the hanger is denoted set_hanger_con. Second, the set at the centre of the part at the top of the tie is called set_ctr. Third, the sets at the locations which connect to the end of the main cable wires are descriptive of the main cable span where they connect, as well as the wire index, previously referred to as {n}{m}. As defined, the wire index represents the column (A, B, or C) and the row (1, 2, 3, or 4) of the main cable wires arrangement. Therefore, the name of these steps is set_[{i}/{i+1}][{n}{m}], which matches the notation from Table 3.10 regarding all indices.

Similarly to the other components previously described in this section, the seeding for the meshing is done by edges by number, with a single meshing element per individual wire. The element type is defined as a B31 beam for each part to later allow for the implementation of the part rigidity compared to the cables. Finally, the elements are meshed based on the established seeding.

For the profile of the cable connector, a similar strategy as employed for the girder is used to increase the rigidity and reduce deformations within the member artificially. The goal of using

a high rigidity is to ensure that the distance between parallel wires remains primarily constant, which assumes that the steel members establishing the connection do not undergo considerable deformations. Moreover, it controls the length of the main cable wires within the cable connector, which should experience minimal deformations. While a solid member in place of the cable connector in the model would slightly increase the model's accuracy, the simplification has no significant effect on the structural properties of the structure. However, it considerably reduces the complexity of computing the model. Therefore, the properties presented in Table 3.11 are used to mimic the desired behaviour.

Table 3.11: Cable Connection Profile Properties

Profiles	A (mm^2)	I_x ($10^6 mm^4$)	I_y ($10^6 mm^4$)	J ($10^6 mm^4$)
Cable Connection	100,000*	100,000*	100,000*	100,000*

*: Arbitrary values used to create a semi-rigid member in place of a change in the modulus of elasticity

In line with the definition of the girder parts, the material used for the main cables is Rigid Steel, which differs from Structural Steel only with its density of nearly zero and Poisson's ratio of zero. The section CableConnectionSection is therefore defined through the CableConnectionProfile and the Rigid Steel material. The beam orientation is defined for the part, which concludes the definition of the cable connection parts.

The main cable wires of the LLRB are restrained at their mid (or one-third/two-thirds exceptionally) span with main cable restraints, as presented in Figure 3.22 [9]. These restraints are clamps that hold the spacing between all the parallel wires. Therefore, the modelling steps for these parts resemble those used for the cable connectors. Since the part is modelled as near rigid, any main cable wire is selected as the starting datum point for the part, and a wire is traced to all other main cable wire locations (as the other datum points for the part), such that 11 wires are defined (12 main cable wires, but the selected main cable wire does not connect to itself). The result for this part is presented in Figure 3.23 [9], in which the location of the bottom right main cable wire (A1, which is at the lowest Y and Z coordinates) connects to the other 11 main cable wire locations. These wires are designed to simulate a rigid part that should experience minimal deformation and essentially act as a solid rigid part that connects all wires. Sets within the part are defined at each datum point with the main cable span on which they are, the wire index of the specific main cable wire location, and a supplementary integer for the index of the cable restraint along the span (0 or 1). These sets are then named set_[{i}/{i+1}][{j}][{n}{m}], which is consistent with the notation presented in Table 3.10. The definition of the rest of the parts follows precisely the same steps as the one for the cable connection parts, which are described above.



Figure 3.22: LLRB Cable Restraint [9]

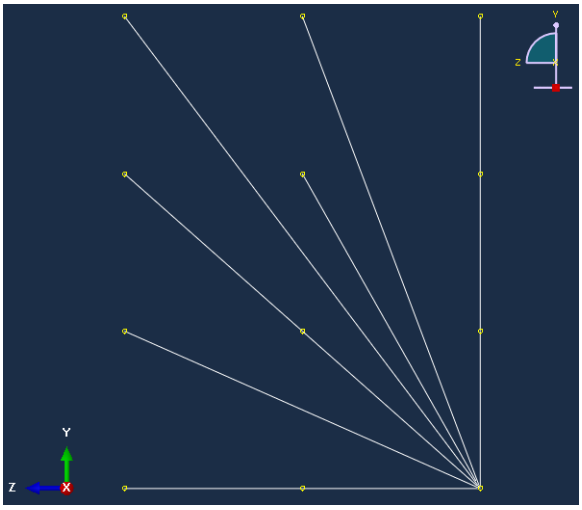


Figure 3.23: Cable Restraint Part

The last parts that must be defined for the LLRB model are the cable supports. These parts do not represent any physical component in the bridge, but are created to artificially reduce instabilities in the Pre-Displacement phase. Instability issues are common in models using axial bar (truss) elements, as these elements cause the global stiffness matrix to become singular. To avoid this issue, additional bar elements are added to problematic nodes, which are the connection points between the hangers and the main cables. Selected axes should be considered for these bars based on the problem at hand, and the end of the bar that is not attached to the model should be fixed at an arbitrary point in space, allowing it to populate the problematic stiffness matrix components with non-zero values. These added elements should have a stiffness near zero so that they do not impact the structural performance, but not zero to prevent the global stiffness matrix from being singular. These bars allow the model to stabilize in its initial state without

aborting the simulation. As the model displaces into a state of equilibrium, these members become irrelevant (assuming that non-linear geometry is considered in the model). A study by Alkharisi and Heyliger [83] briefly explains the use of this type of member in a similar type of model, which is the origin of this stabilization method in this research.

These stabilization elements are designed as 1 m long truss elements with very low cross-sectional area and very low modulus of elasticity. These members, which connect to the cable connector centre point, can deform while exerting negligible force on the model. These parts are either 1 m along the Y or the Z axis, with sets named set_con and set_pin, where set_con is at the datum point at the cable connector centre and set_pin at the other extremity. The parts meshing is done as per the other cable elements, seeding by edges by number with a single element per wire and using the T3D2 truss element type. For the section, the area is set to a value of 2 mm^2 and the material used, Support Material, has a modulus of elasticity of 1 Pa, a Poisson's ratio of 0.3, and a near zero density. The area and elasticity parameters selected are purely arbitrary to result in a negligible axial stiffness when compared to actual real adjacent components. Overall, this material's axial rigidity (EA) is $2\text{x}10^{-6}$ N, compared to a value of $0.28\text{x}10^{9}$ N for a hanger, resulting in a difference factor of $100\text{x}10^{12}$. This means that the structural contribution of the support members can be considered or ignored without noticeable change in the results. Exceptionally, a single Y support part and a single Z support part are created in the model, even if these parts exist at every cable connector centre. These artificial components are all identical, and their structural response is not of interest. Moreover, since they are artificial components, no damage is to be applied to them in future research (as opposed to identical deck components, for example), which explains why all instances are created using the same two parts.

3.3 Model Assembly Definition

All required parts defined in the previous section can be added to the LLRB model assembly. Since most parts are created with their final coordinates, there is no need for translation or rotation within the assembly once the parts are created as instances. The exceptions to this are the YSupport and ZSupport parts, which are each instantiated 46 times (23 hangers on each side of the bridge). These instances are then translated such that their set_con point is at the set_ctr of the Cable Connection parts above hangers (excluding the Cable Connectors above towers). Since the mesh is created at the part level, the mesh type is set to dependent for all instances in the assembly. This is the best alternative for the mesh for a model composed of only wires, since it ensures consistency in the mesh quality, controlled by the specified inputs. Moreover, this

means that the mesh definition is not an iterative process and is slightly more efficient than the independent type alternative. Finally, the sets defined in the parts are transferred to the assembly as {part name}.{set name}. The resulting assembly is presented in Figure 3.24.

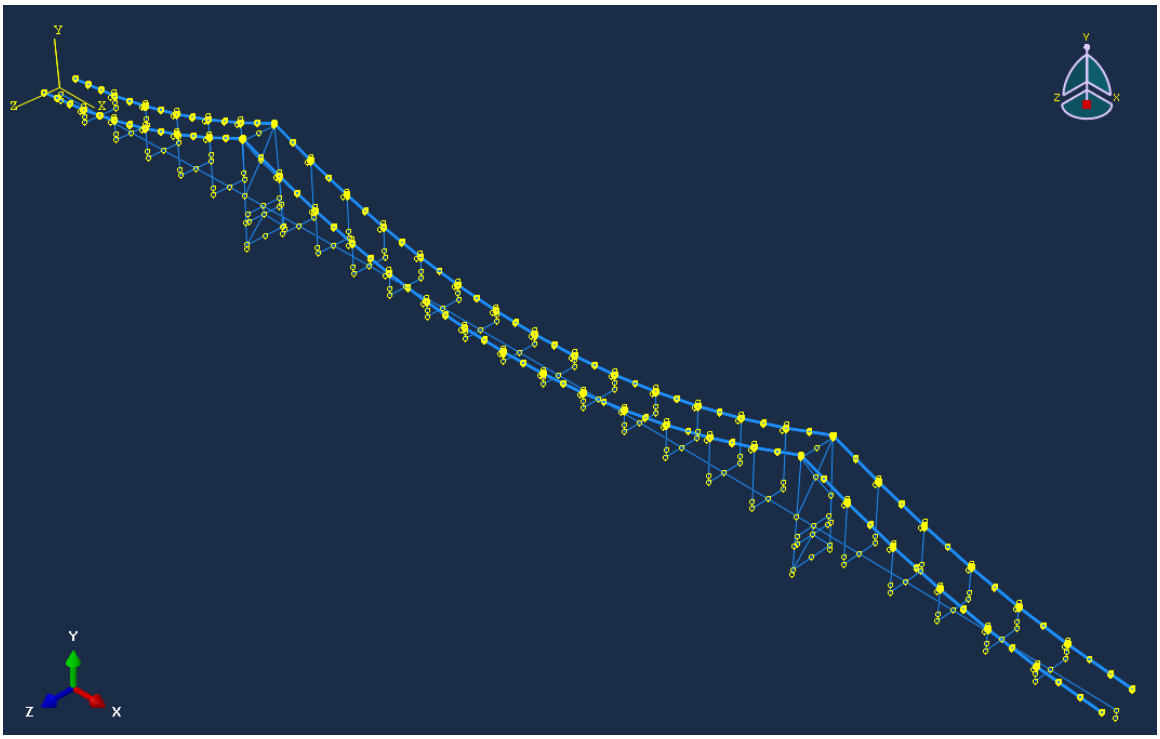


Figure 3.24: LLRB Model Assembly

3.4 Model Steps Creation

The next stage in the model's definition is developing the steps required to accomplish the different model analyses. This section defines the five analysis steps used in the model: Initial, Pre-Displacement, Vertical Loading, Wind Loading (optional) and Frequency (optional). Each step is a simulation stage that controls loading and boundary conditions within the assembly. The model's defined steps are static (Static, General) or frequency analysis (Frequency). Some optional steps are defined only for a subset of the test series, which is the case for the Wind Loading and Frequency steps.

The first step is the Initial step, which is automatically generated in Abaqus and executes the simulation setup. This step defines the starting boundary conditions, the predefined fields (used for applying pretension in the cables — Section 3.8), the assembly with all parts' positions and meshes, the interactions (not used in this model), and the part's geometry. While no active analysis is performed, this setup step is required to form the model's basis.

The second step is the Pre-Displacement step, establishing the transition between the Initial and the Vertical Loading step through a static general step. This step is required to correct the instability issues occurring in the model analysis when the model is subjected to gravity loading. The implication of these stability adjustments in the pre-displacement step is to add a boundary condition to the top of some selected hangers. In each global span (Anchor1-Tower1, Tower1-Tower2, and Tower2-Anchor2), the hanger closest to the middle of the span is chosen on each side, for a total of six hangers. The top of these hangers is shifted downward (Y negative) by 0.01 m through the new boundary conditions created in the pre-displacement step. This predefined temporary displacement was determined to mitigate the issue of the global stiffness matrix being singular at the initial position. By disrupting the initial state of the model and then releasing the forced displacement, there is no impact on the final state of the model, but the aborted error of the model is avoided at the start of the Vertical Loading step. This step uses the default time period of 1 second and has nonlinear geometry (Nlgeom) activated. This setting is used to account for the large deformations within the model by considering the changes in geometry (displacements and rotations) during the analysis. As such, the stiffness matrix and the internal forces are updated through the step, affecting load paths and equilibrium. Such a nonlinear setting is required for a suspension bridge model to accurately assess the result, as large deformations are expected in this type of structure. Furthermore, an automatic stabilization of the model is applied by specifying an energy fraction that is dissipated of 0.0002, the default value. This setting creates an artificial damping of the system to prevent instabilities by specifying the fraction of the strain energy allowed to dissipate in the system. With this option, the adaptive stabilization should also use a maximum stabilization ratio to strain energy of 0.05. This value ensures that the artificial dissipation does not govern the solution to the model by limiting the total amount of damping, preventing over-damping. This helps the model to converge to a solution more consistently while maintaining a high accuracy. The step timing incrementation is set to a maximum of 100 increments, which was sufficient in all scenarios tried within this research. The increment size has an initial value of 0.1 s, with a minimum of $1x10^{-6}$ s and a maximum of 1 s, the length of the step. If the model is found to abort calculations within this step, adjusting the increment sizes can help fix the issue. Automatic incrementation allows the model to self-correct as the model computes.

The third step is the Vertical Loading step, also a static general step, which mainly applies gravity to the model. In this step, gravity is incorporated into the model (Section 3.7 for the loads) while removing the forced pre-displacement of the selected hangers. This causes the model to never be in an unstable state in any of the static steps. Also, this is the step in which the

predefined fields are modified to apply the pretension forces in the cables (Section 3.8). As per the previous step, the nonlinear geometry option (Nlgeom) is activated, the automatic stabilization is set to a specified dissipated energy fraction of 0.0002, and an adaptive stabilization is made with a maximum ratio of stabilization to strain energy of 0.05. For the incrementation, a maximum number of increments of 1000 is used with increment sizes of 0.1 s, $1x10^{-6}$ s, and 1 s for initial, minimum and maximum, respectively.

The fourth step is the Wind loading step, the last static general step defined. This step is only used for the test series in which wind is the focus and is not created otherwise. The purpose of this step is to add the wind-loading components to the model, which are described in Section 3.7. Moreover, all previously defined loads, boundary conditions, constraints, and predefined fields are maintained from the Vertical Loading step. For this step's settings, the stabilization parameters from the previous step are considered, with a modification of the maximum increment and an initial increment size to a value of 0.25 s. This value matches the sampling frequency of the wind signal defined in Section 3.7.

The fifth and last step of the baseline analysis is a Frequency step. This step is only defined for test series in which the natural frequencies must be extracted and is not created otherwise. In this step, gravity is considered built into the base state by default, while the applicable boundary conditions and predefined fields from the previous step are maintained. However, this step is a Frequency type step, falling under the linear perturbation procedure type. The Nlgeom option is propagated from the previous step, and the Subspace Eigensolver is selected. Using the Subspace method over the Lanczos method is for faster solution convergence. The Lanczos method is better at handling larger models with many degrees of freedom but requiring fewer vibration modes. With this method, sparse matrices are handled faster due to the iterative nature of the solver, making the method very efficient for most cases. On the other hand, the Subspace method is more adapted to models of up to moderate size that require more modes. This solver stores the entire mass and stiffness matrices in memory, which is highly efficient for smaller models but becomes unreasonable in larger models due to the required memory [84]. For the LLRB model, the problem size is quantified as 9,475 nodes and 28,048 variables, making it a moderate-size model for which Subspace is ideal. While the modes of vibrations of interest are from the first 10-20 eigenvalues, the first 50 are requested to ensure that results are flexible and scalable for deeper analysis. The rest of the Subspace solver settings are kept at their default values.

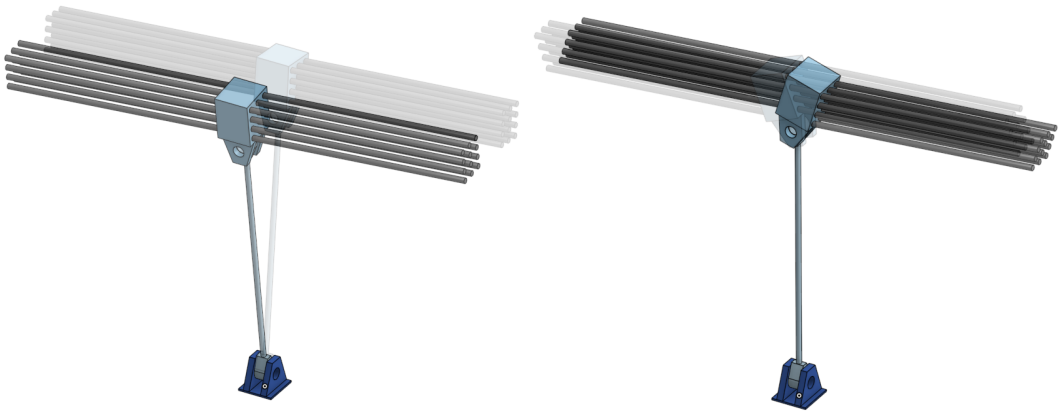
3.5 Model Constraints Development

At this point in the development of the LLRB model, all the instances are defined in the assembly without any relation to each other. Therefore, the constraints between instances must be created to form the bridge. For this task, there are 15 types of constraints defined in the model, for a total of 2,117 constraints. The name and number of constraints are presented in Table 3.12 below, with notations matching those of Tables 3.1, 3.4, 3.7, and 3.10 for the parts names. This section describes the details of each type of constraint. The greatest majority of the constraints defined in the model are beam MPC (Multi-Point Constraints). This constraint enforces a beam-like behaviour between nodes, acting as a simple rigid link that couples all the applicable degrees of freedom. However, an alternate MPC is the tie constraint type, which matches the displacements of two nodes to enforce that there is no relative movement between both connected regions. One thing to note is that when a beam MPC is defined with a length of zero between the two specified nodes (the case in most constraints in this model), it automatically defaults to a tie MPC. For simplicity in the modelling of the bridge, the beam type is always selected to ensure that if two nodes that should be overlapping are not perfectly aligned, the model handles the case as intended with the appropriate coupling of the nodes. Therefore, in most cases, the constraints are automatically changed to a tie type in the preliminary part of the analysis. For all constraints, the criteria for determining the master and slave nodes are as follows: 1) the master node should be in the region with the greatest stiffness, 2) the master node should be in the more significant region, and 3) the master node should be in the region with the coarser mesh [85]. The criteria are followed in order, with subsequent criteria used only if the previous one is inconclusive.

Table 3.12: Constraints Names and Count of Occurrences in the Model

Type of Constraint	Name	Count
MCWire to CableConnection	CableConnectionMCTie[{E/W}][{i/i+1}] [{i}/{i+1}][{n}{m}]	1,200
Hanger to CableConnection	CableConnectionHangerTie[{E/W}][i]	46
Tower to CableConnection	CableConnectionHangerTie[{E/W}][T{1/2}]	4
Support to CableConnection	CableSupport{Y/Z}[{E/W}][{i}]	92
MCWire to MCRestraint	MCRestraintTie[{E/W}][{i}/{i+1}][{j}] [{n}{m}]	672
Hanger to Girder	HangerToGirder[{E/W}][{i}]	46
Tower to TowerBrace	Tower[{E/W}][T{1/2}][{k}]	12
Tower to TowerCrossGirder	Tower[{E/W}][T{1/2}][{k}]	4
TowerCrossGirder to TowerBrace	Tower[M][T{1/2}][{k}]	2
CrossTowerBrace to CrossTowerBrace	Tower[T{1/2}][cross0]	2
TowerBearing to TowerCrossGirder	TowerBearingBot[T{1/2}][{E/W}]	4
TowerBearing to Girder	TowerBearingTop[T{1/2}][{E/W}]	4
TowerCrossGirder to Girder	TowerGirderCentre[T{1/2}]	2
Girder to Deck	DeckToGirder[{i}]	25
AnchorBearing to Deck	DeckAbutment[{S/N}	2
	Total:	2,117

The first beam MPC constraint defined in the model is the connection between the main cable wires and the cable connection parts. As defined in the parts section, the ends of the main cable wires match the coordinates of the cable connection arms. As such, a CableConnection-MCTie[{E/W}][{i/i+1}][{i}/{i+1}][{n}{m}] would connect the following sets: CableConnection-[{E/W}][{i/i+1}].set_[{i}/{i+1}][{n}{m}] and MCWire[{E/W}][{i}/{i+1}][{n}{m}].set_{beg/-end}. The master node is granted to the cable connection due to its very high stiffness compared to the main cable wire, which is considered the slave node. The 26 main cable spans of 12 wires each are constrained this way, except the anchors, for a total of $(26 \times 2 - 2) \times 12 \times 2 = 1,200$ constraints considering both sides. While the constraints in the real components are more complex (Figure 3.25), the cables are modelled as truss elements, which means that the more accurate degrees of freedom may not be perfectly represented.



(a) Bottom Rotational Degree of Freedom (b) Top Rotational Degree of Freedom

Figure 3.25: Cable Connector Behaviour Display

Another beam MPC constraint involving the cable connection parts is their connection to the hangers. As previously mentioned, the top of the hangers must be connected with the bottom of the cable connections, such that a CableConnectionHangerTie[{E/W}][{i}] connects the sets: CableConnection[{E/W}][{i}].set_hanger_con for the master node and the Hanger[{E/W}][{i}].set_top for the slave node. The selection of the master and slave nodes uses the same logic as per the previous constraint; the stiffness of the cable connection is much greater than the cables that connect to it. Accordingly, 46 constraints are defined in this manner, one for each of the 23 hangers on each side. The tower to cable connection constraints follow the same process, with the only difference being that the {i} becomes T{1/2} for the names since the cable connections are stiffer than the vertical tower parts as well, producing four constraints (one per tower).

The last beam MPC constraint with the cable connection parts is those with the cable supports. The constraint is defined between the centre of the cable connection and the connection end of the cable support, with the cable support being a very flexible member. A CableSupport{Y/Z}[{E/W}][{i}] constraint would connect the CableConnection[{E/W}][{i}].set_ctr as the master node to the {Y/Z}Support[{E/W}][{i}].set_con as the slave node. This produces 92 constraints, for two supports at each hanger (23) on each side of the bridge.

Regarding the main cables, there are also beam MPC constraints to be created at points where they are restrained, usually at their midspan. Since the main cable restraint is also made to be rigid, it will be granted the master node. A MCRestraintTie[{E/W}][{i}/{i+1}][{j}][{n}{m}] connects the MCRestraint[{E/W}][{i}/{i+1}][{j}].set_[{i}/{i+1}][{j}][{n}{m}] as the master node to the MCWire[{E/W}][{i}/{i+1}][{n}{m}].set_mid{j}. There are 26 main cable spans, two of which have double restraints, creating a total of $(26 + 2) \times 12 \times 2 = 672$ constraints.

The last cable requiring beam MPC constraints is the hanger, specifically the girder to hanger

bottom constraints. The girder is designed as a rigid body, with a much greater stiffness than the hanger, making it the master node. A HangerToGirder[{E/W}][{i}] handles the connection between the master node Girder[{i}].set_{beg/end} and the slave node Hanger[{E/W}][{i}].set_bot. This creates 46 constraints, one for each hanger (23) on each side.

Next, the tower parts are constrained to one another to form the tower structure through beam MPC constraints. For all the following constraints, the tower part is selected as the master node since it has the greatest rigidity. The first step of this process is connecting the tower (vertical component) to the tower braces and cross girders. This is done for the bottom of the chevron brace and the ends of the lower horizontal brace, by connecting the Tower[{E/W}][{0/1}].set_mid0 as the master node to the TowerBrace[Chevron[0]:{E/W}][{0/1}].set_beg and TowerBrace[Horiz[0]]-[{1/2}].set_{beg/end} as the slave nodes, forming the constraint Tower_[{E/W}][T{1/2}][2]. This constraint has four occurrences, one for each vertical tower part. Then, the ends of the tower cross girder are connected to the tower. The constraint Tower_[{E/W}][T{1/2}][3] is created through the nodes at Tower[{E/W}][{1/2}].set_mid1 as master and at TowerCrossGirder-[{1/2}].set_{beg/end} as slave, creating four additional constraints (one at each end of the tower cross girders). Afterwards, the constraints between the tower and the bottom of the cross brace can be established as Tower[{E/W}][T{1/2}][4], connecting Tower[{E/W}][{1/2}].set_mid2 as master set to TowerBrace[Cross[0]:Bot-{E/W}][{1/2}].set_beg as slave set, adding four constraints to the count. Then, the top of the cross brace and the ends of the upper horizontal brace are constrained to the top of the vertical tower part. This constraint is named Tower_[{E/W}][T{1/2}][5] and connects Tower[{E/W}][{1/2}].set_top as the master node to TowerBrace[Cross[0]:Top-{E/W}]-[{1/2}].set_beg and TowerBrace[Horiz[1]][{1/2}].set_{beg/end} as the slave nodes, adding another four constraints to the model. Next, the top of the chevron brace must be constrained to the centre of the tower cross girder. This constraint is denoted Tower_[M][T{1/2}][3] and connects TowerCrossGirder[{1/2}].set_ctr as master node to TowerBrace[Chevron[0]:E][{1/2}].set_end and TowerBrace[Chevron[0]:W][{1/2}].set_end as slave nodes, adding two constraints to the count. Note that the cross girder is selected as the master node since it has considerably more stiffness than the braces. Finally, the last step for the internal tower constraints is the centre of the cross brace, named Tower[T{1/2}][cross0]. Due to the symmetry of the cross brace, the west bottom brace component is selected as the master member. Therefore, this constraint connects the TowerBrace[Cross[0]:Bot-W][{1/2}].set_end as master node to TowerBrace[Cross[0]:Bot-E][{1/2}].set_end, TowerBrace[Cross[0]:Top-W][{1/2}].set_end, and TowerBrace[Cross[0]:Top-E]-[{1/2}].set_end as the slave nodes, adding a final four constraints to the internal tower constraints

count.

After completing the tower constraints, the tower can be connected to the deck through the girders at the tower. However, before this, the actual connection of the deck to the tower via the bearings is presented in Figure 3.26 [9], which is simplified into a visual representation in Figure 3.27 illustrating the bearing constraints and the expected translation (dark arrows). In this representation, the bottom longitudinal member is the tower cross girder, whereas the top longitudinal member is the girder at the tower. Also, the vertical members at both ends are the tower bearing, and the member at the centre is the vertical wire in the girder at the tower part. As Figure 3.27 demonstrates, the constraints at the external tower bearings allow for a double rotation on the same axis, more specifically about the Z axis in the bridge model. On the other hand, the connection at the centre only restricts displacements along the Z axis. In theory, this connection also prevents rotation about the X and Y axes, but this becomes a redundancy that creates instability in the model, with the external tower bearing already preventing such rotations. As such, the connection with the deck is made with five constraints per tower. First, TowerBearingBot[T{1/2}][{E/W}] is a coupling constraint that uses the TowerCrossGirder[{1/2}].set_{beg/end}_mid as master node and TowerBearing[{E/W}][T{1/2}].set_bot as slave node. The girders are stiffer than the tower bearing; hence, the girder is used for the master nodes. The coupling constraint uses the default kinematic type and constrains the U1, U2, U3, UR1, and UR2 degrees of freedom while leaving UR3 free. This represents the bottom of the external tower bearing and adds two constraints per tower, or four total in the model. Second, TowerBearingTop[T{1/2}][{E/W}] is also a coupling constraint, this one using Girder[T{1/2}].set_{beg/end} as master node and TowerBearing[{E/W}][T{1/2}].set_top as slave node. The coupling uses the same settings as the tower bearing bottom to allow the same type of rotation, as previously discussed. This adds two constraints per tower, for a total of four in the entire assembly. Third, the TowerGirderCentre[T{1/2}] is a coupling constraint between Girder[T{1/2}].set_ctr_bot as master node due to the high rigidity of the girder and TowerCrossGirder[{1/2}].set_ctr as the slave node. The coupling type used is kinematic also, but the constrained degrees of freedom are limited to U3, whereas U1, U2, UR1, UR2, and UR3 are free. This is to recreate the lateral restriction observed at the central connection between the two girders. The final constraint is thus added to the deck-to-tower connections, introducing two more constraints to the model.



Figure 3.26: Anchor Bearings Picture From the 2011 Visual Inspection [9]

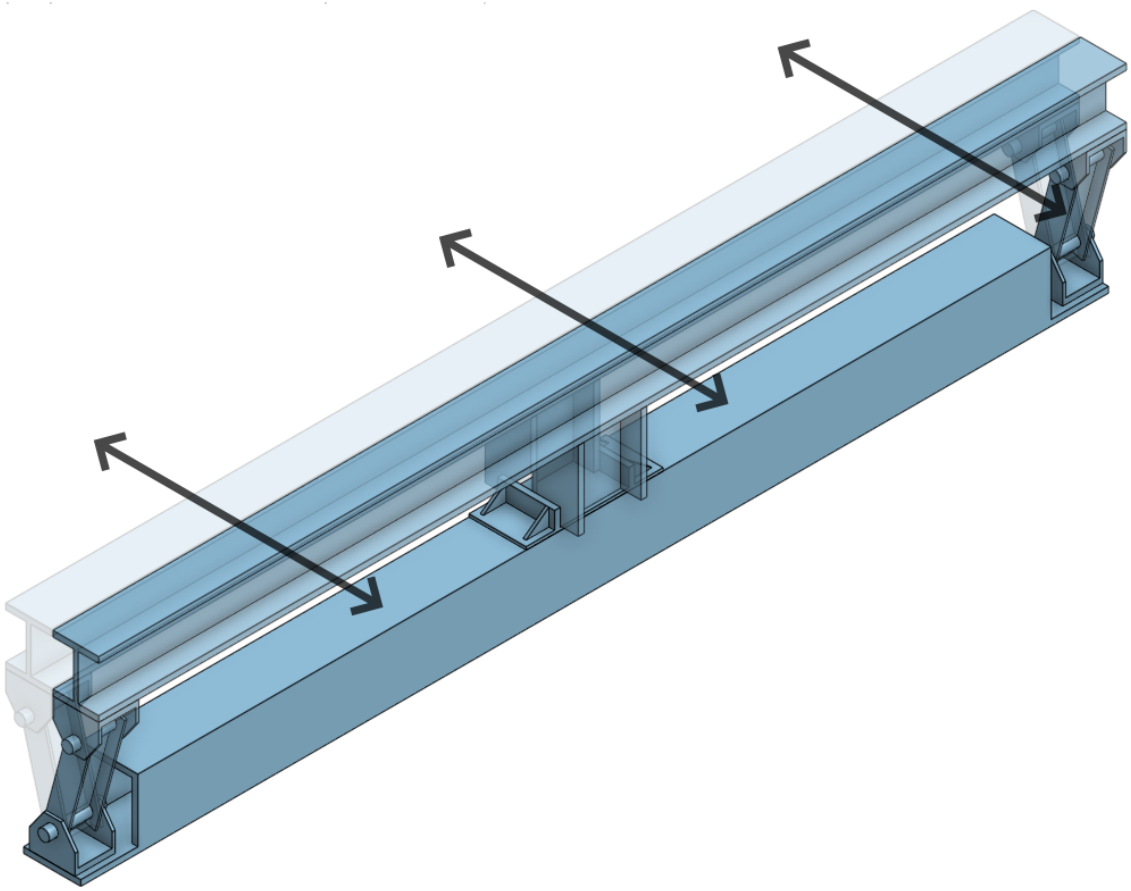


Figure 3.27: Anchor Bearings Behaviour Display

Both ends of the deck use a concept similar to the tower bearing constraints. The only difference is that instead of the bearing members connecting down to a girder, they connect to the anchor. However, this modelling is simplified to a single bearing element that combines the constraints' properties into a single bearing. As such, the ends of the deck are connected to the top of the anchor bearings while the bottom is restricted through a boundary condition (Section 3.6 for the boundary conditions). In the model, this implies that the constraint DeckAbutment[{S/N}] is a coupling constraint connecting Deck[{0/25}/{1/26}].set_end as master node to AnchorBearing[26].set_top as slave node. The master node is attributed to the deck member due to its higher stiffness. As per the tower bearing, the coupling constraint uses the kinematic type and constrains the U1, U2, U3, UR1, and UR2 degrees of freedom, leaving UR3 free. Therefore, two constraints are added to the model.

The last constraint creation to cover is the connection between the deck and the girder elements. A coupling constraint is used to constrain all degrees of freedom with a kinematic type. As the girder has the greatest stiffness, it is selected as the master set. Constraints are defined as Deck-ToGirder[{i}] through the use of Girder[{i}].set_ctr as the master node and Deck[{i-1/i}].set_end

and Deck[{i/i+1}].set_beg as slave nodes. Accordingly, 25 constraints are added, one at each junction between deck and girder elements.

3.6 Model Boundary Conditions Definition

The bridge model requires four sets of boundary conditions created in the Initial step to fix the model in space. Moreover, a fifth condition is added in the Pre-Displacement step only for the pre-displacement of the cable system. These conditions are described in this section, which follows the naming convention and the number of occurrences presented in Table 3.13, with notations matching those of Tables 3.1, 3.4, and 3.10 for the names of the parts, with the addition of {StoT/TtoT/TtoE}, which represents the three main spans: Start to Tower (side span), Tower to Tower (main span), and Tower to End (side span).

Table 3.13: Boundary Conditions Names and Count of Occurrences in the Model

Type of Constraint	Name	Count
Tower Bottom	TowerBottom[{E/W}][{1/2}]	4
Deck Abutment	DeckAbutment[{S/N}]	2
Main Cable End	MCAnchor[{E/W}][{S/N}][{n}][{m}]	48
Cable Support End	[{Y/Z}]SupportPin[{E/W}][{i}]	92
Pre-Displacement	PreDisp[{StoT/TtoT/TtoE}][{E/W}]	6
	Total:	152

First, the bottom of the towers of the LLRB are anchored to large concrete piers. These piers are not modelled in Abaqus, as they are considered fixed and assumed to have no significant impact on the bridge's response. On the contrary, these connections are replaced by fixed boundary conditions, which represent well how the bottoms of the towers are bolted to the piers. As such, a TowerBottom[{E/W}][{1/2}] would use the Tower[{E/W}][{1/2}].set_bot region with the Encastre condition, restricting the six degrees of freedom of the node: U1, U2, U3, UR1, UR2, and UR3.

Second, the ends of the deck must be anchored to the concrete landing through the anchor bearing, which was already discussed in Section 3.5. Thus, a DeckAbutment[{S/N}] condition would be used on the region AnchorBearing[{0/26}].set_bot by restricting the displacement and rotation of U1, U2, U3, UR1, and UR2, and leaving UR3 as free.

Third, the last spans of the main cable wires at each end of the bridge are anchored to the concrete landing. This connection, presented in Figure 3.28 [9] for the actual bridge, is simplified as a pin condition at the ends of the already defined main cable wires. It is assumed that the saddles at the anchor are sufficient to prevent considerable displacement in the cables and

are, therefore, not included in the model. A MCAnchor[{E/W}][{S/N}][{n}][{m}] pins the region MCWire[{E/W}][{0/25}/{1/26}][{n}{m}].set_{beg/end}, which restricts all displacement degrees of freedom, while leaving the rotational ones free.



Figure 3.28: Main Cable Wires Anchorage From the 2011 Visual Inspection [9]

Fourth, the cable supports connect to the cable connection centres. The other end of a cable support instance must then be pinned to prevent the global K matrix from being singular. Even if these boundary conditions do not exist in the real bridge, they do not impact the structural performance due to the cable supports' negligible axial rigidity (EA). Therefore, the [{Y/Z}]SupportPin[{E/W}][{i}] condition pins the {Y/Z}Support[{E/W}][{i}].set_pin node in space.

Finally, a boundary condition must be created in the Pre-Displacement step to engage the cable system before applying gravity in the Vertical Loading step. This boundary condition, PreDisp[{StoT/TtoT/TtoE}][{E/W}] is a displacement of -0.01 m in the Y direction, defined with a uniform distribution and a Ramp amplitude on the CableConnection[{E/W}][{3/13/23}].set_ctr region. This boundary condition is deactivated starting in the Vertical Loading step to avoid impacting the response.

3.7 Loading Generation and Definition

Two types of loads are defined within the model. The first is the gravity of -9.81 m/s^2 (Y direction) in the Vertical Loading step, created with a uniform distribution on the entire model. The default ramp amplitude is selected, as other alternatives did not produce more consistent results. The

gravity load considers the density of the defined materials with the cross-sectional area values of each profile/section in the model.

The second is the wind loading, which is applied as lift forces, drag forces, and overturning moments on the appropriate physical components in the model. As such, the lift, drag and moment are applied to the deck elements, whereas the towers and cables only have drag forces. This simplification for the towers and the cables reduces the complexity of the model by reducing the number of forces while still replicating the critical part of the loading: the drag force. The rest of this section explains how the wind loading is applied to each component in Abaqus. Since wind is not a static load, it should vary in time based on a wind velocity signal. This signal is defined in Abaqus as a Tabular Amplitude WindAmplitude in which the amplitudes are calculated as a scaled signal of the square of the wind velocity, as per the wind loading equations presented as Equations 2.22, 2.23, and 2.24.

$$A_i = \frac{U_i^2}{U_{max}^2} \tag{3.10}$$

Where A_i is the amplitude of the signal at time index i, U_i is the wind velocity (m/s) at time index i, and U_{max} is the maximum wind velocity (m/s) in the signal. As such, every wind load applied as part of the wind signal is defined as a fraction of the value associated with the maximum wind velocity. This method allows the definition of a single amplitude signal for all loads. An equivalent alternative would have been to define the amplitude signal as the wind velocity values. The wind signal used for this analysis is created with the script in Algorithm 3.1. The different steps relevant to the creation of the wind signal are represented visually in Appendix G for a typical execution of the algorithm. This algorithm starts by creating a Kaimal spectrum (Figure G.1) with the mean wind speed, the integral length scale, and the standard deviation of the wind speed fluctuations. A signal in the frequency domain (Figure G.4) is created by multiplying the discrete Fourier Transform of a generated random white noise signal (Figures G.2 and G.3) by the square root of the obtained spectrum. This signal considers white noise and the Kaimal spectrum to replicate wind behaviour. This signal can be converted to the time domain with an inverse discrete Fourier Transform (Figure G.5). While this version of the signal has the appropriate frequencies to simulate variations in wind, it does not simulate how the wind velocity changes over time. To avoid this limitation, a random signal with normal distribution is created, and a cumulative sum of the signal is applied at each element (walk), forming a smooth function for the variation in the mean wind speed (Figure G.6). The mean wind speed component and variation are added to the previously created wind signal, forming a signal in which the mean wind speed changes

over time (Figure G.7). Finally, turbulence in the signal is increased with a high-frequency noise added directly to the signal in the time domain (Figure G.8). All referenced figures are presented in Appendix G.

Algorithm 3.1 Wind Signal Generator

```python
import numpy as np

# Parameters for the wind signal
sampling_frequency = 4  # Hz (samples per second)
duration = 5 * 60  # seconds (5 minutes of data)
mean_wind_speed = 20  # m/s (initial mean wind velocity)
time = np.linspace(0, duration, int(sampling_frequency * duration))

# Define the Kaimal spectrum parameters with broader high-frequency range
L = 200  # Integral length scale (in metres)
sigma_u = 4  # Standard deviation of wind speed fluctuations

# Frequency domain representation
frequencies = np.fft.rfftfreq(len(time), 1/sampling_frequency)
frequencies[0] = 1e-6  # Avoid division by zero for the first frequency

# Define the Kaimal spectrum (in frequency domain)
def kaimal_spectrum(f, U_mean, L, sigma_u):
  return (4 * sigma_u**2 * L / U_mean) / (1 + 6 * f * L / U_mean)**(5/3)

# Generate the Kaimal spectrum
S_f = kaimal_spectrum(frequencies, mean_wind_speed, L, sigma_u)

# Generate white noise as the basis for the signal
white_noise = np.random.normal(0, 1, len(time))

# Apply the Kaimal spectrum to the white noise
wind_spectrum = np.fft.rfft(white_noise) * np.sqrt(S_f)

# Convert back to the time domain
wind_signal = np.fft.irfft(wind_spectrum)

# Add a random walk for mean wind speed variation (slow, long-term changes)
np.random.seed(5)  # For reproducibility
step_size = 0.05  # Control the magnitude of the random walk
mean_wind_speed_variation = np.cumsum(np.random.normal(0, step_size, len(time)))  # Random walk

# Add the mean wind speed and its random variation to the wind signal
wind_signal += mean_wind_speed + mean_wind_speed_variation

# Increase the turbulence by adding high-frequency noise
high_freq_noise = np.random.normal(0, 2.0, len(time))
wind_signal += high_freq_noise
time = [1 / sampling_frequency * round(t * sampling_frequency) for t in time]
```

The generated wind signal of 5 minutes is created with a sampling frequency of 4 Hz, so it should capture frequencies in the signal as high as 2 Hz as per the Nyquist frequency. The rest of

the parameters in the presented script are those used in generating the wind velocity signal given in Figure 3.29, with the frequency distribution with normalized magnitudes (normalized absolute with signal length of a Fast Fourier Transform) available in Figure 3.30.

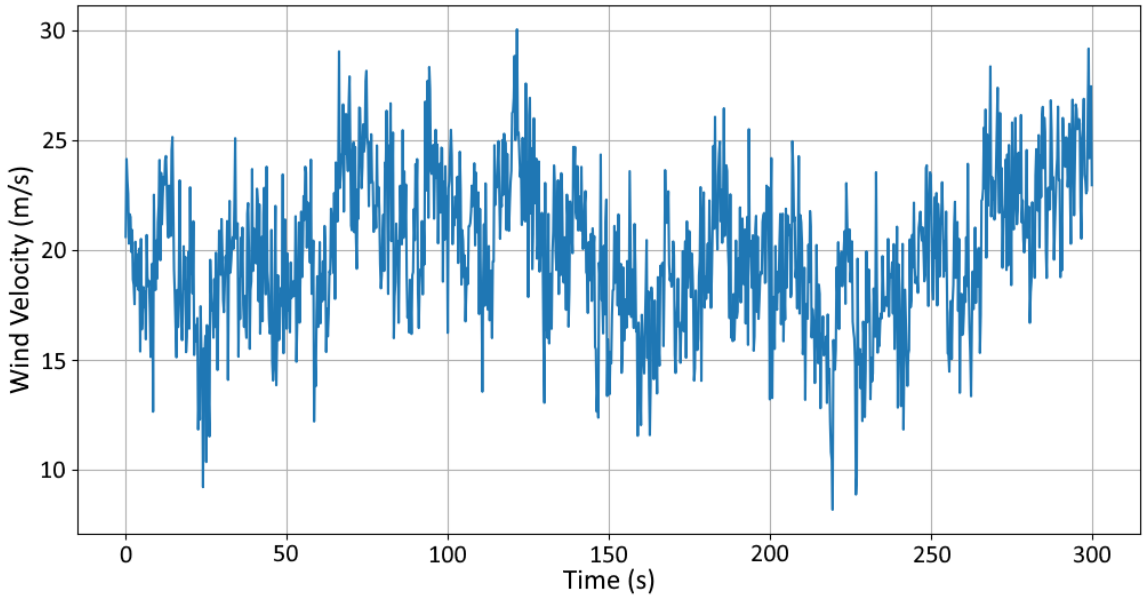


Figure 3.29: Complete Wind Speed Signal

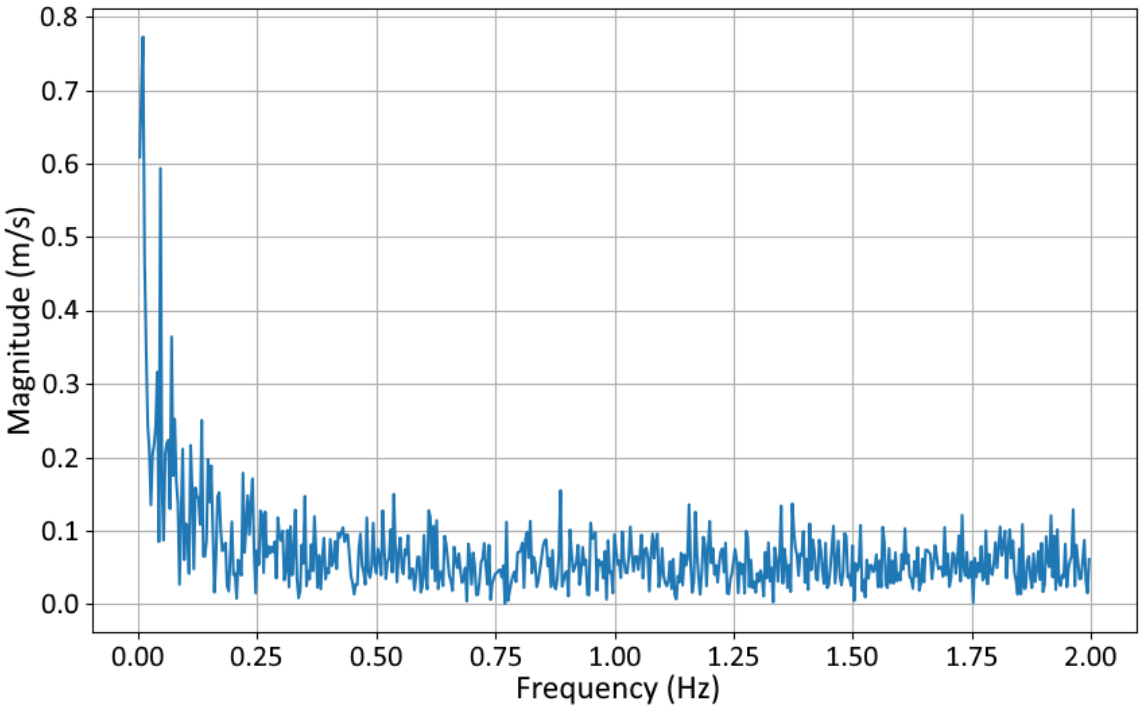


Figure 3.30: Wind Signal Frequencies Distribution

However, due to computing limitations, the longest signal used in this research is 8 seconds, which is 33 points in time. The isolated 8-second signal is extracted from a generated signal in

Figure 3.31, with the parameters from Algorithm 3.1. For the rest of this section, it should only be noted that the maximum wind velocity is 23.9949 m/s in the signal.

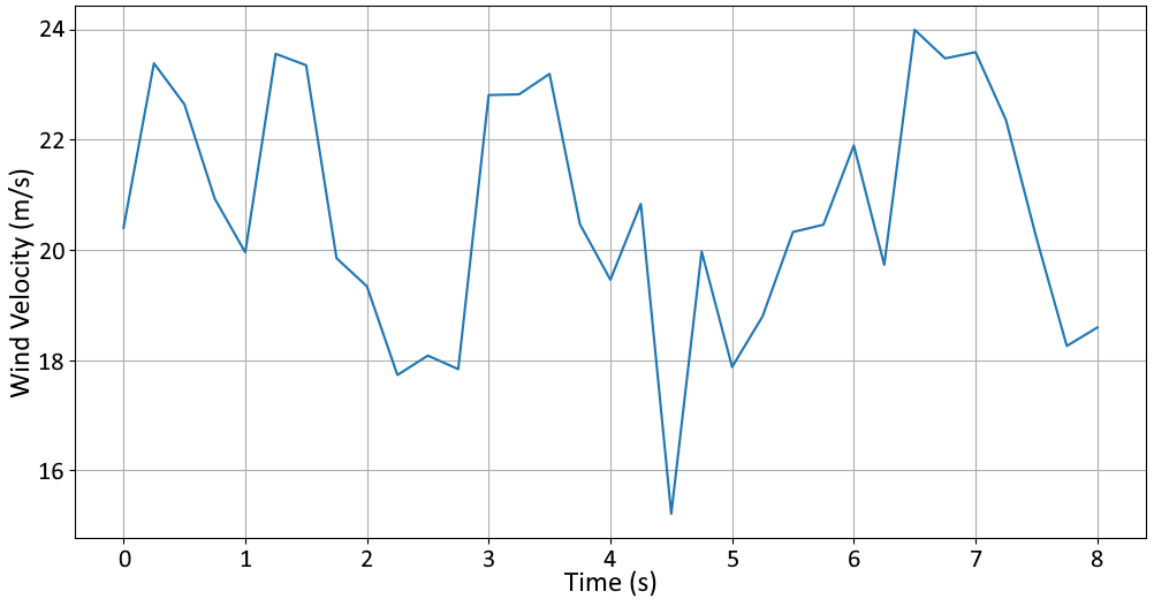


Figure 3.31: Short Wind Velocity Signal

Due to the lack of information from the LLRB regarding its aerodynamic properties, the values are estimated based on similar work. These values are not a true reflection of the properties of the bridge, which implies that the response to wind is not necessarily accurate. However, the objective in this section is to evaluate if the behaviour of the bridge changes under wind loading is applied on a bridge with damaged main cables. Even if the obtained values may be inaccurate, it is expected that the observed phenomenon remains proportional to the correct values, which allows for the identification of patterns. In reality, a wind tunnel analysis would be required to establish an accurate wind-loading experiment. First, the air density is assumed to be $1.225\,\mathrm{kg/m^3}$, and the following properties are considered in the wind loading calculations on the deck. These values are also presented in a summary in Table 3.15.

$$C_D = 1.6; C_L = 0.1; C_M = 0.05; B = 9.489\,m; H = 0.572\,m$$

Since no direct information is available regarding the aerodynamic coefficients of the LLRB, these values are approximated based on a different suspension bridge modelling study [43]. However, shorter suspension bridges are not typically the main focus of aerodynamic analyses, making the estimation of these parameters challenging in this research. Nevertheless, as explained in the next chapters, the objective of the dynamic wind loading analyses is to determine if the response of the bridge under wind loading varies based on the damage in the main cables. An additional

increase is applied to the drag coefficient and a decrease to the moment coefficient to account for the more rectangular cross-section of the LLRB's deck. It is important to emphasize that these values should only be considered rough approximations of the wind loading, as a more comprehensive assessment would require dedicated wind tunnel testing. Applying the wind loading equations (2.22, 2.23, and 2.24), the values for the deck can be obtained.

$$F_D = \frac{C_D \rho U^2 H}{2} = \frac{(1.6)\,(1.225\,kg/m^3)\,(23.9949\,m/s)^2\,(0.572\,m)}{2} = 322.745\,N/m$$

$$F_L = \frac{C_L \rho U^2 B}{2} = \frac{(0.1)\,(1.225\,kg/m^3)\,(23.9949\,m/s)^2\,(9.489\,m)}{2} = 334.636\,N/m$$

$$M = \frac{C_M \rho U^2 B^2}{2} = \frac{(0.05)\,(1.225\,kg/m^3)\,(23.9949\,m/s)^2\,(9.489\,m)^2}{2} = 1,587.65\,Nm/m$$

While the drag and the lift loads can be defined as Line Loads in Abaqus with the obtained values, there is no option to apply line moments. Therefore, the moments are applied at the deck element intersections as the sum of the moment over the tributary length on the deck.

Drag loading is applied to the vertical components of the towers with a drag coefficient of 2.05, as it resembles the shape of an I-beam, with a considered width of 0.8865 m. These parameters are used to determine the linear drag load on the towers, applied as a Line Load in Abaqus.

Next, drag loads are applied to the hangers, for which there are two types. The first type is for the hanger beam elements, which have a width of 0.18256 m and are considered to be in the shape of a plate, for a coefficient drag of 2.0, yielding the appropriate value to use as a Line Load in Abaqus.

The second type is for the regular hangers, which have a diameter of 0.04222 m and are in the shape of cylinders with a drag coefficient of 1.2. The resulting drag is separated equally between the top and bottom nodes of the hangers as a Concentrated Force in Abaqus, using the hangers' tributary length (half).

For the main cables, the fact that the wires are in a rectangular-shaped bundle complicates the determination of the wind parameters. However, a study by Liu [86] demonstrated that the combination of three wires in a series resulted in a cumulative drag coefficient of 1.5 for the center-to-center spacing to diameter ratio of 2.4, which is the case here. Using this value with the combined height of the cables of 0.1801 m, the linear drag force for the main cables can be obtained and applied as Concentrated Loads in Abaqus with the same principle as the hangers.

The loads in the model are named as per Table 3.14, and the summary of the coefficients and wind components are available in Table 3.15, with {E/W} representing either "E" or "W" for the east or west side of the bridge, {i} matching the hanger index starting at 0 at the first anchor and increases by 1 at every end of main cable span (up to 26), with index 6 and 20 being replaced by "T1" and "T2", {top/bot} representing either "top" or "bot" of the hangers, {beg/end} representing either the "beg" or the "end" of the main cable wires, {n} representing the column index of the wire in the parallel arrangement as A, B, or C, and {m} representing the row index of the wire in the parallel arrangement as 1, 2, 3, or 4. .

Table 3.14: Load Names

Type of Load	Name
Gravity	Gravity
Drag on the Deck	DeckDrag
Lift on the Deck	DeckLift
Moment on the Deck	DeckMoment[{i}]
Drag on the Tower	TowerDrag
Drag on Hanger Beams	HangerBeamDrag
Drag on Hangers	HangerDrag[{E/W}][{i}][{top/bot}]
Drag on Main Cables	MCWireDrag[{E/W}][{i}][{n}{m}][{beg/end}]

Table 3.15: Estimated Wind Loads Parameters and Linear Components

Components	C_D	C_L	C_M	H (m)	B (m)	F_D (N/m)	F_L(N/m)	M (Nm/m)
Deck	1.6	0.1	0.05	0.572	9.489	322.745	334.636	1,587.65
Tower	2.05	-	-	0.8865	-	640.880	-	-
Hanger Beam	2.0	-	-	0.18256	-	128.760	-	-
Hanger	1.2	-	-	0.04222	-	17.867	-	-
Main Cable	1.5	-	-	0.18011	-	95.274	-	-

3.8 Predefined Fields Modelling

As for the predefined fields, this feature creates pretension in the hangers and the main cables. Due to the nature of the structural drawings, only the expected final position of the bridge is available, meaning there is no information on the initial state of the cables. This is because the deck was initially made of timber and later replaced by the current concrete deck. As such, the final position of the cable system is presented in this entire section. However, the starting state of the bridge is considerably different, as the displacement in the cables and the deck is considerable when the weight of the system is added to the unstressed cables. To bypass this problem with the LLRB, pretension is integrated into the model through a temperature change between the Pre-Displacement and the Vertical Loading steps. Before presenting the temperatures, it is important

to note that these temperatures only influence the material's expansion and do not affect any other properties. Therefore, the temperature in the Initial and Pre-Displacement steps is set to 421.44 to 431.84 K for the main cable wires based on the pretensions ranging from 5,742.9 to 6,207.8 kN for the combined 12 wires. The hangers' starting temperatures range from 380.99 to 390.78 K for pretensions of 288.4 to 320.3 kN. Then, all temperatures are set to 293 K starting from the Vertical Loading step. The complete set of pretension and temperature values is presented in Table C.1 in Appendix C, which were obtained through a convergence algorithm explained in Section 4.3.2. The conversion between the obtained pretension and the starting temperature can be calculated through the linear thermal expansion relation in Equation 3.11 and Hooke's Law in Equation 3.12.

$$\delta = L\alpha\Delta t \tag{3.11}$$

Where δ is the change in length of the member (m), L is the length of the member (m), α is the coefficient of thermal expansion (m/mK), and Δt is the temperature change (K). Then, for a perfectly elastic material, Hooke's Law (3.12) can also be used for the temperature calculation in conjunction with the thermal expansion equation, as presented in Equation 3.13.

$$\sigma = E\epsilon; \frac{T}{A} = E\frac{\delta}{L} \tag{3.12}$$

$$t = t_0 - \frac{T}{EA\alpha} \tag{3.13}$$

Where σ is the stress (Pa), ϵ is the strain (m/m), E is the modulus of elasticity (Pa), T is the force in the member (N), and A is the cross-sectional area of the cable (m^2). For example, the first main cable wire, between location index 0 and 1, can get its starting temperature as presented below.

$$t = t_0 + \frac{T}{EA\alpha} = 293\,K + \frac{5,778,489\,N}{(200\times10^9\,Pa)\,(0.01911\,m^2)\,(11.7\times10^{-6}\,m/mK)} = 422.234\,K$$

The change from the higher temperature to the lower temperature causes the members to contract in the Vertical Loading phase as the gravity loading is applied. The combined effect of the loading, which elongates the wires, and the thermal contraction results in a minimal difference between the provided and the final coordinates. This is the basic idea behind the convergence criterion of the pretension algorithm. Furthermore, this method also ensures that the towers are

still vertical in the XY plane after the application of the gravity loads.

3.9 Summary

In general, this chapter demonstrated the modelling process used to generate the LLRB model. All required details to reproduce the finite element model are provided in this chapter, whereas the next chapter introduces the scripting component of the research, the model validation, and the definition of the damage test cases.

Chapter 4

Scripting and Bridge Model Automation

4.1 Chapter Overview

In Chapter 3, the development of the undamaged Lower Liard River Bridge (LLRB) finite element model was described, following the modelling steps implemented in the Graphical User Interface (GUI) of Abaqus. However, the primary objective of this research is to develop an automated process for generating finite element models of any standard suspension bridge designs in Abaqus using Python scripting. This tool will support further studies by facilitating the development of a structural response database, which will be used to train a machine learning algorithm for identifying structural damage in specific components. Since automation is the goal, manually creating a single model of the LLRB in Abaqus does not meaningfully contribute to this objective, as the process would need to be repeated for each bridge analyzed. Additionally, Abaqus is a finite element analysis software that does not specialize in structural design or analysis. This implies that for a large structural model with numerous parts, if a part's dimensions (the length and direction of wires) were to be modified, most of the model's geometry would need to be manually updated accordingly. An example of this issue that applies to this project is the determination of the deck elevations that were attempted through the convergence of a genetic algorithm. While the direct results of this process, described more thoroughly in Section 4.3.1, were not used in the final version of the bridge model, its results led to the use of a convergence algorithm for the pretension in the cables instead. Nevertheless, the iterations for the deck elevation convergence could never have been realized through a single model, as a single change in the coordinates of a member would have repercussions in numerous other instances. Therefore, the model would need to be manually adjusted for these changes for every iteration, which would not be realistically realizable and would be subject to human errors within such a repetitive task. Moreover, creating

and analyzing hundreds or thousands of damage cases is not a realistic task and would be highly vulnerable to human errors. These reasons are the foundation of the motivation behind the use of Python scripting for Abaqus instead of the use of the GUI.

As such, using scripting to automate the model creation opens the door to previously unattainable possibilities. First, the set of inputs can be generalized to a point where it can handle the creation of any suspension bridge with minimal effort. Second, the model can be readjusted anytime by changing the inputs without requiring additional work. Therefore, any miscalculations or inaccuracies identified retrospectively can be corrected without additional effort. Third, it allows for any sort of iterations or test cases to be performed on the model. This point is critical in realizing this current research, as described in this chapter.

This chapter describes how scripting is used within the current research to reach the results presented in Chapter 5. The explanations begin with the development of the Python program that is in charge of generating a single bridge model from specified inputs. This part covers the automation process performed for each step presented in Chapter 3. Thereafter, the model's tuning is described, including the iterations of the deck elevations to correct the tension within the cables and determine the cable pretension values. The explanations of the different test cases computed with the model are provided and how they are automatically created and submitted to run in Abaqus. Finally, the last section of this chapter covers the automatic output analysis of test cases.

4.2 Automatic Model Creation

This section describes the model's creation process through Python scripting and explains the program's structure. Inputs for the program are specified with an Excel spreadsheet template and a set of JSON files. These inputs and their implications are discussed in Section 4.2.1 referencing relevant sections in Chapter 3. The Excel inputs are processed into a standardized JSON file to transfer parameters from the program launcher to Abaqus. Subsequently, Abaqus is launched through a Windows Command Prompt command and a script is launched to create the model. After the model is created, the job in Abaqus is created and started, which marks the end of this section. The following section covers the output processing from the job results. Overall, the general steps performed by the program are presented in Figure 4.1, in which the key elements are identified.

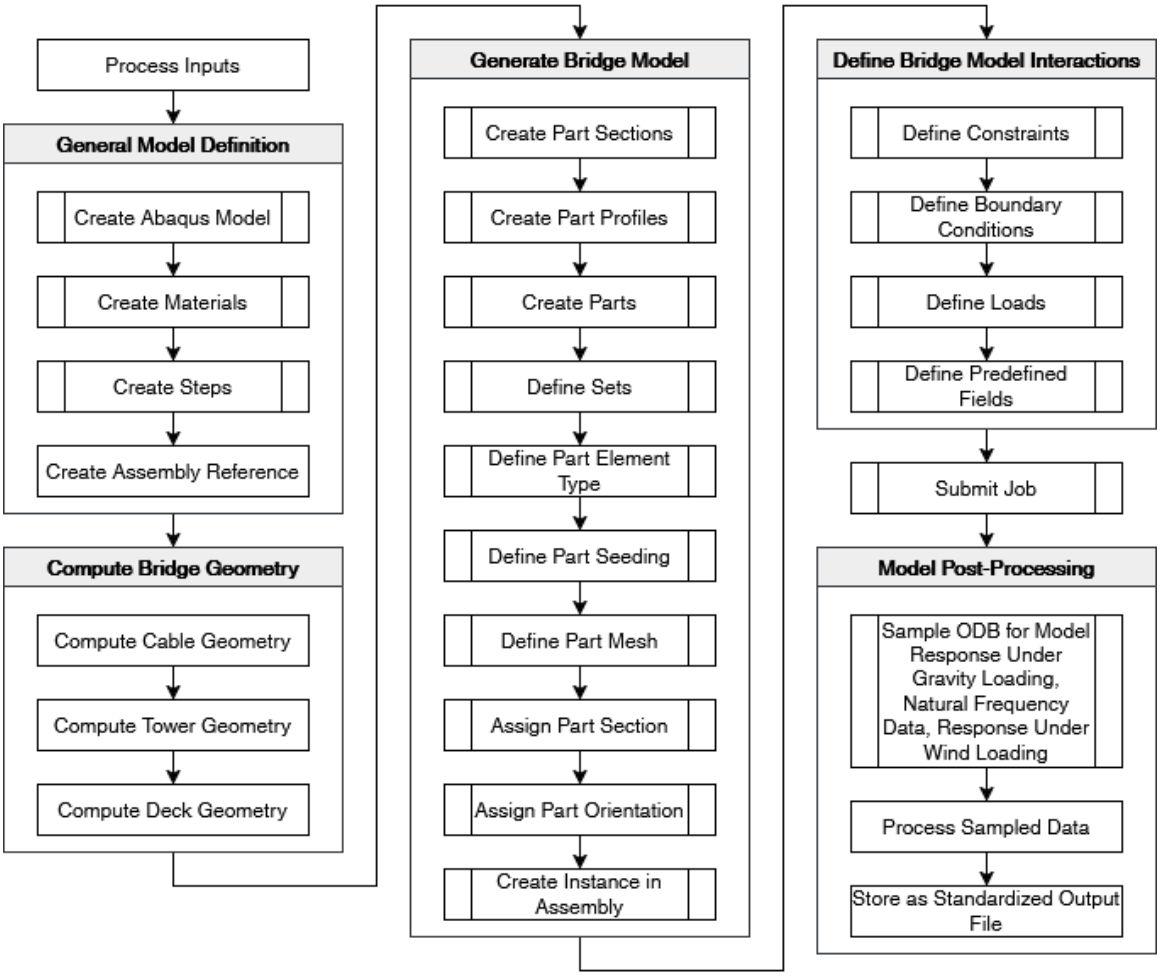


Figure 4.1: General Program Process

4.2.1 Inputs Description and Processing

For the program to have access to all required parameters to generate a suspension bridge model, all the inputs must be provided before launching it. This process is repetitive with many sections in Chapter 3, and should rather be used as a guide to use the program. For this reason, the inputs available for the program are presented in Appendix B, which covers the steps and the materials JSON parameters, as well as details on each sheet of the Excel input file. An additional section also covers the method used to parse and process the provided inputs, which explains how a standard general input file is generated at every program launch to facilitate communication with Abaqus.

4.2.2 Abaqus Launch

Once the inputs are processed, the command to launch Abaqus is sent, and a script to run is provided, resulting in the start command presented in Algorithm 4.1. This launches Abaqus with

the Graphical User Interface, but can be disabled by changing the keyword "script" to "noGUI" in the command.

Algorithm 4.1 Abaqus Launch Command

```
# Launches Abaqus through the command prompt in Windows with the starting script runModel.py
os.system("abaqus cae script=runModel.py")
```

4.2.3 Model Definition

With Abaqus opened, the script runModel.py is executed, which launches the creation of the model. The formation of the model is accomplished through a series of steps that handle specific parts of the assembly generation. These steps are presented throughout this section and include the following main items: the input deserialization; the creation of the blank model; the definition of the materials; the definition of the steps; the creation of the blank assembly; the geometric calculations for the parts creation; the parts and instances creation; the constraints definition; the boundary conditions definition; the loading definition; and the predefined fields definition. It should be noted that most steps have a complementary section in Chapter 3. Furthermore, longer sample codes demonstrating basic Python commands for Abaqus throughout this section are presented in Appendix D.

4.2.3.1 Program Files Organization

To properly understand how the program creates the correct command sequence to define the model, the organization of the code must first be explained. Scripts defined in this project are treated as modules, the foundation of packages. This program was created by writing two main packages: Bridge and AbaqusScripting. The AbaqusScripting package was written to be completely independent of the bridge modelling. Therefore, it covers every part of the program that is not bridge-specific. For example, this package could be used to facilitate modelling the framing system of a steel building. The package comprises three sub-packages: DefiningFunctions, Modules, and Utilities. The defining functions sub-package is composed of scripts that contain functions used to organize the creation of the model. These functions have a wide range of tasks, such as organizing the materials definition, the steps definition, the model definition, and the wires definition (parts). However, except for the model definition (as a blank model), this package has no interaction with Abaqus; it does not send any commands to Abaqus. This part is handled by the Modules (meant as Abaqus Modules) sub-package. This sub-package contains many class

definitions, which are used to represent direct components in Abaqus or to execute required commands. The different classes that are defined in the modules of this sub-package represent the assembly, the boundary conditions, the connectors, the constraints, the engineering features, the job, the loads, the materials, the meshes, the parts, the predefined fields, the profiles, the sections, and the steps in Abaqus. The adopted standard for writing these classes in this project is that the initialization of an instance creates an accessible object that represents the component within the program, parses the passed parameters for the component to be created, and then sends the command to Abaqus to be executed. Classes in these modules have all the information required to easily access the component in Abaqus, which can include, for example, the component's name. However, some components are handled by Abaqus in a way that makes them not easily accessible by their name. Therefore, the relevant classes also have a reference attribute that directly references the object in Abaqus, allowing access to that component in the later stages of the model creation. In general, the defining functions always initialize instances of classes defined in the Modules sub-package. Then, the last sub-package covers various utility functions and classes. This includes type verification functions, excel file parsing functions, a model saving function, JSON operations functions, custom errors, and many other utilities.

The other package defined for this project is the Bridge package, which includes everything related to a suspension bridge model. This package contains modules that handle all the logic for creating the bridge model. While the Modules sub-package of the AbaqusScripting package creates the commands for Abaqus based on provided parameters, the Bridge package is in charge of initializing the correct class instances from Modules with the appropriate parameters. Overall, the Bridge package uses the AbaqusScripting package to perform its tasks. The files organization is presented in Figure 4.2 for a more comprehensive view.

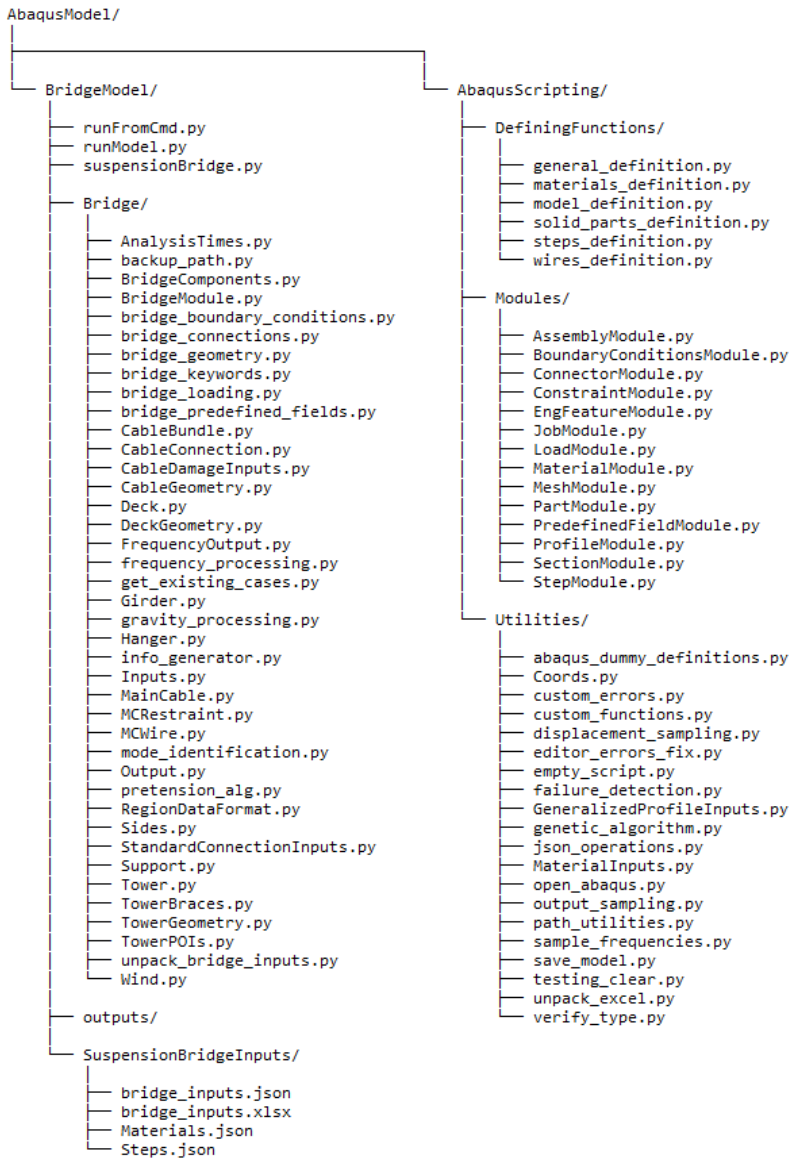


Figure 4.2: Program Files Organization

4.2.3.2 Inputs Deserialization

As mentioned in Section B.14, at this stage, the inputs are stored in the JSON file (bridge_-inputs.json), which can be deserialized into an instance of the class Inputs. This format makes the parameters easily accessible throughout the program as they are organized as class instances when necessary. It should be noted that this Inputs class does not consider the content of the Steps and Materials files, as these are treated independently. The idea behind this is that the content of Inputs is used throughout the entire program. In contrast, the content of Steps and Materials is used only in the initial steps of the model definition. Additionally, the definition of steps and materials is not specific to creating a bridge model, so they are part of the AbaqusScripting

package, which is handled in a different script. On the other hand, the Inputs class is specific to creating a suspension bridge. This would facilitate the adaptation of the program to, for example, a cable-stayed bridge by making an essential part of the model definition reusable.

4.2.3.3 Blank Model Definition

This section is part of the general definition of the model, which is common to any model creation and is therefore handled through the AbaqusScripting package. The commands to create a new model are presented in Algorithm 4.2. At the creation, the model is named model_name (a defined variable) and uses the standard explicit model type of Abaqus. STANDARD_EXPLICIT is a recognized keyword, and mdb is a recognized object in Abaqus scripting. Then, the model reference is obtained as a Model object from Abaqus, which is used throughout the rest of the program to send commands to the Abaqus model. Finally, the last command deletes the default model created by Abaqus. This completes the creation of the new blank model.

Algorithm 4.2 Blank Model Creation Commands

```
# Create the standard explicit model named through model_name
mdb.Model(modelType=STANDARD_EXPLICIT, name=model_name)
# Creates a reference to the created Abaqus Model object
model = mdb.models[model_name]
# Deletes the default model 'Model-1' from Abaqus
del mdb.models['Model-1']
```

4.2.3.4 Materials Definition

The next step is to create the materials located in the Materials JSON file, which is also part of the general definition of the model. These inputs are loaded into a Python dictionary, and each material entry is treated in a loop. The loop checks the material type provided and creates an instance of the appropriate child class of Material, defined in the Modules sub-package. There is currently only one child class of Material, ElasticMaterial, which is used to create elastic materials. The program was written very modularly to make it easily scalable for future work on either the bridge model or any other type of model. The initialization of an instance of ElasticMaterial calls a method that extracts the properties of the passed inputs dictionary and then creates the material with the commands presented in Algorithm D.1. One might recognize the model object from Section 4.2.3.3, which is used to send the command to create the material. Then, the Abaqus Material object reference is obtained to provide the material properties based on the extracted parameters.

4.2.3.5 Steps Definition

A similar process is used for the step inputs, as part of the general model definition. The Steps file is loaded into a Python dictionary, and the steps are sorted based on their "pre_step_name" value with a sorting algorithm. Then, a loop iterates through the ordered steps and creates an instance of a child class of Step, a class defined in the Modules package. The children classes available are StaticGeneralStep and FrequencyStep, which both have a similar process: extract the properties from the provided dictionary for the step and create the step. However, the StaticGeneralStep has additional methods to set other parameters, such as the time increments, the nlgeom setting, the stabilization options, and the adiabatic heating effects setting. For the StaticGeneralStep, the commands for the step creation are presented in Algorithm D.2, which also includes the definition of the additional parameters as an example for an automatic time incrementation method with the dissipated energy fraction stabilization. Naturally, in the actual modules, these commands are separated into organized methods based on the parameters, but this is meant as a simplified example of the commands performed. For example, different methods exist for every incrementation and stabilization method available.

The FrequencyStep class works similarly, but fewer parameters are available in the current program version. Due to the large number of parameters available in Abaqus for this type of step, the commands used for the current project are more streamlined. The commands used for this context are presented in Algorithm D.3.

4.2.3.6 Blank Assembly Definition

The last step in the general definition is creating the reference to the assembly. This is handled by an Assembly class in AbaqusScripting. The initialization of an instance of Assembly automatically runs the commands presented in Algorithm 4.3, which defines the reference to the Abaqus Assembly object and sets the coordinate system to Cartesian. Other methods of Assembly are later used to create new instances or translate existing instances.

Algorithm 4.3 Assembly Setup Commands

```
# Create a reference to the Abaqus Assembly object
assembly = model.rootAssembly
# Set the coordinate system of the assembly to Cartesian
assembly.DatumCsysByDefault(CARTESIAN)
```

4.2.3.7 Geometry Calculations and Parts Setup

Once the general definition of the model is completed, the geometric calculations and the organization of the parts may be accomplished. This task is handled by three classes, each with a sequence of methods performed to translate the parameters from the Inputs into a clear structure of part details. The geometry classes defined are CableGeometry, TowerGeometry, and DeckGeometry. Overall, each class comprises methods that create a data structure for each type of part. For CableGeometry, this includes determining the area and coordinates of every hanger, main cable wire, main cable restraint, and cable connection. For TowerGeometry, the coordinates of every vertical tower, girder, and brace part are obtained. Then, DeckGeometry calculates the coordinates of all deck, girder, and bearing parts. Each class creates a data structure for the parts that includes all the details necessary for their creation in the next step. The details of these calculations are not provided, but they follow the method described in Section 3.2.

4.2.3.8 Parts and Instances Definition

Once a clear structure is defined for each component, the parts can be created in Abaqus. This is accomplished in the class Bridge, which creates the profiles, the sections, the parts, the meshes, assigns the sections, assigns the beam orientations, and creates the instances. Loops are defined for each component type, which process the results from the three geometry classes (Section 4.2.3.7 above) and the parameters from Inputs.

To create the profiles, instances of the GeneralizedProfile class (child class of Profile) from AbaqusScripting are used. These instances utilize the command in Algorithm 4.4 to generate the profile based on the provided parameters.

Algorithm 4.4 Profile Creation Command

```
# Create a generalized profile in Abaqus and a reference to the Abaqus Profile object
profile_ref = model.GeneralizedProfile(
  name=name,
  area=profile_info.area,
  i11=profile_info.ix,
  i12=profile_info.ixy,
  i22=profile_info.iy,
  j=profile_info.j,
  gamma0=profile_info.gamma0,
  gammaW=profile_info.gammaW
)
```

The sections are created through instances of either TrussSection or BeamSectionBefore, which are both children classes of Section in AbaqusScripting. The use of either the truss or the beam class

depends on the type of part, as defined in Section 3.2. For the beam sections, the aforementioned class uses the before analysis integration in Abaqus (required for generalized profiles), as opposed to the BeamSectionDuring child class that uses the during analysis setting. The commands for creating a truss section are presented in Algorithm 4.5, while those for a beam section are in Algorithm 4.6. It should be noted that the beam section uses the Abaqus Profile reference obtained in Algorithm 4.4.

Algorithm 4.5 Truss Section Creation Commands

```
# Create the truss section in Abaqus
model.TrussSection(name=name, area=area, material=material)
# Gets a reference to the Abaqus Section Object
section_ref = model.sections[name]
```

Algorithm 4.6 Beam Section Creation Commands

```
# Create the beam section in Abaqus, uses material parameters from the previously defined material
model.BeamSection(
  integration=BEFORE_ANALYSIS,
  name=name,
  profile=profile,
  table=((young_mod, shear_mod), ),
  poissonRatio=poisson_ratio
)
# Gets a reference to the Abaqus Section Object
section_ref = model.sections[name]
# Sets the density parameter for the section
if density is not None:
  section_ref.setValues(density=density)
# Sets the shear_centre parameter for the section
if shear_centre is not None:
  section_ref.setValues(shearCenter=shear_centre)
```

Afterwards, the parts for each component type are created through instances of a child of the Part class: WirePart. This class uses the commands defined in Algorithm 4.7 to create a constrained sketch, create a part, create datums, create a wire polyline, and create the standard set set_alledges (Section 3.2). The presented version is simplified for a wire with only two datums, whereas the extended version in the program contains a loop that iterates through the provided additional coordinates to create the merged polyline.

Algorithm 4.7 Wire Part Creation Commands

```
# Create the empty constrained sketch
model.ConstrainedSketch(name="__profile__", sheetSize=sheetSize)
# Create the part in Abaqus
model.Part(dimensionality=THREE_D, name=name, type=DEFORMABLE_BODY)
# Create a reference to the Abaqus Part object
part_ref = model.parts[name]
# Create datum points for each coordinate - a loop in the real version
part_ref.DatumPointByCoordinate(coords=coords1)
part_ref.DatumPointByCoordinate(coords=coords2)
# Create a reference to the Abaqus Datum objects
datums = part_ref.datums
# Create the wire between the available datums
part_ref.WirePolyLine(
  points=((datums[1], datums[2]),),
  mergeType=mergeType,
  meshable=meshable
)
# Delete the created sketch
del model.sketches["__profile__"]
# Create a set containing all edges in the part
part_ref.Set(edges=part_ref.edges, name="set_alledges")
```

Based on the part, additional sets can be created as needed with the command in Algorithm D.4, which is taken from a Part method.

With the parts created, their mesh can be defined by seeding the parts, assigning the element types, and generating the mesh. This step is accomplished through either of the MeshDep child classes: MeshDepByValue and MeshDepByFactor. As mentioned in Sections 3.2 and 3.3, these classes are used for dependent meshes for this model. The mesh by factor is used for the tower components, whereas the mesh by value is used throughout the rest of the model (Section 3.2). The streamlined commands used to form a mesh are presented in Algorithm D.5 for a mesh by factor. Thereafter, Algorithm D.6 is used for a mesh by value, in which only the seeding is presented as the other commands are shared methods of the parent class MeshDep (no change from by factor).

Thereafter, the sections previously defined can be assigned to the parts through the command in Algorithm D.7. These lines are extracted from a WirePart method.

For the parts modelled as beams, the section orientation must be assigned with the command presented in Algorithm D.8, taken from a method in WirePart. The normal directions used are determined logically per part type from the coordinates system.

With the assignment of the beam orientations, all the parts are created with their required properties. From there, these parts can be added as instances to the assembly. As the parts were modelled with their actual coordinates in the final model, no translations or rotations are required; the instances should already be created at their correct location. An exception is for

the cable supports, which are discussed later in this section. For all other parts, the command in Algorithm 4.8 is used directly, which presents the content of a method of the Assembly class.

Algorithm 4.8 Instance Creation Command

```
# Add the instance to the assembly with a dependent mesh
assembly_ref.Instance(name=instance_name, part=part_ref, dependent=ON)
```

For the cable supports, all instances originate from the same part. The instances are also created through Algorithm 4.8, with the addition of the translation presented in Algorithm 4.9, which is also part of a method of Assembly.

Algorithm 4.9 Instance Translation Command

```
# Translate the instance by a specified vector
assembly.translate(
  instanceList=(instance_name, ),
  vector=translation_vector
)
```

4.2.3.9 Constraints Creation

From this point, all required instances should be defined in the assembly at their correct locations. The next step is to define the relationship between the instances through the constraints. The latter is defined through a series of functions in the module bridge_connections in the Bridge package. The logic of these functions represents the explanations from Section 3.5, accessing the sets through the instances in the assembly and creating the constraint with Algorithm D.9 and Algorithm D.10 for a coupling and a multipoint constraint, respectively. The former is executed with the Coupling class and the latter through the MultipointConstraint class, which are both defined as children class of Constraint. As per the explanations from Section 3.5, the multipoint constraint type is defined as beam by default for simplicity.

4.2.3.10 Boundary Conditions Creation

The boundary conditions are then defined similarly to the constraints, using functions from the module bridge_boundary_conditions in the Bridge package. The logic follows the details presented in Section 3.6, and the Abaqus commands are generated through the children classes of Boundary: Pinned (example in Algorithm D.11), Encastre (example in Algorithm D.12), and DisplacementBC (example in Algorithm D.13). As per their name, these classes represent pinned, encastre, and displacement boundary conditions, respectively.

4.2.3.11 Loading Definition and Application

Subsequently, the loading on the bridge can be defined with the module bridge_loading functions as part of the Bridge package. The first load considered is gravity, with commands defined in GravityLoad, a child class of Load in the Modules sub-package. The loading is created in Algorithm 4.10, in which the gravity_magnitude value is taken directly from the initially provided inputs.

Algorithm 4.10 Gravity Loading Creation Command

```
# Apply gravity loading on the entire model
model.Gravity(
  comp2=-gravity_magnitude,
  createStepName=step_name,
  distributionType=UNIFORM,
  field="",
  name=name
)
```

While gravity is considered in all the models discussed in this research, wind is only used in the wind analysis test suite. For these models, wind is applied as combinations of Load children class: ConcentratedLoad, MomentLoad, and LineLoad. The concentrated loads are used since truss members may not have a line loading due to their limited DOFs. As such, concentrated loading is used at the end of those members to mimic the linear load along their lengths. Algorithms 4.11, 4.12, and 4.13 present the commands creating the concentrated, moment, and line loads applied to predefined regions.

Algorithm 4.11 Concentrated Load Creation Command

```
# Apply a concentrated load to a region
self.model.ConcentratedForce(
  name=load_name,
  createStepName=step_name,
  region=region,
  cf1=cf1,
  cf2=cf2,
  cf3=cf3,
  distributionType=UNIFORM,
  field="",
  localCsys=None
)
```

Algorithm 4.12 Line Load Creation Command

```
# Apply a line load to a region
model.LineLoad(
  name=load_name,
  createStepName=step_name,
  region=self.region,
  comp1=comp1,
  comp2=comp2,
  comp3=comp3
)
```

Algorithm 4.13 Moment Load Creation Command

```
# Apply a moment load to a region
model.Moment(
  name=load_name,
  createStepName=step_name,
  region=region,
  cm1=cm1,
  cm2=cm2,
  cm3=cm3,
  distributionType=UNIFORM,
  field="",
  localCsys=None
)
```

4.2.3.12 Predefined Fields Creation

Finally, the last step in creating the suspension bridge model is the implementation of the predefined fields. If the pretension setting is activated in the initial inputs, a loop iterates over the hangers and the main cables and applies the temperature changes. As presented in Section 3.8, the temperature is defined in the Initial step at a high value, which is to be reduced in the Vertical Loading step to a base value to create the pretension in the cables. First, the required temperature difference between the initial and base temperatures can be obtained through the functions presented in Algorithm A.1 from Appendix A for hangers and main cable wires. Due to the handling method of the materials inputs independently from the bridge-specific inputs, the material properties for this calculation are not available at this stage. This is mainly to avoid passing the material properties as parameters throughout the bridge definition. Instead, the values are pulled from the Abaqus Material Objects. With the required temperature difference determined for each cable, the pretension can be applied through the Temperature class, a child of the PredefinedField class in the Modules sub-package. Algorithm D.14 presents the sequence used for every cable, assuming that temp_diff was previously defined through the proper aforementioned calculation functions. The base temperature is always considered to be 293 K, which should not affect the results.

4.2.4 Abaqus Job Creation and Submission

With the completion of the model creation, the job can be created. Depending on the Abaqus License and the device used, parallelization can be activated with multiple processors or with GPU acceleration for the calculations. This command is sent to Abaqus using the Job class in the Modules sub-package, for which a basic example with no parallelization is presented in Algorithm D.15, using default parameters. An assembly regeneration command is also invoked after defining the job, which is required for the job to consider the changes made in the assembly. The example then displays the job submission in Abaqus. These commands are performed directly on the Abaqus mdb object, instead of the model reference used for other tasks.

4.3 Model Refinement

As the only available structural drawings for the LLRB date back to 1943 and are incomplete, the deck elevation is not directly accessible. However, an appropriate estimation of the position of the members under dead load is made through interpolation and data from more recent years. The problem with having access to only the final (loaded) coordinates of the members in the bridge is that there is no direct method to obtain the initial coordinates (before loading). Since the model is created from the provided coordinates, as soon as the bridge is loaded in the Vertical Loading step, it experiences considerable deflections. This means that the members' final position differs from those initially specified. Two solutions are developed and discussed in this section to obtain the final position of the bridge as per the coordinates extrapolated from the drawings. The first option is the modification of the starting elevations along the deck to ensure that the bridge sags into the desired final position. This alternative is explored in Section 4.3.1, but convergence proves impossible with the available parameters. This conclusion led to using the second option, which uses pretension in the cables in their initial state, as presented in Section 4.3.2. This tension is then released simultaneously as the gravity loading is applied, making the deflections minimal by maintaining constant tension in the cables between each step. As such, the final coordinates of the bridge can be entered as inputs for the initial state of the bridge.

4.3.1 Deck Elevations Iterations

The first intuitive solution to prevent the bridge deck from deflecting below the desired final position is to increase the initial elevations of the deck members while maintaining the individual lengths of the hangers. As the loads are applied on the elevated deck, it should naturally sag into

a lower position. With the correct initial coordinates, it would be expected that the final position of the deck could be reached following this approach. The first step for this naive approach is to define a set of deck elevations (along the length of the bridge) that serves as the objective to reach. The difference between the final position of the deck and the objective position defines the error for this algorithm. The first trial to achieve the objective set of values is a simple interpolation of the initial hanger elevations based on the final deck elevations. For this, Equation 4.1 calculates the next set of inputs for the hangers' top and bottom Y coordinates.

$$\overrightarrow{Y}_{i+1} = \overrightarrow{Y}_i - \overrightarrow{\eta}_i \odot \left(\overrightarrow{D}_i - \overrightarrow{D}_{obj}\right); \eta_{i_j} = \begin{cases} \frac{Y_{i_j} - Y_{i-1_j}}{D_{i_j} - D_{i-1_j}}, & for\ D_{i_j} - D_{i-1_j} \neq 0 \\ 0, & otherwise \end{cases} \tag{4.1}$$

Where $\overrightarrow{Y}_i$ is the $n \times 1$ vector of Y coordinates for the hangers for the iteration i (m), with n being the number of hangers, $\overrightarrow{\eta}_i$ is the $n \times 1$ vector of slopes for the iteration i, $\overrightarrow{D}_i$ is the $n \times 1$ vector of final deck elevations for the iteration i (m), $\overrightarrow{D}_{obj}$ is the $n \times 1$ vector of objective deck elevations (m), and $\odot$ is the Hadamard product operation (element-wise product). The elevation at the bottom at the towers is taken as the average of the two adjacent hangers, whereas the top is fixed. From two starting iterations, this algorithm may be applied to determine the next set of inputs to try, until $\overrightarrow{Y}_{i+1} \approx \overrightarrow{Y}_i$, which indicates the convergence to a solution. However, this algorithm converges very slowly and is greatly affected by bad initial guesses for the two starting iterations. Therefore, an alternative approach was considered, which instead determines the best root candidate of a polynomial fit on all previous iterations. Tests were performed with second-degree polynomials, but it was found that a simple linear regression is a better fit. Equation 4.2 presents the algorithm used as an improved version of Equation 4.1. This new algorithm uses the function get_best_poly_root presented in Algorithm E.1. The latter determines the more suitable root if it is within the already tested bounds (the minimum and maximum inputs tested). If the maximum degree specified is greater than one, then more than a single valid root can be obtained. In such cases, the best root is selected by absolute proximity to the average existing inputs. This is done so that the algorithm would give more weight to solutions closer to what has already been tested in edge cases. Then, if no valid roots are found between the previously tested range, the function returns new bounds to test instead of a root. This is implemented to avoid testing solutions far from the tested range, which may cause the model to abort. Instead, the bounds are expanded conservatively to ensure the model computes more consistently.

$$\overrightarrow{Y}_{i+1} = \overrightarrow{R}_{i+1}; R_{i+1_j} = get_best_poly_root\left(\left(\overrightarrow{Y}_{k_j}\right)_{k=0}^{i}, \left(\overrightarrow{D}_{k_j} - \overrightarrow{D}_{obj_j}\right)_{k=0}^{i}\right) \tag{4.2}$$

Where $\overrightarrow{R}_i$ is the $n \times 1$ vector of most suitable roots for the hangers for the iteration i (m) and *get_best_poly_root* is the function defined in Appendix E as Algorithm E.1. This algorithm converges to a solution much faster than the previous interpolation method.

After running this algorithm and reaching convergence, it was determined that while the final position of the deck matched the target, the tensions in the hangers were very far from those expected from the structural drawings and the field measurements from the 2018 study. These differences can be observed in Figure 4.3, in which the tensions in the hangers are presented as points along the deck, along with the average expected tensions from the drawings and the field tensions measured in 2018. The field tensions are represented as the average of the four hangers, which are theoretically the same due to the double symmetry of the bridge. The field tensions were obtained by determining each hanger's first few natural frequencies through cable excitation and measurements from a laser vibrometer. Then, assuming an ideal string under tension, the natural frequencies should form a harmonic series with each frequency as an integer multiple of the fundamental frequency. This fundamental frequency is estimated as the slope of the first few natural frequencies against their mode number through a simple linear regression. With the fundamental frequency obtained, Equation 4.3 can be used to estimate the tension in the tested cable.

$$f_n = \frac{n}{2L}\sqrt{\frac{T}{\mu}} \tag{4.3}$$

Where f_n is the nth natural frequency of the wire (Hz), L is the unconstrained length of the wire (m), T is the tension in the wire (N), and μ is the linear mass of the wire (kg/m). However, the obtained natural frequencies of the hangers may not be disclosed as part of the non-disclosure agreement. As such, the Y axis scale is hidden on the figure to protect exact values while still presenting the disparity between the model and the field tensions. Nonetheless, the tensions are normalized so that the origin of the figure is at the bottom of the Y axis. It can be noticed that the gaps and missing tensions match the locations of the towers and the hanger locations that use beams instead of wires. A consistency check with a different modulus of elasticity is also provided to ensure that a different value for an older bridge does not cause the issue. Further sensitivity analysis are performed on the model to ensure that the changes in the tension values are not due to a sensitivity issue. After extensive variations in the tested material properties and section geometries, it is determined that the obtained tensions are accurate and not the result of a sensitivity issue.

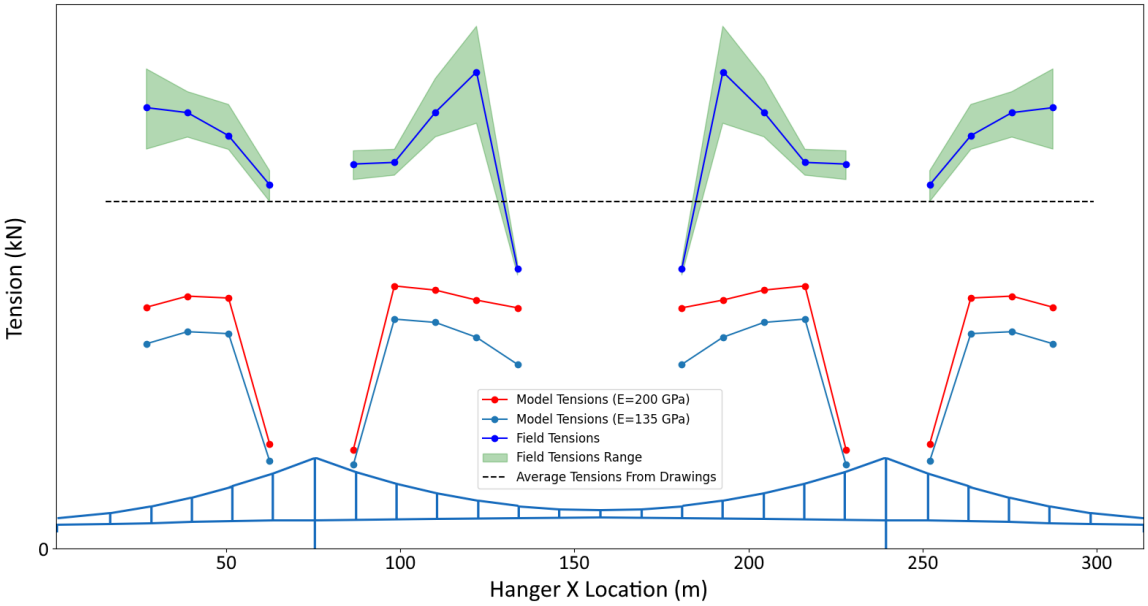


Figure 4.3: Hanger Tension Comparison - Root Finding Algorithm

The cause of the considerable error in the tensions is that the shape of the deck is extremely sensitive, and the load path is very easily disturbed. For example, a slight increase in the elevation of the tower-adjacent deck members causes a considerable increase in the load transferred to the towers, decreasing the load in the hangers. This is because the top of a hanger needs to sag significantly to experience a tensile force, at which point the load from the deck would already pass through the towers. After some preliminary testing, it was found that manual readjustments of the deck elevation at critical locations can have a considerable impact on the tension in the hangers. At this stage, it was determined that the previously developed algorithms would not be sufficient to solve for the tensions because each hanger location is treated independently. This implies that the deck elevations at the hanger locations that use beams instead of wires could never be solved, and the same applies to the deck elevations at the tower locations. As the deck elevation at the tower locations is one of the most critical for tension distribution, a different approach is needed to minimize the error in tension to match the average field values by changing the deck's initial elevations. Naturally, convergence could never be achieved manually due to the large amount of inputs and outputs. Moreover, each input impacts all outputs, meaning that solving the problem directly or with a conventional algorithm is impossible. A genetic algorithm has been developed specifically to address this problem. A genetic algorithm is a form of unsupervised machine learning that uses the concept of natural selection to find the optimal solution to a problem (Section 2.6). In this problem, an individual is defined as a set of deck elevations along half of the length of the bridge, and the inverse of the residual sum of squares of hanger tensions between the model and

the field values defines the fitness score of an individual. Therefore, an individual with significant differences in tensions compared to field values would have a low fitness score since the residual sum of squares would be high. The genes for the individuals represent only the first half of the length of the bridge, and values are mirrored on the other side to maintain symmetry. The score calculation is presented in Equation 4.4.

$$S = \left(\sum_{i=1}^{n} (F_i - M_i)^2 \right)^{-1} \tag{4.4}$$

Where S is the fitness score of an individual (N^{-2}), F_i is the measured tension of the hanger in the field i (N), and M_i is the model tension of the hanger i for the individual (N). Therefore, the algorithm's objective is to increase the fitness score by matching the tensions in the model to the target tensions from the field. To do so, the bottom elevation of the deck members must be changed until the objective is met. This process is done with the principle of natural selection; the strongest individuals (high fitness score) in a generation can reproduce and pass their genes to new individuals (children). Over multiple generations, individuals get stronger as only the best genes are passed to new individuals. The defined algorithm for this purpose is presented in Appendix F as Algorithm F.1, which uses the classes and functions defined in Algorithm F.2. The created function "create_individuals" typically performs the following steps to produce a new set of genes:

1. Tournament selection: performs k tournaments of size *tournsize* and returns the best genes found in each tournament.

2. Mating: performs mating between tournament winners, exchanging some genes between individuals.

3. Mutating: performs mutations in the tournament winners by modifying some genes.

4. Individual selection: randomly selecting one of the k newly generated individuals.

A tournament randomly selects several individuals in the existing population and keeps the strongest individual by comparing their fitness scores. The selected individuals from the rounds of tournaments are then matched in pairs to mate. Each pair has a *mate_prob* probability to mate, and if a pair mates, each gene has a *mate_gene_prob* probability to be exchanged between the two individuals. Then, each individual has a *mutate_prob* probability of mutating, and if it does mutate, each gene has a *mutate_gene_prob* probability of mutating. A gene mutation is simply a change to the existing gene by a value generated with a Gaussian distribution based

on the gene's upper and lower bounds. The previous operations then create a set of individuals that would form the next generation. However, due to the complexity associated with determining an individual's fitness score (running the entire model and extracting the hanger tensions), the function only returns a new random individual to be added to the generation. For this problem, the first generation is created with an individual who uses the values obtained with the deck elevations linear algorithm previously defined in Equation 4.2. The second individual in this generation is then generated randomly within the specified bounds, and the regular steps are then used to create individuals and add them to the current generation. Approximately 800 individuals (model iterations) are added to the first generation, at which point an elitism selection is applied, keeping only the best 250 individuals based on their fitness score. Since the first generation is generated mainly through random processes, this elimination ensures that the weaker individuals do not pass on their genes. Then, these 250 individuals are used to create the next generation, which is also composed of 250 individuals. To offer context, this test sequence involved approximately 400 hours of computation time.

If a solution to the problem exists, this solving sequence should progressively approach the correct elevations while continuously improving the fitness score of new individuals. However, after a few generations, it was noticed that the fitness score was barely improving, meaning that the tensions were generally not getting much closer to the targets. Figure 4.4 presents the evolution of the best fitness scores between generations, which demonstrates that an accurate solution would not be achievable.

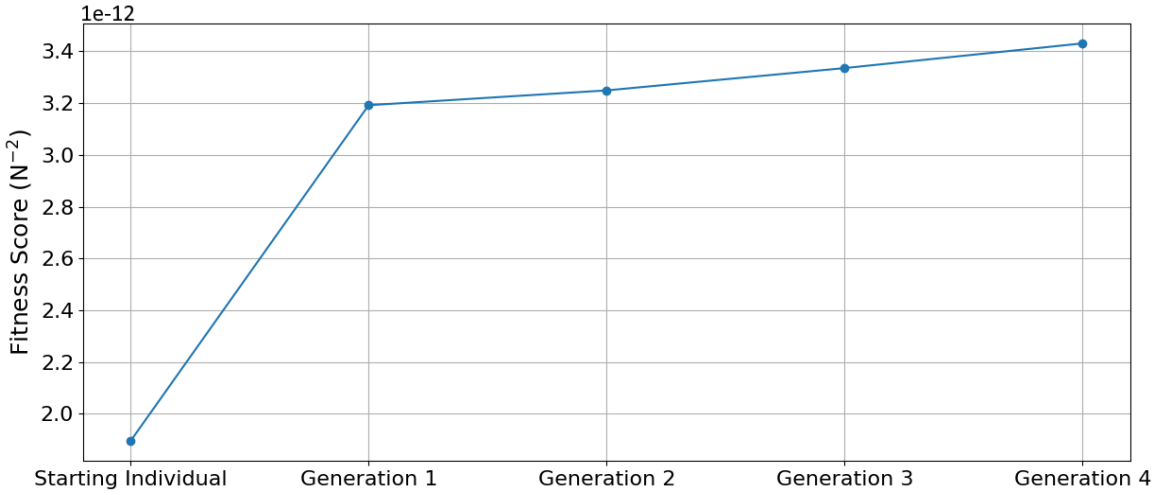


Figure 4.4: Fitness Score Evolution per Generation

For context, if all 18 hangers have an error in tension of 1,000 N, the fitness score would be $5.556 \text{x} 10^{-8}$ N^{-2}.

$$S = \left(\sum_{i=1}^{18} (1,000\,N)^2 \right)^{-1} = 5.556 \times 10^{-8}\,N^{-2}$$

As such, it would not be reasonable to assume that a solution of this precision exists for the model. Moreover, the tensions obtained in the field are not obtainable in the model, as the algorithm would greatly reward genes that increase tension in the hangers. The fact that no individual gets close to the target tensions implies that they are not feasible with the available parameters. This is an expected conclusion since the field tensions exceed the expected average tensions by approximately 50 kN on average. Overall, the best five individuals are presented in Figure 4.5, demonstrating that convergence is highly unlikely.

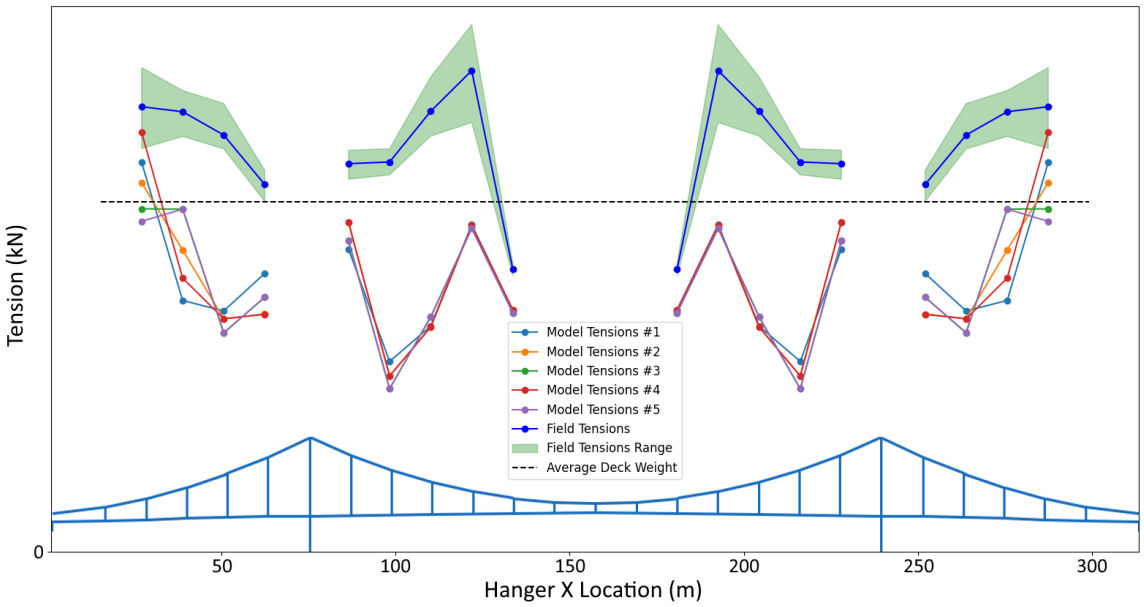


Figure 4.5: Hanger Tension Comparison - Genetic Algorithm

4.3.2 Cable Pretension Iterations

After demonstrating that the deck elevations could not be modified to have the correct final shape while maintaining accurate hanger tensions, the following solution is to apply an initial pretension in the cables, which is then released as the gravity loading is applied. Naturally, these pretensions do not represent real pretension from the bridge construction; instead, they are just a modelling technique used to match the geometry in the structural drawings. The objective of this method is to provide the final tension in cables as the pretension, such that the change from pretension to gravity-induced tension has little to no impact on the location of the cables. This solution must then be applied to the main cables and hangers, ensuring that the final geometry is respected. On

the other hand, tower and deck elements should experience minimal deflections as their stiffness is much greater than the wires. This means their deflection should have a minimal impact on the deck elevation after applying the gravity loading. An iterative process is used to obtain the final tensions in the cables, which uses the previous final tensions as pretensions for the next iteration. Applying the subsequent pretension works better for faster convergence if the average of the last k iterations is considered instead of only the previous iteration. This principle is presented in Equation 4.5.

$$\overrightarrow{P}_i = \overrightarrow{T}_i; T_{i_j} = \begin{cases} \sum_{h=i-k+1}^{i} \frac{\tau_{j_h}}{k} & for\, i \geq k-1 \\ \sum_{h=0}^{i} \frac{\tau_{j_h}}{i} & otherwise \end{cases} \tag{4.5}$$

Where $\overrightarrow{P}_i$ is the $n \times 1$ vector for the pretension of iteration i (N), with n being the number of cables (hangers and main cables combined), $\overrightarrow{T}_i$ is the $n \times 1$ vector of the average tension in the cables in previous iterations (N), τ_{j_i} is the tension in the cable j in iteration i (N) and k is an arbitrary number of previous cases to average for the next iteration. The pretension for the first iteration (iteration 0) is set to zero for every cable, and the baseline value for k is 10 but increases by 1 for each consecutive previous failed case. As such, if a case aborts computation, the subsequent trial should be with a k value of 11, and so on. This feature is added to avoid perpetual aborting cases. Per the previously presented algorithms, double symmetry is enforced for every iteration for all bridge sides. Moreover, the target deck elevations are taken from the structural drawings with a polynomial fitting for missing values, which are presented in Section 3.2.2. The application of pretension is more thoroughly explained in Section 3.8. Still, the global principle is that the members are created at a very high temperature, which is then corrected to an ambient temperature during the Vertical Loading step. This temperature change decreases the length of the members as the loads are applied, resulting in a net change in tension of approximately zero in the wires. This means that the coordinates of the members are the same in their final position as in their initial position.

Convergence of the algorithm is reached when the pretension inputs and the final tensions are approximately identical in an iteration. After letting this algorithm run for 100 iterations, the maximum tension difference between an input and an output is 1.25 N for the hangers and 5.20 N for the combined 12 wires of the main cables. Overall, these iterations converge to the values presented in Figure 4.6, which match the average tensions expected from the structural drawings. The target deck elevations are generally maintained throughout the analysis, and the final tension values converge automatically to the expected tensions from the original design. Moreover, the

total tension in the main cable wires is proportional to the angle of the cables (Figure 4.7, which is another indication of the method's success. This simple method, which only required the use of Equation 4.5, is used to obtain the final bridge properties to use as inputs for the model.

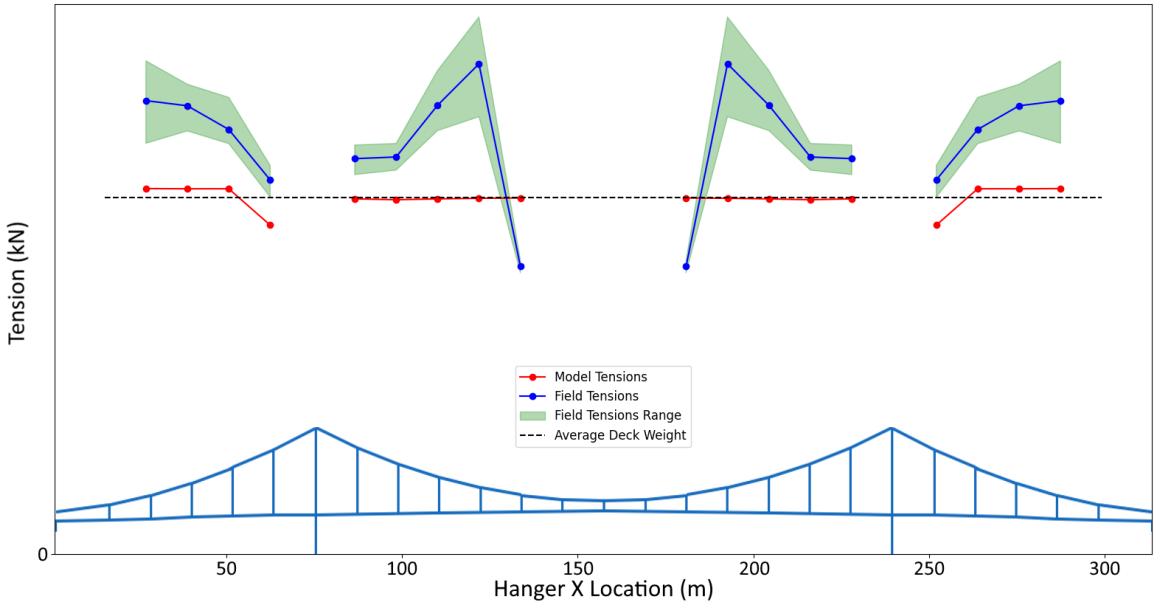


Figure 4.6: Hanger Tension Comparison - Pretension Algorithm

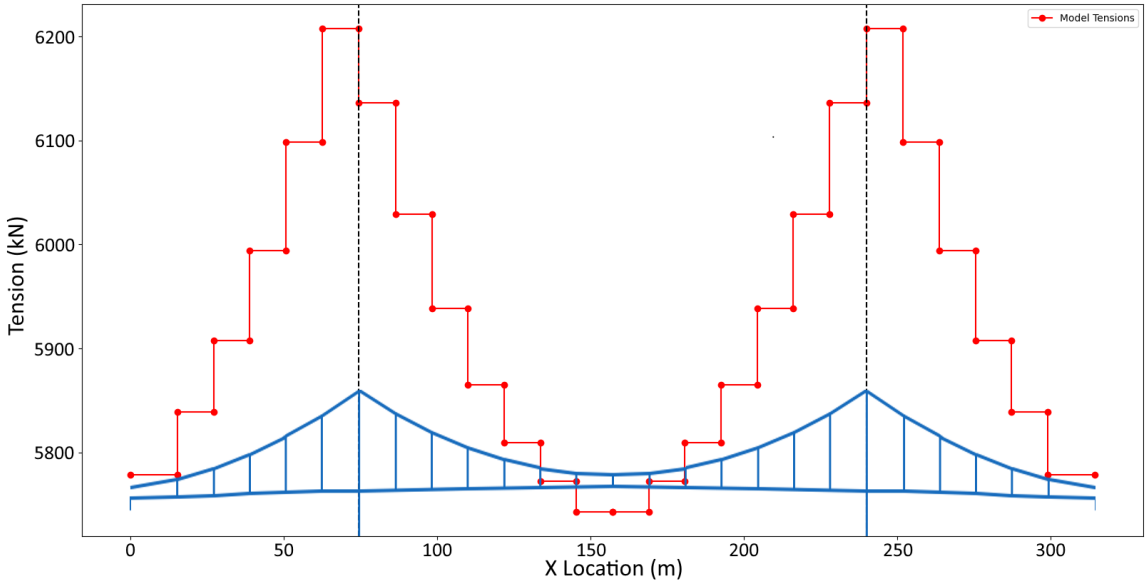


Figure 4.7: Main Cable Tensions - Pretension Algorithm

Overall, the comparison with the expected tensions from the structural drawings is the only available means of validation. The hanger tensions in the latest field study would be expected to fall within the same range as these theoretical values. The observed differences are likely due

to either significant measurement errors or critical structural issues in the LLRB related to load distribution. Regardless, the undamaged model should be validated using the undamaged values from the drawings.

4.4 Test Cases Definition

With the model now capable of being generated using the automated tool developed, the damage test cases to be performed must be defined. As previously mentioned, these test cases consider damage only in the main cable segments, aligning with the objective of this research. These different test series can be separated into three main test series, which cover a wide range of damage patterns applied to the model. These test series are enumerated below and explained in more detail in the following sections.

1. FREQ11: Frequency and dead load static response analysis of the simplified 1x1 pattern main cable wire bridge model with uniform damage patterns.

2. FREQ34: Frequency and dead load static response analysis of the complete 3x4 pattern main cable wire bridge model with uniform damage patterns.

3. WIND11: Dynamic wind analysis of 1x1 pattern main cable wire bridge model with uniform damage patterns.

4.4.1 FREQ11 Test Series

This first test series uses the simplified version of the bridge model with a single main cable wire that has the combined properties of the 12 individual wires of the actual bridge. This model is generated with different damage cases, and for each case, the static general and the frequency analyses are performed. These yield static displacement results under regular gravity loading and natural frequency results. Therefore, the required final displacements and natural frequencies are obtained and logged for each damage pattern using the automatic output sampling method described in Section 4.5. Each test case in this series has parameters to identify the location of the damage on the main cables. These parameters indicate how many consecutive spans are affected by the damage and the index of the location of the first main cable span damaged. These indications are explicitly defined for the bridge's southeast quarter, including the main cable spans from the first anchor to the midspan on the east side. To test different patterns of symmetry and asymmetry about the other quarters, different location patterns are defined, which indicate

whether the damage is mirrored about each axis. Finally, the damage intensity is specified as a fraction of the missing cross-sectional area of the affected cables.

1. WiresNumber - Number of Damaged Wires: Number of consecutive main cable spans damaged [“Sing”, “Doub”, “Quad”, “All_”].

2. WireIndex - Index of Damaged Wires: Index of the first damaged main cable spans [“01”, “02”, “03”, ...].

3. LocationPattern - Location Pattern: Pattern of the damage locations along and accross the bridge [“Solo”, “XSym”, “ZSym”, “DSym”, “OSol”, “ASym”].

4. DamageIntensity - Damage Intensity: Intensity of the damage in the wires involved [0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.50, 0.70, 0.90].

More specifically, the number of damaged wires WiresNumber is the number of damaged consecutive main cable spans. If a single span, also referred to as segment, is affected, the notation “Sing” is used. “Doub” is used for two spans, while “Quad” denotes four spans. A special case is defined for which all the spans are damaged. These indications only concern the spans from the first anchor to the midspan on the bridge’s east side. The baseline undamaged model takes the notation “Base” for this parameter. Throughout these cases, damage is typically located in consecutive spans when the case type allows it (i.e. “Doub” is always two consecutive main cable spans). This mimics some patterns observed in the damage reported in the 2011 visual inspection [9]. Overall, damage cases can be assembled through Figure 4.8, for which each path represents a case performed for the FREQ11 test series.

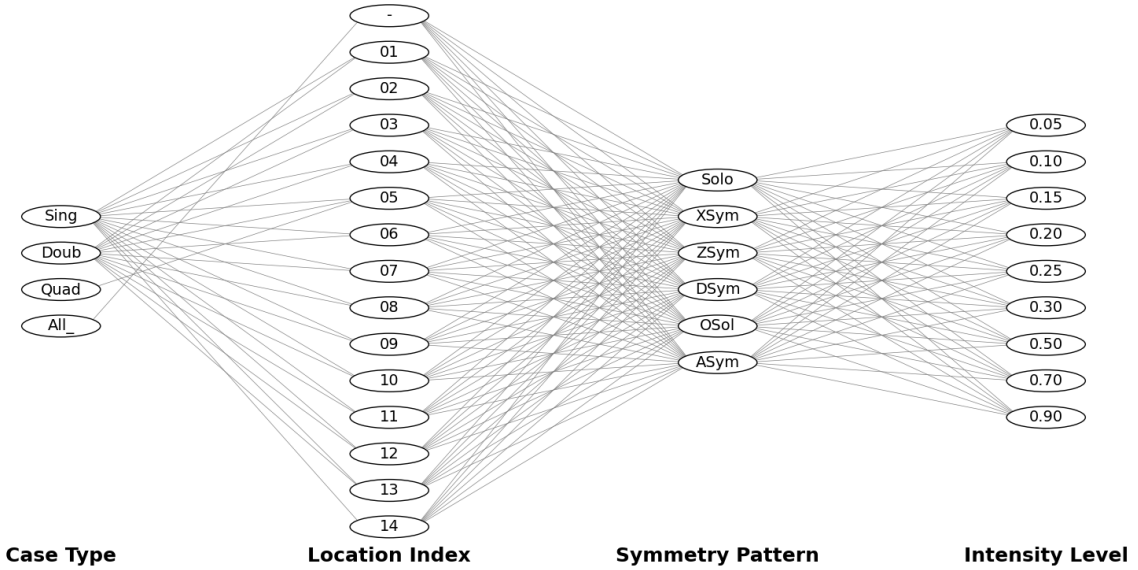


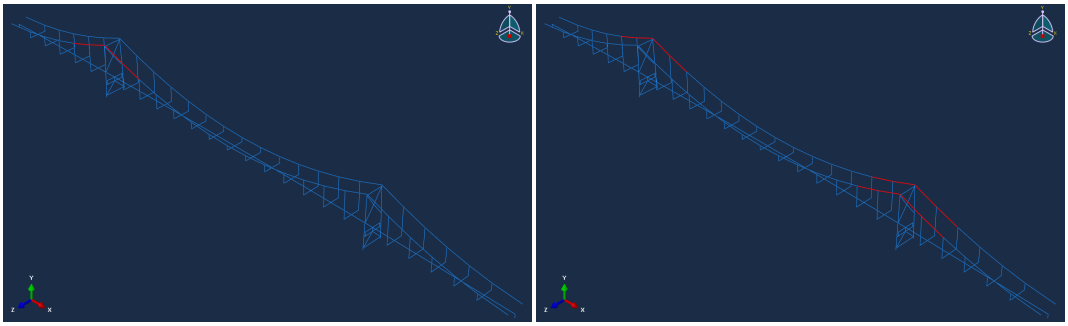
Figure 4.8: Damage Cases Notation

With the number of main cables affected, the location index of the damage is specified as an index representing the lowest main cable span index that sustained damage, WireIndex. In the case of the "Sing" cases, this value can then range from "01" to "13", 13 being the span preceding the midspan. On the other hand, "Doub" cases have WireIndex values in the "01" to "12" range since specifying an index of 13 would cause damage on spans 13 and 14, therefore damaging a different quarter. "All_" and "Base" cases have no WireIndex due to their nature and are marked with "___" to represent it.

The location patterns must be used to apply damage to other quarters. These patterns allow damage to be applied to a single quarter, two quarters in any pattern, three quarters, or all quarters. The "Base" case is here marked with "_____". The way these patterns are handled in terms of damaged quarters is presented below, along with the "Quad" "05" cases in Figure 4.9.

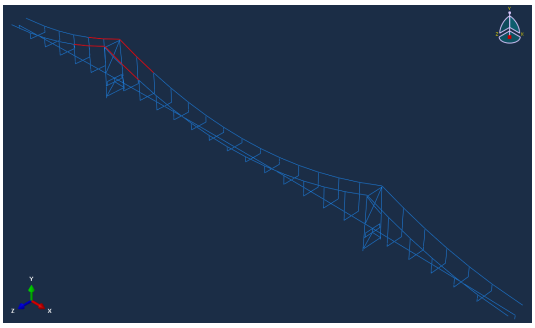
- Solo - One damaged quarter: south-east
- XSym - Symmetry about the X axis with two damaged quarters: south-east and south-west
- ZSym - Symmetry about the Z axis with two damaged quarters: south-east and north-east
- DSym - Double symmetry with four damaged quarters: south-east, south-west, north-east, and north-west
- OSol - Opposite of the Solo case with three damaged quarters: south-west, north-east, and north-west

- ASym - Symmetry about a diagonal with two damaged quarters: south-east and north-west

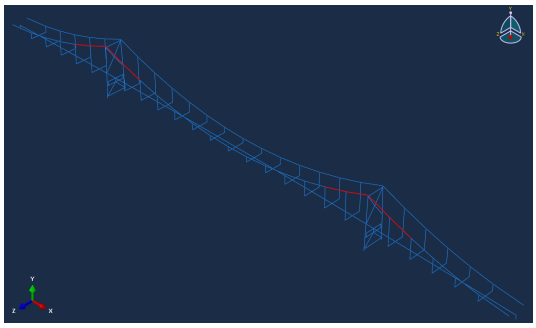


(a) Solo Pattern

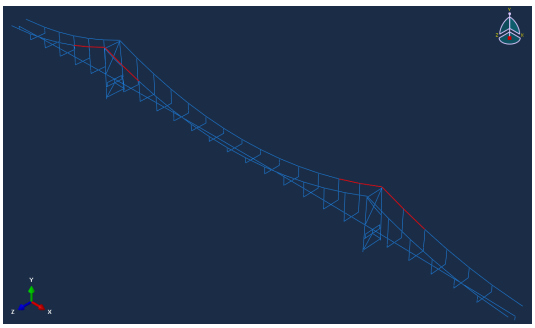
(b) OSol Pattern



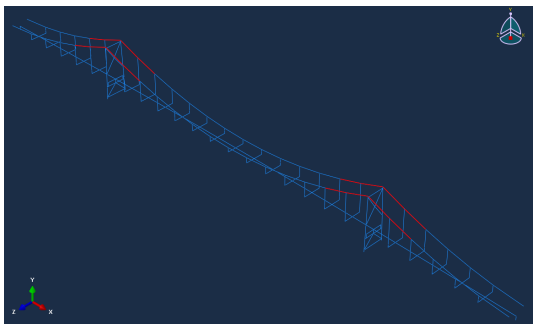
(c) XSym Pattern



(d) ZSym Pattern



(e) ASym Pattern



(f) DSym Pattern

Figure 4.9: Damage Symmetry Patterns

Finally, the damage level in the affected wires is specified as a cross-sectional area reduction, denoted as a damaged fraction of the entire wire, a float with two decimals. As mentioned in Section 2.3, no reduction of ductility is considered, since it is assumed that the remaining cables are unaffected by corrosion and that the affected ones have already failed. The damaged levels in this test series are in increments of 0.05 from 0.05 to 0.30, then in increments of 0.20 from 0.30 to 0.90. These damage levels are applied as cross-sectional reduction over the entire length of the cable segment, which can be converted to an equivalent damage level that considers the effective length of loss of stiffness due to the cable restraints, explained in Appendix H. It should

be mentioned that the damage levels tested extend over reasonable cross-sectional area reductions in the cables. From a simple estimation, it is expected that the bridge would fail if any main cable segments would reach the damage level of 0.70. Still, these test cases could be useful in the determination of global patterns in the relationship between applied damage and bridge properties. Overall, this means all WiresNumber and WireIndex combinations have 54 cases (nine damages and six patterns). These cases are presented in Table 4.1, which illustrates the damage location by main span index by case. This table demonstrates that a full sweep of all available indices is conducted for the "Sing" and "Doub" cases, while the "Quad" cases are only applied around the tower, specifically between indices 6 and 7. This creates a total of 1459 test cases for the FREQ11 test series.

Table 4.1: FREQ11 Test Cases Enumeration

WiresNumber	WireIndex	DamageIntensity (Count)	Patterns (Count)	Cases (Count)	Case Name Template	1	2	3	4	5	6	7	8	9	10	11	12	13
Base	0	1	1	1	[Base][____][__][____]													
All_	0	9	6	54	[All_][LocationPattern][__][DamageIntensity]	■	■	■	■	■	■	■	■	■	■	■	■	■
Sing	1	9	6	54	[Sing][LocationPattern][01][DamageIntensity]	■												
Sing	2	9	6	54	[Sing][LocationPattern][02][DamageIntensity]		■											
Sing	3	9	6	54	[Sing][LocationPattern][03][DamageIntensity]			■										
Sing	4	9	6	54	[Sing][LocationPattern][04][DamageIntensity]				■									
Sing	5	9	6	54	[Sing][LocationPattern][05][DamageIntensity]					■								
Sing	6	9	6	54	[Sing][LocationPattern][06][DamageIntensity]						■							
Sing	7	9	6	54	[Sing][LocationPattern][07][DamageIntensity]							■						
Sing	8	9	6	54	[Sing][LocationPattern][08][DamageIntensity]								■					
Sing	9	9	6	54	[Sing][LocationPattern][09][DamageIntensity]									■				
Sing	10	9	6	54	[Sing][LocationPattern][10][DamageIntensity]										■			
Sing	11	9	6	54	[Sing][LocationPattern][11][DamageIntensity]											■		
Sing	12	9	6	54	[Sing][LocationPattern][12][DamageIntensity]												■	
Sing	13	9	6	54	[Sing][LocationPattern][13][DamageIntensity]													■
Doub	1	9	6	54	[Doub][LocationPattern][01][DamageIntensity]	■	■											
Doub	2	9	6	54	[Doub][LocationPattern][02][DamageIntensity]		■	■										
Doub	3	9	6	54	[Doub][LocationPattern][03][DamageIntensity]			■	■									
Doub	4	9	6	54	[Doub][LocationPattern][04][DamageIntensity]				■	■								
Doub	5	9	6	54	[Doub][LocationPattern][05][DamageIntensity]					■	■							
Doub	6	9	6	54	[Doub][LocationPattern][06][DamageIntensity]						■	■						
Doub	7	9	6	54	[Doub][LocationPattern][07][DamageIntensity]							■	■					
Doub	8	9	6	54	[Doub][LocationPattern][08][DamageIntensity]								■	■				
Doub	9	9	6	54	[Doub][LocationPattern][09][DamageIntensity]									■	■			
Doub	10	9	6	54	[Doub][LocationPattern][10][DamageIntensity]										■	■		
Doub	11	9	6	54	[Doub][LocationPattern][11][DamageIntensity]											■	■	
Doub	12	9	6	54	[Doub][LocationPattern][12][DamageIntensity]												■	■
Quad	5	9	6	54	[Quad][LocationPattern][05][DamageIntensity]					■	■	■						

4.4.2 FREQ34 Test Series

The FREQ34 test series is similar to the FREQ11 series; however, it uses the 3×4 main cable wire pattern with 12 individual wires. It follows the same notation as FREQ11 since the damage is applied uniformly to all wires of the main cables. However, the 3x4 main cables create much longer computing times due to the increased number of elements in the model. As such, only the most critically impacted locations are tested with this setup, namely on the spans at the anchor, the tower and the bridge midspan. These selected cases are presented in Table 4.2, for a total of 595 additional cases.

Table 4.2: FREQ34 Test Cases Enumeration

WiresNumber	WireIndex	DamageIntensity (Count)	Patterns (Count)	Cases (Count)	Case Name Template	1	2	3	4	5	6	7	8	9	10	11	12	13
Base	0	1	1	1	[Base][____][__][____]													
All_	0	9	6	54	[All_][LocationPattern][__][DamageIntensity]	■	■	■	■	■	■	■	■	■	■	■	■	■
Sing	1	9	6	54	[Sing][LocationPattern][01][DamageIntensity]	■												
Sing	6	9	6	54	[Sing][LocationPattern][06][DamageIntensity]						■							
Sing	7	9	6	54	[Sing][LocationPattern][07][DamageIntensity]							■						
Sing	13	9	6	54	[Sing][LocationPattern][13][DamageIntensity]													■
Doub	1	9	6	54	[Doub][LocationPattern][01][DamageIntensity]	■	■											
Doub	5	9	6	54	[Doub][LocationPattern][05][DamageIntensity]					■	■							
Doub	6	9	6	54	[Doub][LocationPattern][06][DamageIntensity]						■	■						
Doub	7	9	6	54	[Doub][LocationPattern][07][DamageIntensity]							■	■					
Doub	12	9	6	54	[Doub][LocationPattern][12][DamageIntensity]												■	■
Quad	5	9	6	54	[Quad][LocationPattern][05][DamageIntensity]					■	■	■	■					

4.4.3 WIND11 Test Series

The WIND11 test series is far more limited than the natural frequency analyses due to the computing time of the model. Therefore, a more streamlined approach is used in which only the Doub damage cases are used with the same location index selected in FREQ34. Moreover, only the DSym symmetry pattern and the 0.30 damage level are considered, taking the total number of cases down to 6. Table 4.3 is a depiction of the selected cases for the WIND11 test series.

Table 4.3: WIND11 Test Cases Enumeration

WiresNumber	WireIndex	DamageIntensity (Count)	Patterns (Count)	Cases (Count)	Case Name Template	1	2	3	4	5	6	7	8	9	10	11	12	13
Base	0	1	1	1	[Base][____][__][____]													
Doub	1	1	1	1	[Doub][DSym][01][0.30]	■	■											
Doub	5	1	1	1	[Doub][DSym][05][0.30]					■	■							
Doub	6	1	1	1	[Doub][DSym][06][0.30]						■	■						
Doub	7	1	1	1	[Doub][DSym][07][0.30]							■	■					
Doub	12	1	1	1	[Doub][DSym][12][0.30]												■	■

4.4.4 Tests Summary and Discussion

Overall, the different test series defined in this section have a variety of objectives based on their test cases. First, the central analysis revolves around FREQ11, a simplified version of the model that allows the computation of a high number of cases, covering a wider range of simulations. This series aims to demonstrate whether the location and the intensity of damage in the main cable have a quantifiable effect on the natural frequencies associated with the vibration modes of the deck. Compared to the most probable actual damage on the LLRB, the Sing, Doub, and Quad have a low distribution in their location. However, this series aims to establish a proof of concept and to determine if a parametrization of the bridge by damage location would eventually be possible through machine learning. To initiate this relation, multiple symmetry patterns are defined to cover a broader range of cases and to study the impact of disrupting the symmetry in the bridge.

For the test case types (WiresNumber), the Sing cases are defined to enable the parametrization of individual damage locations. On the other hand, the Doub cases exist to examine the effect of the combination of a low number of damage locations. In this type, damage locations are always in consecutive spans in a bridge quarter to replicate the typical damage pattern observed in the LLRB, where cable damage tends to concentrate in specific areas. Moreover, this considerably reduces the number of cases to be computed; enforcing the damage to be in two consecutive spans and accounting for symmetry yields 13 location dispositions. Removing this consecutiveness condition would bring the equivalent number of location patterns to 351, making it unreasonable for this study. Therefore, consecutive spans are selected in conjunction with the various symmetry patterns. Furthermore, an extensive range of damage levels is considered to provide redundancy and to notice irregularities in the results.

In the FREQ34 test series, limited test cases are defined to compare the 1x1 and 3x4 model variations. By simulating only critical test cases, it should be determined if the added complexity of the 3x4 model significantly impacts the quality of the results or if the 1x1 model is sufficient.

Throughout both the FREQ11 and the FREQ34 analysis, a static analysis under a static dead load is provided. This is done to determine if symmetric damage cases can be differentiated based on the vertical displacement of the deck since it is one of the limitations of the SHM related to natural frequencies.

Finally, the WIND11 was created to determine whether the deck's rotation and lateral displacement under wind loading could be used to identify damage locations in the main cables.

4.5 Output Automatic Analysis

To automate the process of analyzing the test cases, an output database (odb) analysis script is created to fetch the required measurements and store them in a standardized output JSON file. Also, a simple mode identification algorithm has been developed to analyze the natural frequencies. This section briefly presents these two components, which are required to automate the analysis of a large number of cases.

4.5.1 Output JSON Object

This standardized output file has been designed to contain all relevant values for the test case analysis. This includes base coordinates and displacements of selected components, tensions and pretensions of all cables, and detailed frequency summaries. The selected members for which

the beginning and end base coordinates and displacements are saved are the girders, the vertical towers, and the hangers. These items are saved as lists of three floats, one for each dimension (x, y, z). Then, the main cables and hangers' pretensions and final tensions are saved as a single float per cable. For the main cable with 12 wires, this is the sum of the 12 tensions found in the main cable.

The next item is the frequency objects, which include the first n (50 in this study) natural frequencies. For each frequency, the maximum displacement in the model in each dimension and the absolute maximum displacement in the model are provided (X, Y, Z, and MAX), along with the equivalent four values for rotations, all as floats. The effective masses for all six degrees of freedom are provided along with their participation factors as floats. Additionally, the frequency in Hz is saved as a float, the mode number as an integer, the name of the identified mode as a string or null, the type of the mode (Vertical, Lateral, Rotational, or null) as a string or null, the symmetry of the identified mode (Symmetric, Asymmetric, or null) as a string or null, and the index of the identified mode as an integer or null. Furthermore, the deck displacements for the mode are represented as a list of six floats (UX, UY, UZ, RX, RY, RZ) for each deck element. For ease of use, an identified_modes object also exists in the output file, which has an enumeration of the modes of each identified mode for all combinations of type and symmetry. Then, a float is stored directly in the output object as max_frequency.

The last item is used when the wind is activated, and the base coordinates and displacements of the girders are recorded throughout the wind loading step (a single set of translations for each time step). It is stored the same way as the girders item but with lists of lists of floats for the coordinates instead of lists of floats (to account for the time dimension).

4.5.2 Mode Identification Algorithm

While Abaqus solves the model to obtain the natural frequencies, it is not specialized for structural analysis and, therefore, does not inherently identify the vibration modes types. Thus, performing an analysis of over 2,000 cases of the LLRB model becomes impossible if the target is the comparison of the deck's vibration modes. This prompts the creation of an algorithm that can identify the type of modes and their symmetry. Performing this task with a perfect success rate of identification is far outside the scope of this research. Still, this could be completed using machine learning to train a specialized model adaptable to any suspension bridge. However, for this project, a simple identification algorithm can be created with a high enough success ratio for the desired level of analysis. The algorithm is performed in two steps, the first one being the identification of the

mode type (Vertical, Lateral, Torsional or None), followed by the determination of the symmetry of the mode if a deck vibration is identified.

First, for the deck modes of vibration, the deck is expected to have high displacements proportional to the maximum displacement in the entire model. In more detail, this implies that for a vertical mode, the deck's maximum vertical displacement (Y axis) should be considerable compared to the absolute maximum displacement in the complete model. The same applies to the lateral displacement (Z axis) for lateral modes and to the rotation about the deck length when compared to the absolute maximum rotation in the model for torsional modes. To establish a calibration of the model, the ratio of maximum displacements in the deck over those in the rest of the model can sampled for the baseline case (undamaged model). This can be followed by a visual analysis of the first 50 vibration modes in Abaqus, which allows the comparison of the obtained ratios for the different modes. The ratio results are then compared between the identified modes and the rest of the modes. Taking the example of the vertical modes, the factor associated with each mode is calculated as the maximum vertical displacement in the deck divided by the absolute maximum displacement in the model. These factors are categorized based on whether the associated mode has manually been identified as a vertical mode. For the lateral modes, the factors are instead calculated with the maximum lateral displacement in the deck. In contrast, the torsional modes are calculated with the maximum rotation of the deck about its length divided by the maximum absolute rotation in the model. The categorization of these factors allows for the determination of thresholds that can identify if a mode exhibits the behaviour of a sought deck vibration mode. The results from this analysis are presented in Figure 4.10, and demonstrate that the minimum factor of the identified modes is always greater than the maximum factor of the rest of the modes. Therefore, if a vertical deck displacement factor is greater than 0.4, it should be considered a vertical mode. Then, if a lateral deck displacement factor is greater than 0.19, it should be considered a lateral mode. Finally, if a rotational (about the X axis) deck displacement factor is greater than 0.07, it should be considered a torsional mode.

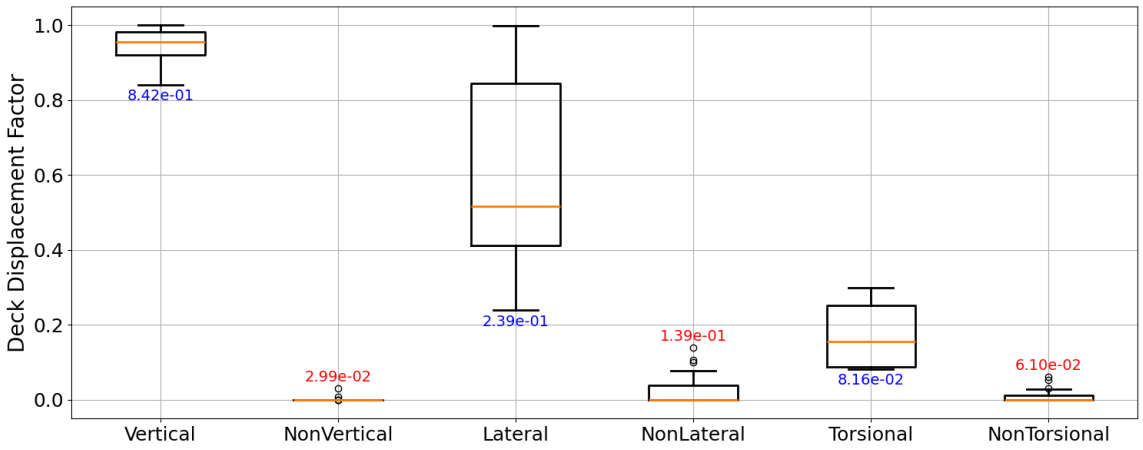


Figure 4.10: Deck Displacement Factors for Mode Type Identification

The second part of the algorithm is a similar process to the mode type identification, but instead considers the maximum absolute difference between pairs of displacements on either side of the deck midspan. The pairs are formed of opposite locations about the midspan, using the midspan as the axis of symmetry. The maximum difference within a pair for either the vertical displacement, the lateral displacement or the rotation about the X axis is then divided by either the maximum vertical displacement in the deck, the lateral displacement in the deck, or the maximum rotation about the X axis in the deck. This forms a symmetry factor used to differentiate between symmetric and asymmetric modes, given that a vibration mode has been identified in the deck. These factors are presented in Figure 4.11, and a ratio of 1.0 is selected as the threshold for all mode types. This means that the mode is asymmetric if a mode type is identified and the symmetry factor is greater than 1.0. Otherwise, the mode is determined as symmetric. The logic behind this is that an asymmetric mode is expected to have opposite displacements on either side of the midspan, meaning that the maximum difference in pairs should be approximately double the maximum displacement in the deck. This explains the consistent values of around 2.0 on the Figure 4.11. On the other hand, the symmetric modes should be identical on either side of the midspan, which would theoretically result in a symmetry factor of 0.0. This is observed in the figure for symmetric results, with values relatively close to 0.0 throughout most modes. An alternative using the concept of effective mass was also considered for this analysis, but issues were raised regarding the identification of lateral modes, which displays overlapping values between the symmetric and asymmetric modes.

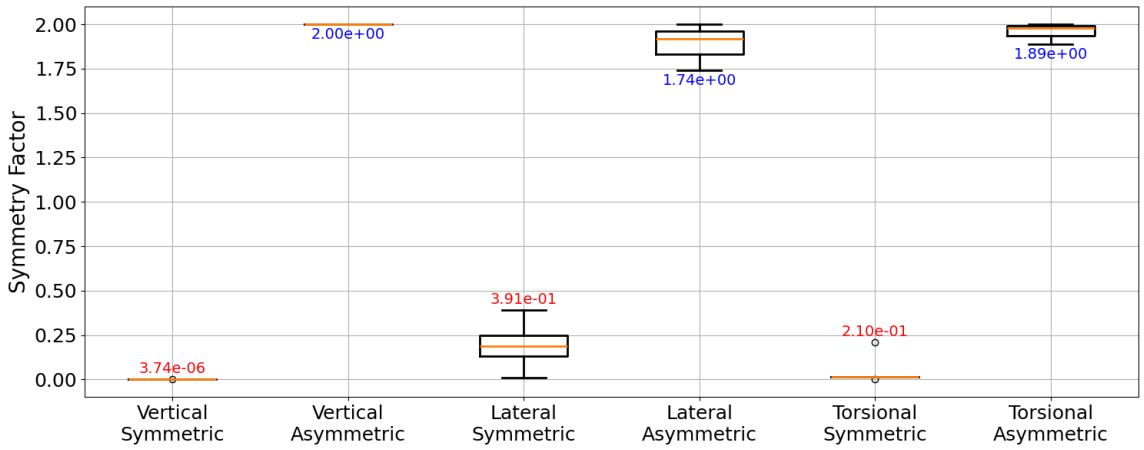


Figure 4.11: Deck Symmetry Factors for Mode Symmetry Identification

Overall, the resulting algorithm logic is available in Algorithm 4.14, which uses the previously obtained factors. This algorithm populates the relevant fields in the output file presented in the previous section.

Algorithm 4.14 Mode Identification Algorithm

```
if max_deck_rx / r_max > 0.07:
  mode_type = TORSIONAL_MODE
  if max_sym_diff_rx / max_deck_rx < 1.0:
    symmetry = SYMMETRIC_MODE
  else:
    symmetry = ASYMMETRIC_MODE
elif max_deck_uy / u_max > 0.40:
  mode_type = VERTICAL_MODE
  if max_sym_diff_uy / max_deck_uy < 1.0:
    symmetry = SYMMETRIC_MODE
  else:
    symmetry = ASYMMETRIC_MODE
elif max_deck_uz / u_max > 0.19:
  mode_type = LATERAL_MODE
  if max_sym_diff_uz / max_deck_uz < 1.0:
    symmetry = SYMMETRIC_MODE
  else:
    symmetry = ASYMMETRIC_MODE
else:
  mode_type = None
  symmetry = None
```

For the remainder of the analysis, it is assumed that the vibration modes identified by this algorithm correspond to those of the deck, even if significant effects in the cables are present. As presented in Figures 5.1 and 5.2, the first torsional modes are examples of this phenomenon, in which the cables are a considerable part of the vibration mode. However, significant torsion is still identified in the deck for these modes, justifying their classification in this study.

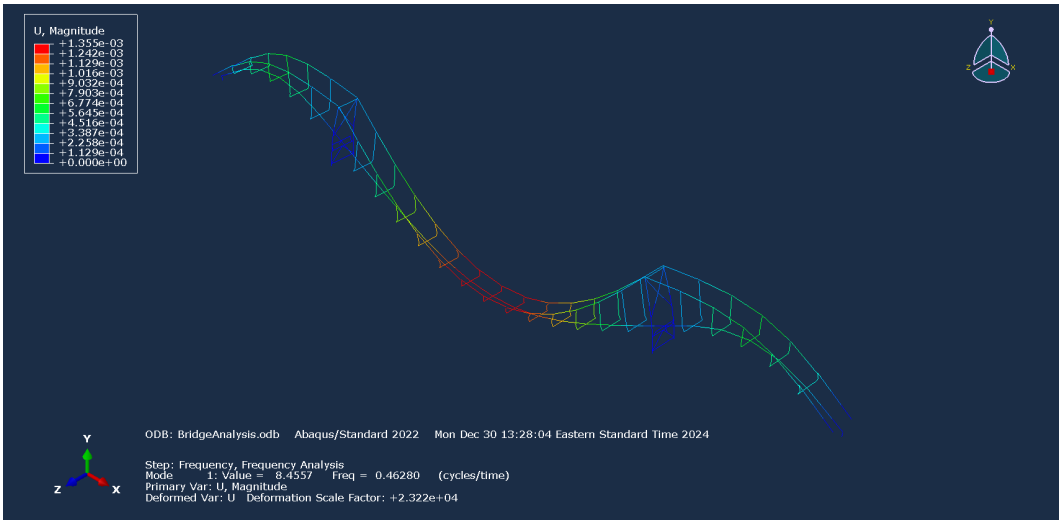
Chapter 5

Results and Discussion

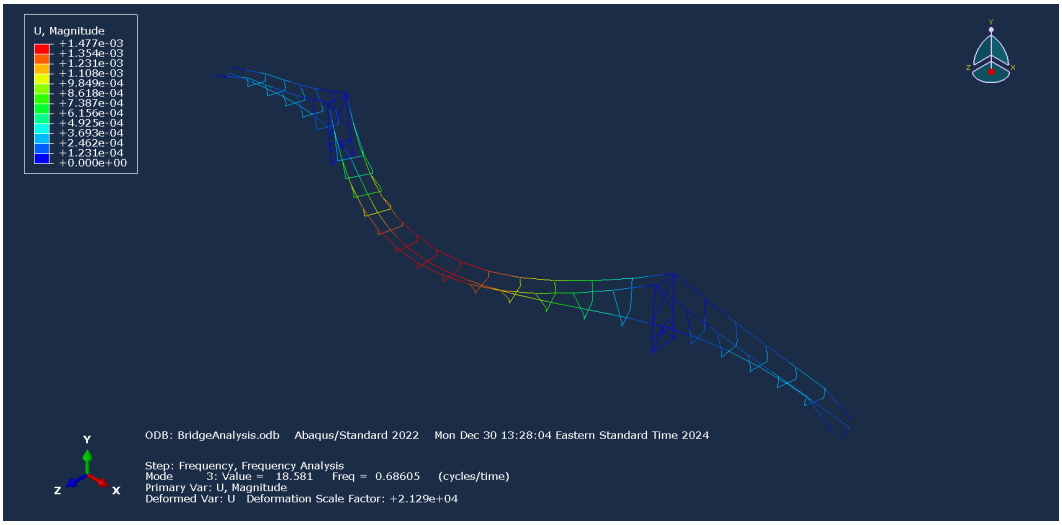
This chapter covers the results obtained from the cases defined at the end of Chapter 4. The results are organized into three distinct sections, each focusing on a specific aspect of the analysis. The first section covers the main topic of the research: the changes in natural frequencies with varying damage in the main cables. This section includes results from the FREQ11 and the FREQ34 series to establish a comparison. The second section is a complementary analysis of the FREQ11 and FREQ34 series that compares the deck displacement with the same damage scenarios. The third section is the standalone analysis of the WIND11 cases, presenting the deck displacement with time. For each section, the results obtained are presented and explained, followed by a discussion by section of the implications of these results.

5.1 Natural Frequencies Variations Analysis

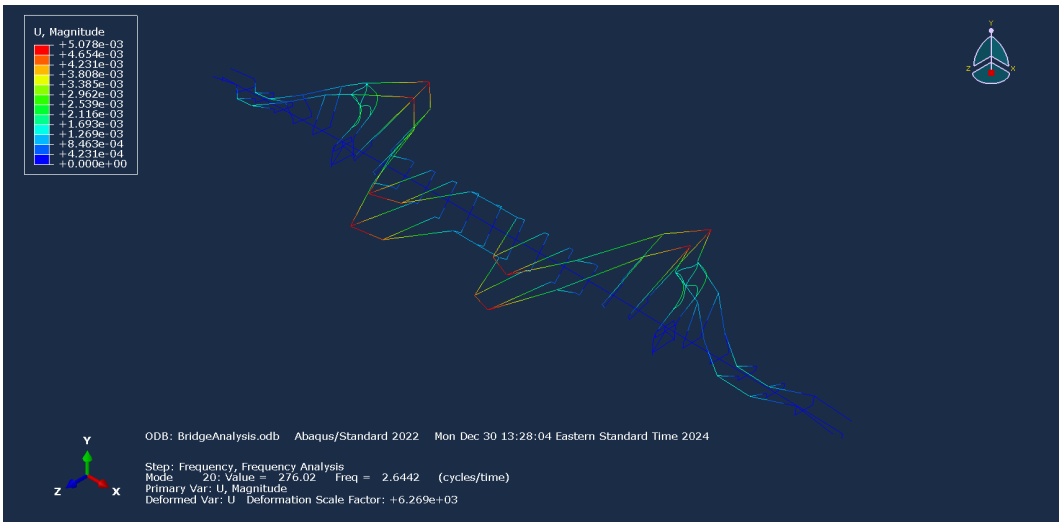
Prior to presenting the values for the natural frequencies, different modes of vibration are presented in Figures 5.1 and 5.2 for the FREQ11 baseline model. These figures illustrate the three main deck mode shapes: vertical, lateral, and torsional. The theory behind these shapes is explained more thoroughly in Section 2.1.3. Each of these modes can be categorized according to the symmetry of the vibration mode shape about the YZ plane going through the midspan of the deck, as either symmetric (Figure 5.1) or asymmetric modes (Figure 5.2).



(a) 1st Vertical Symmetric Mode (Mode 1 - 0.46280 Hz)

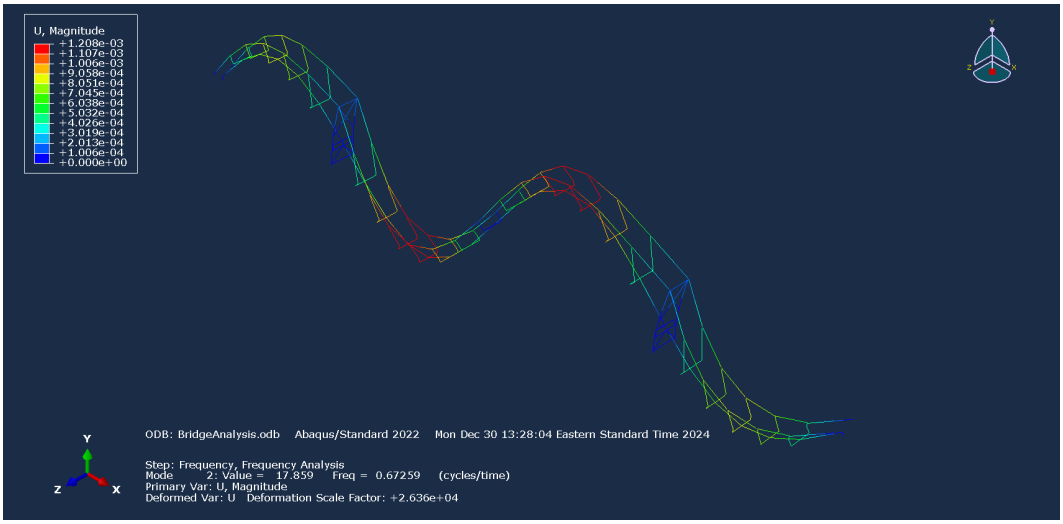


(b) 1st Lateral Symmetric Mode (Mode 3 - 0.68605 Hz)

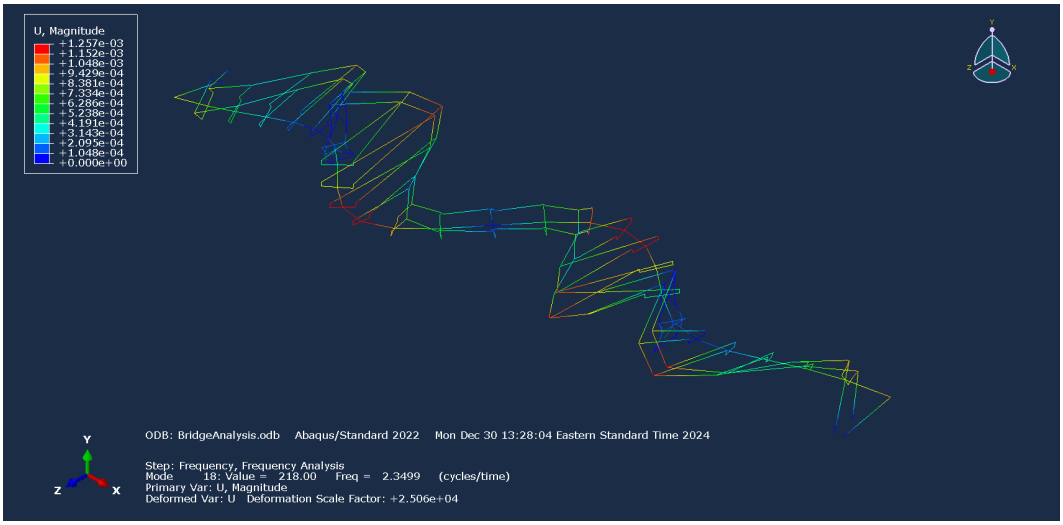


(c) 1st Torsional Symmetric Mode (Mode 20 - 2.6442 Hz)

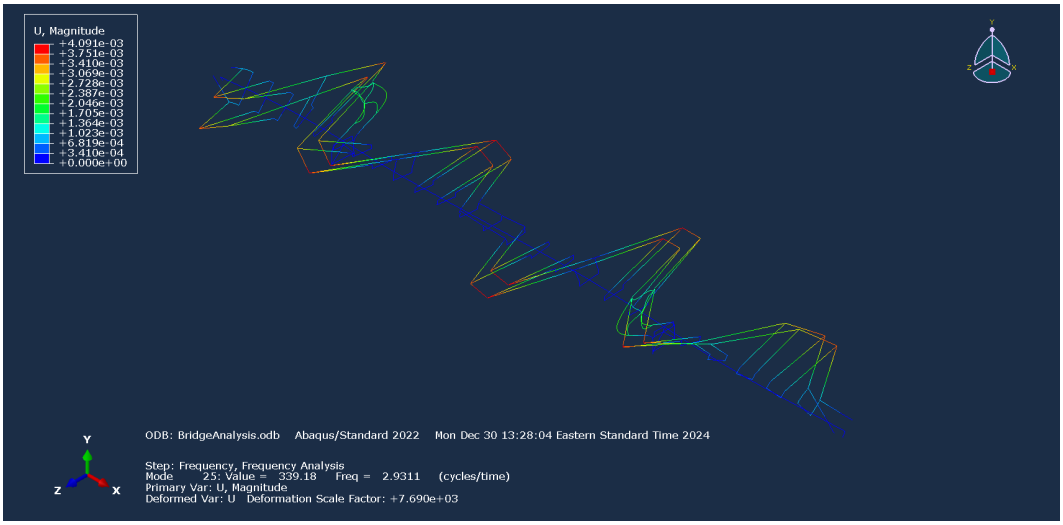
Figure 5.1: FREQ11 Baseline Model First Symmetrical Modes of Vibration



(a) 1st Vertical Asymmetric Mode (Mode 2 - 0.67259 Hz)



(b) 1st Lateral Asymmetric Mode (Mode 18 - 2.3499 Hz)



(c) 1st Torsional Asymmetric Mode (Mode 25 - 2.9311 Hz)

Figure 5.2: FREQ11 Baseline Model First Asymmetrical Modes of Vibration

The deck mode shapes presented are the first of each type for both symmetrical and asymmetrical modes. The other mode shapes that characterize the bridge are presented in Figure 5.3 for the FREQ11 baseline model, and Figure 5.4 for the FREQ34 baseline model. In these figures, the first 50 modes are presented, with their associated frequency and their mode name, if a vibration mode has been identified.

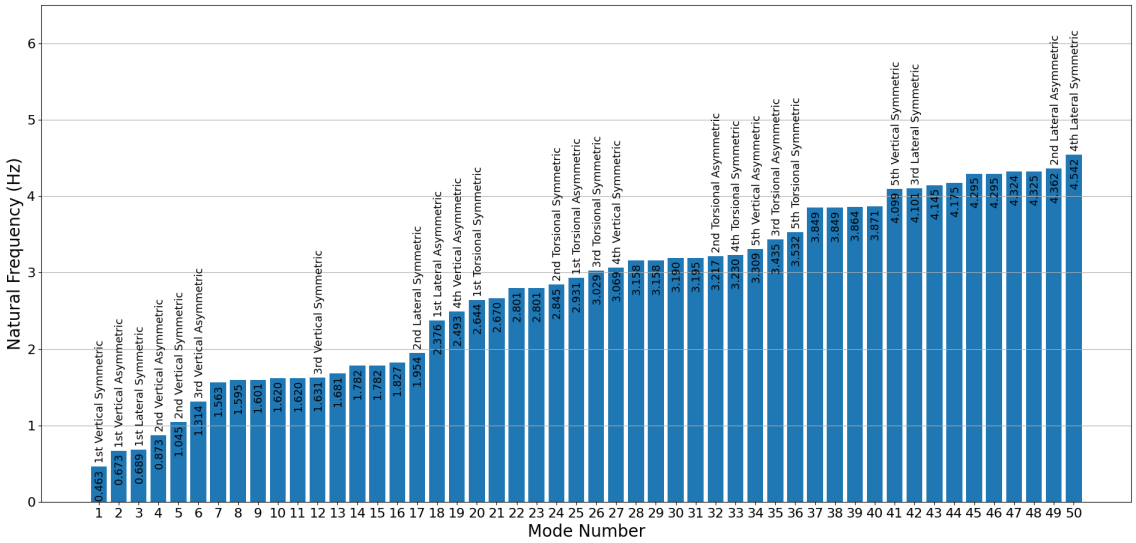


Figure 5.3: FREQ11 Baseline First 50 Natural Frequencies with Identified Modes

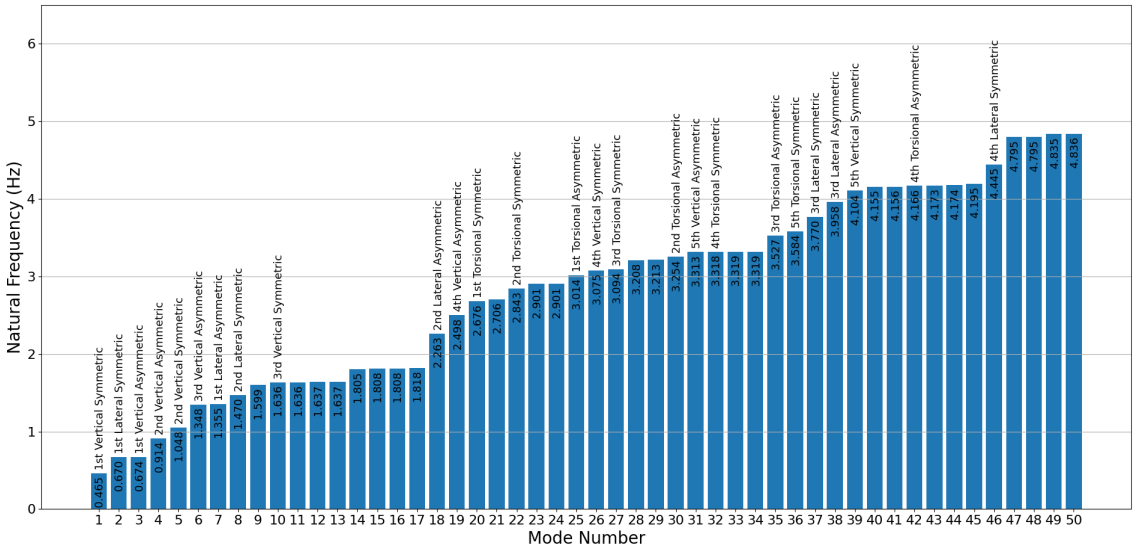


Figure 5.4: FREQ34 Baseline First 50 Natural Frequencies with Identified Modes

These modes were first identified manually and then through the created mode identification algorithm described in Section 4.5.2. While the algorithm agrees with a manual check made for

the baseline model, some damage scenarios with specific vibration modes are misidentified for the mode type (vertical, lateral, and torsional). This is a more common occurrence for the lateral and the torsional cases, which are more easily mistaken for modes dominated by cable vibrations due to some cable coupling at these frequencies in the model. As such, a preliminary filtering is done over the natural frequencies to remove the identified issues. For instance, most output for damage levels of 0.90 are removed from the results since the structure experiences high discontinuities at this level of damage, and the natural frequency changes are inconsistent with the other damage levels. To maintain conciseness, only the first three modes of each combination of mode type and symmetry are presented. Therefore, the results presented in this section are filtered based on this criterion, but some results still have unexpected peaks and discontinuities. These are attributed to step stabilization parameter changes in the model, which were adjusted to better fit some test cases, as discussed in further detail later in this chapter.

It is noticed that the values obtained for the first vibration modes for each type are considerably higher than those obtained in the different studies presented in Figure 2.4, which is expected due to the shorter length of the main span of the LLRB. However, the first mode obtained is a vertical mode instead of a lateral mode as per longer structures. This change in behaviour is attributed to the considerable difference in the length of the main span, which is of approximately 165 m for the LLRB.

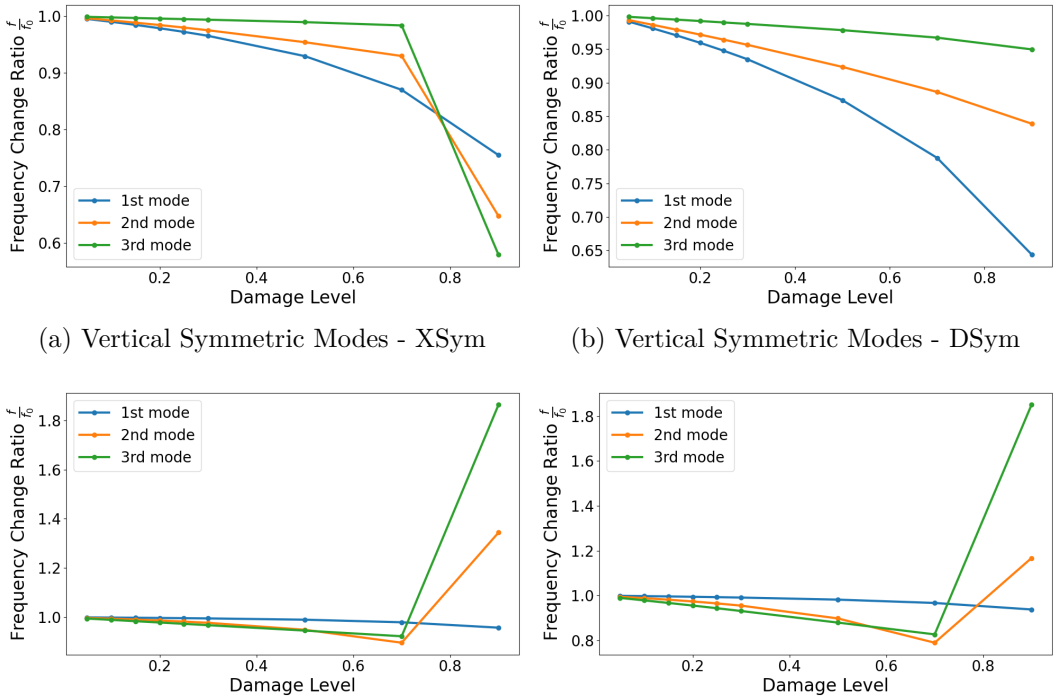
The remainder of this section is organized according to the WiresNumber type (All_, Sing, Doub, and Quad) and subdivided by test series (FREQ11 and FREQ34) and symmetry patterns (XSym and DSym) for comparison. It is important to note that the created mode identification algorithm performs considerably worse if there is any asymmetry along the X axis. Given the large number of cases, the XSym and DSym cases are preferred for the natural frequency analysis to ensure consistent results. For each type, the four combinations of series and patterns are compared, followed by a comparison of the Sing and Doub cases.

The analysis considers only the deck modes identified in the baseline model, resulting in 24 modes for FREQ11 and 26 for FREQ34. However, to streamline the presentation of the results, only the first three modes of each mode type and symmetry combination are displayed, for a maximum of 18 modes per model.

5.1.1 All_ WiresNumber

As described in Section 4.4, the All_ WiresNumber corresponds to the case where all the main cable wires in a quadrant are damaged. Then, this quadrant can be mirrored to other quadrants

based on the six available symmetry patterns. As mentioned, this analysis covers the XSym and DSym patterns, which are the only two symmetrical about the XY plane. The first mode type presented is the vertical mode of vibration in Figure 5.5, which illustrates the ratio of the damaged model to the baseline model for the natural frequencies based on the applied level of damage. Sub-figures are available for symmetrical and asymmetrical cases, as well as XSym and DSym cases.



(a) Vertical Symmetric Modes - XSym

(b) Vertical Symmetric Modes - DSym

(c) Vertical Asymmetric Modes - XSym

(d) Vertical Asymmetric Modes - DSym

Figure 5.5: FREQ11 All_ Vertical Modes of Vibration

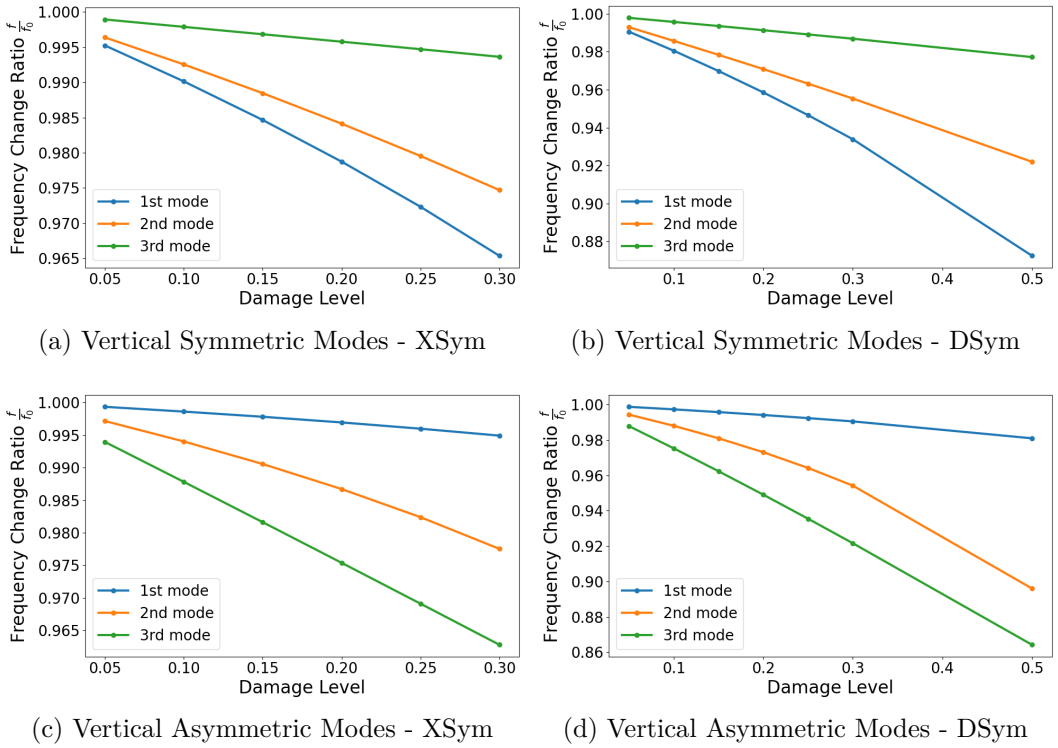
At the outset, the vertical vibration modes demonstrate the strongest relationship between damage and frequency change ratio. The general trend is a decrease in the natural frequencies as cable damage increases. However, a discrepancy is noticed in the 0.90 damage level for the 2nd and the 3rd asymmetric modes. Since the same pattern is noticed in both the XSym and the DSym symmetry patterns, it is unlikely due to step stabilization changes. Upon closer examination of the output file of these specific cases, it is determined that the discontinuity is due to a misidentification by the mode identification algorithm, which is a common occurrence for high levels of damage. Since the algorithm identifies modes based on the maximum displacement in the deck (UY, UZ, or RX) relative to the overall maximum translation or rotation within the model, significant changes in the model's characteristics can result in shifts that exceed the calibration range of the baseline model. The symmetry of the mode is determined based on the maximum absolute difference

in deck displacements between both sides, which is a reliable method within reasonable damage scenarios but may fail in extreme cases. This is the principal reason for omitting the 0.90 damage level in most of the analyses presented in this chapter.

Otherwise, the other damage levels exhibit a relatively smooth relationship for frequency changes. The trend is linear in lower damage levels but then takes an exponential form at damages levels of 0.30 and above. The first mode is the most affected for the symmetric modes in terms of the change in natural frequencies, followed by the second and then the third mode. In contrast, the opposite trend is observed for the asymmetric modes. Note that the ranges of the frequency change ratio are similar for the symmetric and asymmetric modes.

Finally, apart from the discontinuities at 0.90 damage, the XSym and the DSym curves are very similar, with only a slight change in the amplitude of the frequency shifts.

These results may be compared to the FREQ34 test series for equivalent scenarios, as presented in Figure 5.6. However, due to the instabilities experienced with the FREQ34 model, most of the higher levels of damage for the All_ WiresNumber are missing.



(a) Vertical Symmetric Modes - XSym

(b) Vertical Symmetric Modes - DSym

(c) Vertical Asymmetric Modes - XSym

(d) Vertical Asymmetric Modes - DSym

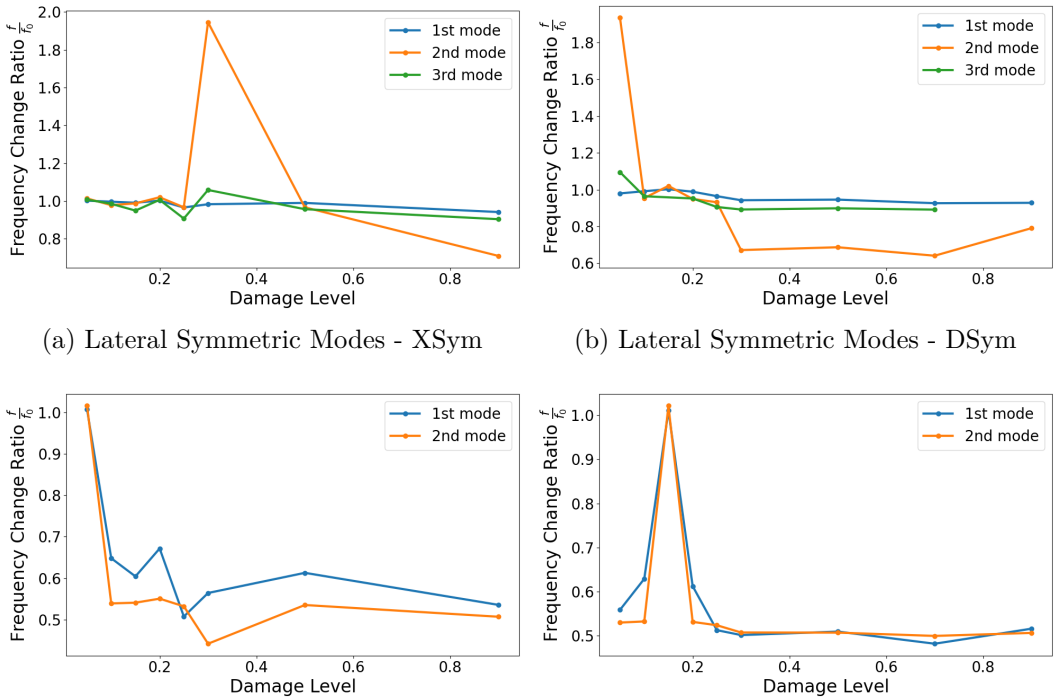
Figure 5.6: FREQ34 All_ Vertical Modes of Vibration

For background on the analysis of the FREQ11 against the FREQ34 test series, the model using the 3x4 rectangular pattern for the main cable wires is more complex to stabilize and run. This implies that the model is much more susceptible to abort while solving a case, and it takes, on

average, five times longer to run a single case. Therefore, if the results of the FREQ11 series align with those of the FREQ34 series, it would be justified to use the 1x1 pattern simplification for future analyses of this type. On the other hand, if the results are considerably different, additional considerations should be made for alternative wire patterns.

The results from Figures 5.5 and 5.6 indicate that the FREQ34 model behaviour is almost identical to the FREQ11 model in the lower damage range. This validates the equivalency of the FREQ11 model to the FREQ34 model based on the uniform damage application method.

Next, the FREQ11 lateral vibration mode results are presented in Figure 5.7 below.



(a) Lateral Symmetric Modes - XSym

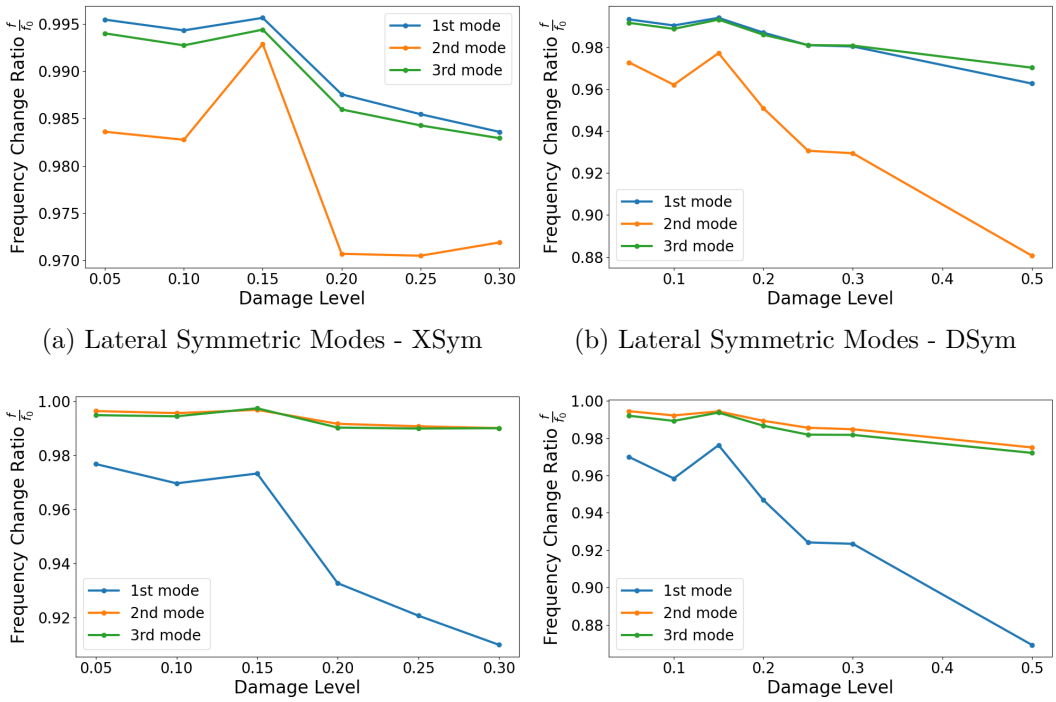
(b) Lateral Symmetric Modes - DSym

(c) Lateral Asymmetric Modes - XSym

(d) Lateral Asymmetric Modes - DSym

Figure 5.7: FREQ11 All_ Lateral Modes of Vibration

It is determined that the results are far less promising than for the vertical modes for establishing a strong correlation. First, the lateral asymmetric DSym figure demonstrates an apparent issue with the mode identification algorithm since even low damage levels display a 0.5-0.6 frequency change ratio; ratios closer to unity for lower damages would be expected. A peak closer to 1.0 is also noticed at the 0.15 damage level, which would confirm this theory. While this explains the abnormal results found in the asymmetrical modes, the symmetrical modes still exhibit inconsistent patterns regardless of the identification algorithm. No consistent patterns were established within these modes. However, the FREQ34 series presents considerably different results, as demonstrated in Figure 5.8.



(a) Lateral Symmetric Modes - XSym

(b) Lateral Symmetric Modes - DSym

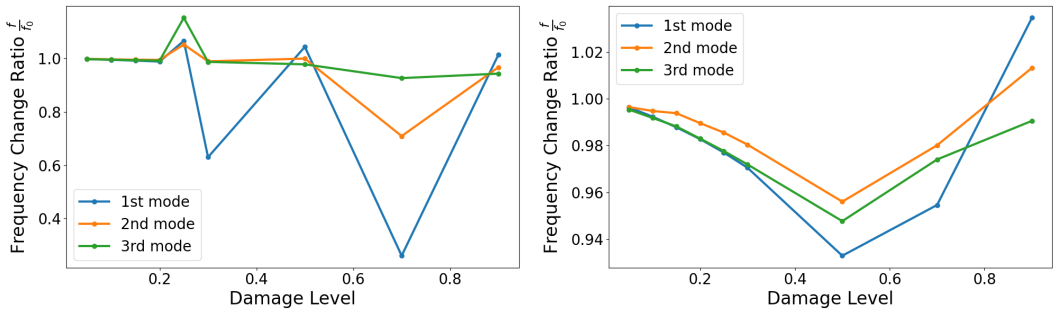
(c) Lateral Asymmetric Modes - XSym

(d) Lateral Asymmetric Modes - DSym

Figure 5.8: FREQ34 All_ Lateral Modes of Vibration

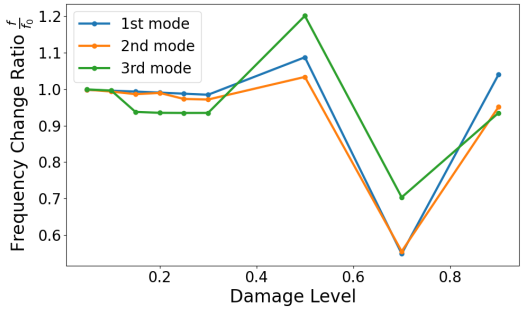
These results differ considerably from the 1x1 wire pattern, exhibiting more consistency throughout the available damage levels. However, it is essential to note that these results only cover up to 0.30 or 0.50 damage levels due to the missing test cases. The general tendency in these results is also a decrease in the natural frequencies as the damage increases in the cables. A key difference between the symmetric and the asymmetric modes is observed in the DSym results, in which the second symmetric mode behaves precisely as per the first asymmetric mode. Regardless, these results do not suggest that a strong relationship exists between the change in damage and the natural frequencies of the lateral vibration modes for the All_ cases. Moreover, the consistent peak at the 0.15 damage level complicates establishing a clear relationship, as certain frequency change ratio could suggest damage on either side of the 0.15 damage level. Overall, it is notable that the FREQ34 results exhibit greater stability in the results.

Next, the FREQ11 results for the torsional modes are presented in Figure 5.9.

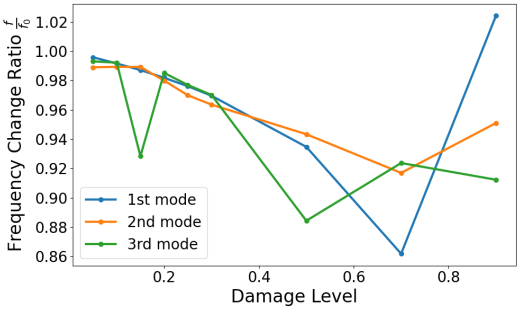


(a) Torsional Symmetric Modes - XSym

(b) Torsional Symmetric Modes - DSym



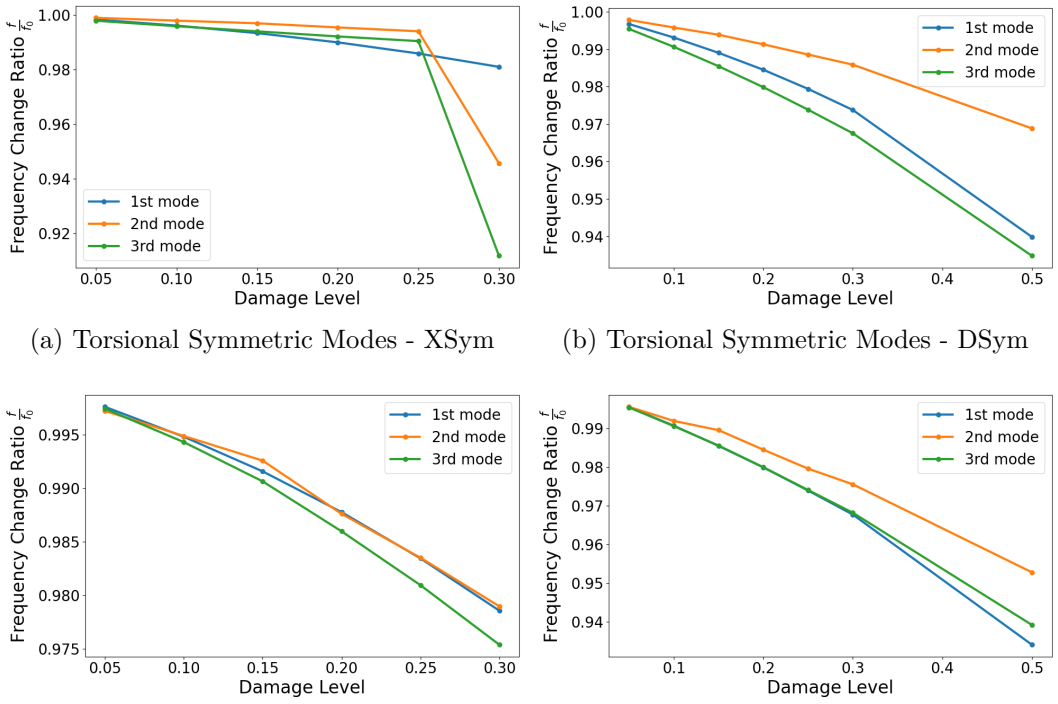
(c) Torsional Asymmetric Modes - XSym



(d) Torsional Asymmetric Modes - DSym

Figure 5.9: FREQ11 All_ Torsional Modes of Vibration

For most of these figures, the patterns are inconsistent, except at lower damage levels for the symmetric DSym case. While some peaks can be attributed to issues with the mode identification algorithm, the overall conclusion is that the results fluctuate too much to establish a clear correlation. For the symmetric DSym case, curves for damage levels of 0.50 and lower are smoother, but this changes for the 0.70 and the 0.90. If this is due to a misidentification from the algorithm, as observed in other previous cases for 0.90, then this mode could be promising but otherwise not valuable for a machine learning model analysis. Regarding the FREQ34 cases, the improvement in the stability of the results is even more significant than for the lateral modes, which can be observed in Figure 5.10.



(a) Torsional Symmetric Modes - XSym

(b) Torsional Symmetric Modes - DSym

(c) Torsional Asymmetric Modes - XSym

(d) Torsional Asymmetric Modes - DSym

Figure 5.10: FREQ34 All_ Torsional Modes of Vibration

The results, available only for damage levels up to 0.30 or 0.50, resemble those of the vertical modes. The change in natural frequencies at lower damage levels is almost linear, with the rate of change increasing slightly as 0.50 is approached. Interestingly, the behaviour observed for the 0.30 damage for the symmetrical mode with the XSym case is very similar to the pattern from the vertical symmetric mode from the FREQ11 series for the XSym. Otherwise, the results for these scenarios are consistent for low damage levels.

In general, the results from these cases demonstrate a correlation between the damage level applied on main cable wires and the changes in the natural frequencies of deck vibration modes. Some stronger relationships are available for the vertical modes for both symmetrical and asymmetrical modes. However, the difference between XSym and DSym patterns is only reflected in the magnitude of the frequency change ratio. This implies that it is impossible to differentiate between the results for an XSym case with medium damage and a DSym case with low damage when using only the vertical modes for the All_ scenario. On the other hand, the natural frequencies of the lateral modes are not good indicators of damage in the main cables. Finally, it is unclear at this stage if the frequencies of the torsional modes are adequate metrics of the damage since their instabilities in the FREQ11 series made them irrelevant. In contrast, the FREQ34 results demonstrated more promise for the development of a machine learning model.

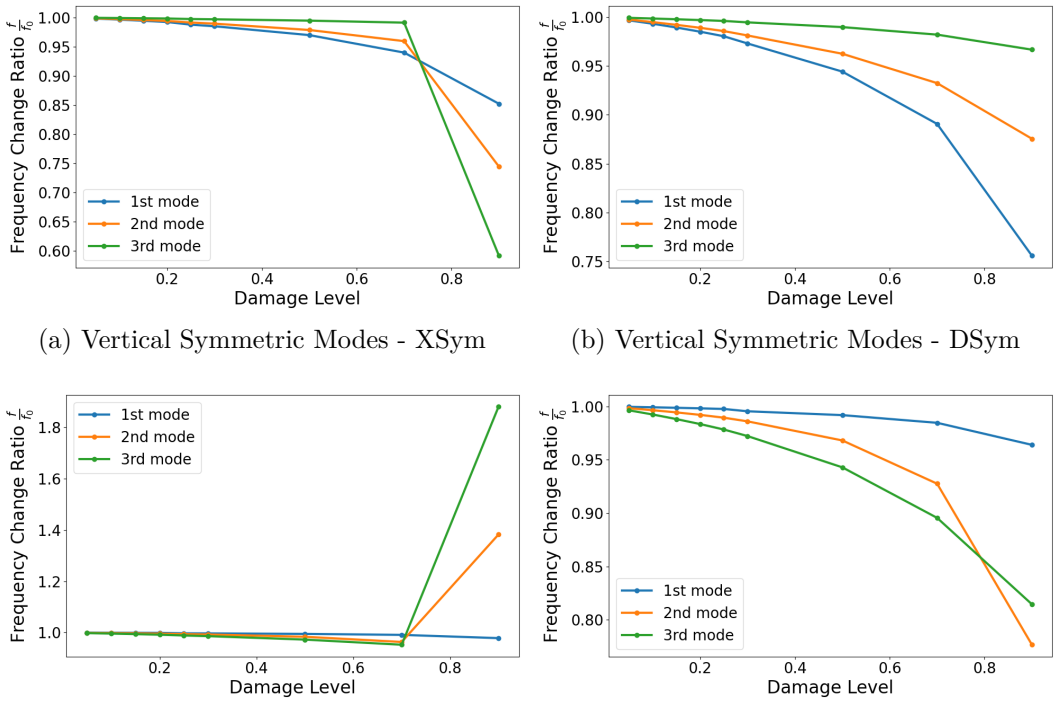
Naturally, the All_ scenario is not a pattern expected in a real suspension bridge, and the direct results from this analysis do not contribute considerably to this study by themselves. The value of these cases lies in the fact that key behaviours within these results establish a baseline for other cases, leading to preliminary conclusions that may be applicable to other cases:

- The shifts of natural frequencies of the vertical vibration modes present the most apparent relationships among the deck modes to the applied damage in the cables.
- The lateral vibration modes of the deck do not change consistently under the applied damage.
- For uniform damage application, the FREQ34 cases offer more consistency in the available results than the FREQ11 counterpart.
- The FREQ11 model is more stable than the FREQ34 model, especially at higher damage levels.
- XSym and DSym are differentiated mainly by the magnitude of the change in natural frequencies.

5.1.2 Quad WiresNumber

The analysis of the Quad cases is very similar to the All_ cases, with only a centralized critical set of wires being damaged. From a preliminary assessment of the results, it is determined that, as per the All_ analysis, the lateral modes offer minimal correlation between the natural frequencies of the deck and applied damage. The same holds true for the FREQ11 torsional modes, where the results are inconsistent. Due to the irrelevancy of these results, they are presented only in Appendix I in Figure I.1, Figure I.2, and Figure I.3 for the FREQ11 lateral, FREQ34 lateral, and FREQ11 torsional, respectively.

The remaining cases are presented in this section, starting with the vertical modes for the FREQ11 series in Figure 5.11.



(a) Vertical Symmetric Modes - XSym

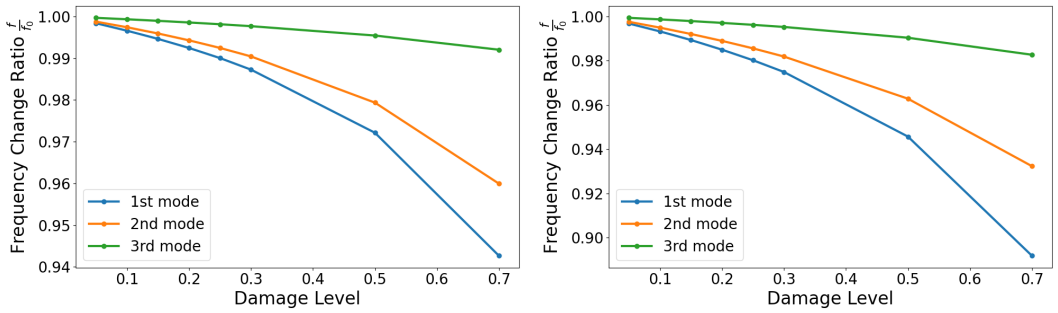
(b) Vertical Symmetric Modes - DSym

(c) Vertical Asymmetric Modes - XSym

(d) Vertical Asymmetric Modes - DSym

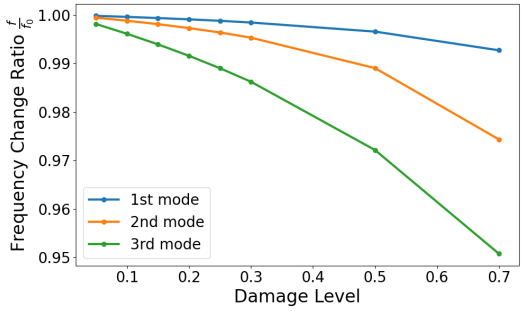
Figure 5.11: FREQ11 Quad Vertical Modes of Vibration

It is worth noting that these results closely resemble their All_ equivalent counterparts. With the exception of some inconsistencies at the 0.90 damage level, the observed curves are largely identical, with changes primarly in the magnitude of the change ratio. This equivalency is also evident in the FREQ34 cases, presented in Figure 5.12. The FREQ34 cases exhibit similar magnitude changes to the FREQ11 cases when compared to the All_ cases.

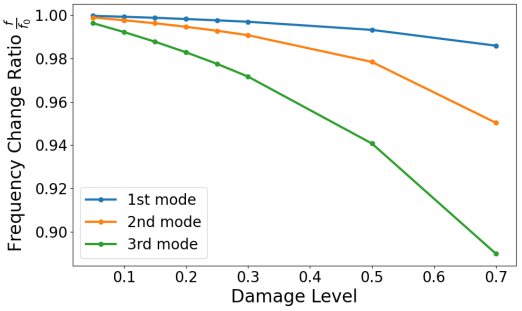


(a) Vertical Symmetric Modes - XSym

(b) Vertical Symmetric Modes - DSym



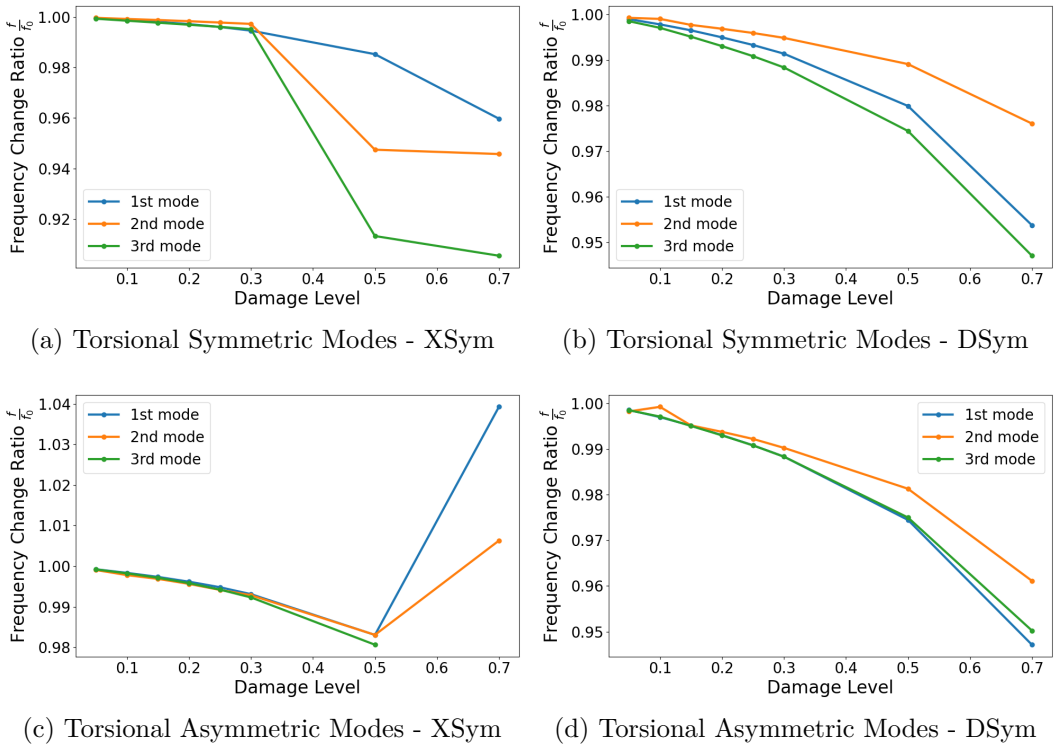
(c) Vertical Asymmetric Modes - XSym



(d) Vertical Asymmetric Modes - DSym

Figure 5.12: FREQ34 Quad Vertical Modes of Vibration

The last set of results for the Quad cases are presented in Figure 5.13, which presents the results for the FREQ34 torsional modes.



(a) Torsional Symmetric Modes - XSym

(b) Torsional Symmetric Modes - DSym

(c) Torsional Asymmetric Modes - XSym

(d) Torsional Asymmetric Modes - DSym

Figure 5.13: FREQ34 Quad Torsional Modes of Vibration

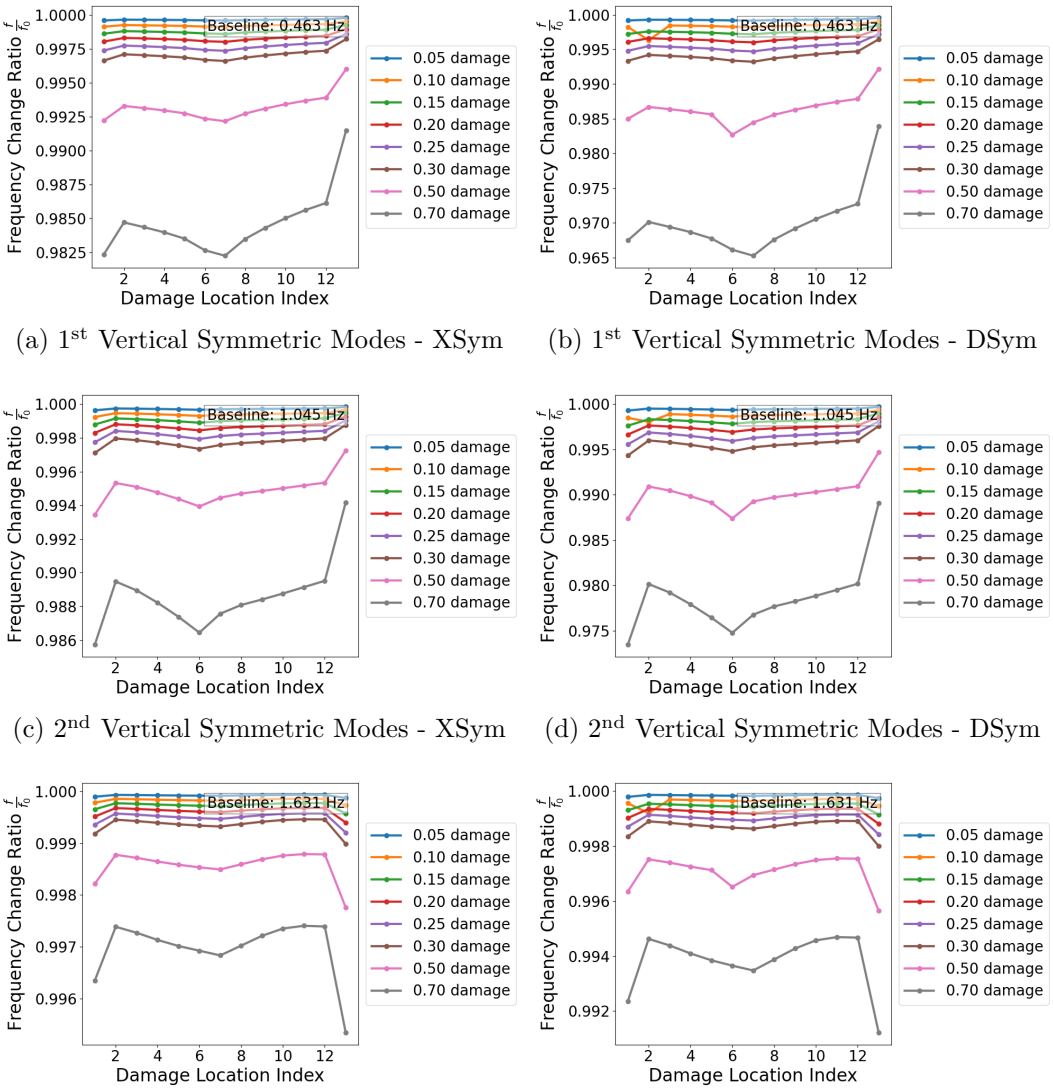
While there are more distinctions in the figures when comparing them to the All_ cases, the same general patterns are also observed for the torsional modes. Overall, all the correlations established for the All_ cases apply to the Quad cases. The only notable difference is a change in the magnitude of the frequency change ratio, with All_ cases being subjected to more changes. For most cases, the Quad cases produce a change in the frequency ratio of about 50 to 75% of the one observed in equivalent All_ cases. For example, a Quad case would typically have a 0.97-0.98 change ratio if the equivalent All_ case has a ratio of 0.96. This has considerable implications, as it provides insight on which damage contributes most to the changes in natural frequencies. From these results, it could be deduced that the cables surrounding the towers have the most effect on the natural frequencies. On the other hand, the changes in natural frequencies are not necessarily directly additive, and the Sing and Doub cases are needed to determine which alternative is causing this phenomenon.

5.1.3 Sing WiresNumber

With the cases involving damage in long continuous sections covered, the next step is to analyze those with isolated damaged cables. In this current case, Sing, the damage is applied to isolated cables. The XSym and DSym symmetry patterns indicate that only two or four main cable spans

are affected across the entire bridge. By varying the location of the applied damage through the different spans composing the main cables, the goal is to identify any relationships that could potentially reveal the damage location based on changes in frequencies. The expected critical locations are at the anchors (WireIndex of 1), around the towers (WireIndex of 6 and 7), and at the midspan of the main span of the bridge (WireIndex of 13). It should be noted that the 0.90 damage level did not provide accurate results and is therefore removed from all figures in this section. This can be justified by the fact that it is not a realistic level of damage for the cables, which would means that the behaviour of the bridge changes, making the mode identification algorithm fail.

The first set of results is for the vertical symmetric modes with the FREQ11 series, presented in Figure 5.14.



(a) 1st Vertical Symmetric Modes - XSym

(b) 1st Vertical Symmetric Modes - DSym

(c) 2nd Vertical Symmetric Modes - XSym

(d) 2nd Vertical Symmetric Modes - DSym

(e) 3rd Vertical Symmetric Modes - XSym

(f) 3rd Vertical Symmetric Modes - DSym

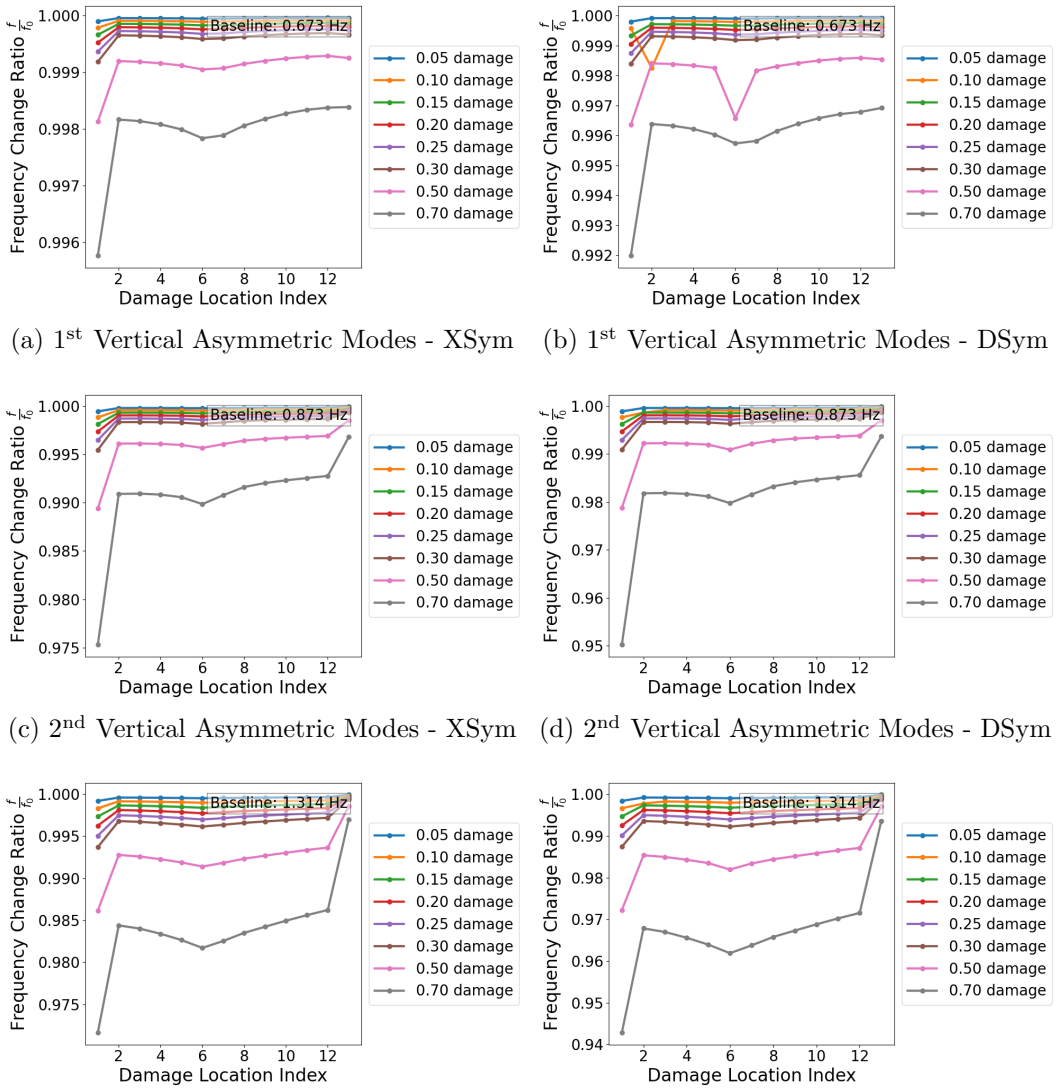
Figure 5.14: FREQ11 Sing Symmetrical Vertical Modes of Vibration

These figures reveal how the natural frequencies of the deck change with the modification of the damage location and level, with the first two modes exhibiting highly similar behaviours. First, a sharp decrease in the frequency magnitude is noticed when the cable closest to the anchor is damaged (location index of 1), which is even more accentuated in the second mode. Second, the application of damage to the cables at the midspan of the main span has a considerably lower impact on the natural frequencies than when applied to any other location. Third, except for the location index 1 (at the anchor), the most significant frequency decrease occurs at location index 6 for the first mode and at location index 7 for the second mode, with a roughly linear change in natural frequency observed for the remaining adjacent locations. This tendency suggests that, aside from the anchor, the most susceptible locations for cable damage in the vertical symmetric

mode are near the tower, either just before or just after, depending on the mode. This behaviour can be explained by the fact that this main cable section experiences the greatest tension due to its steepest angle in the XY plane. On the other hand, the sudden increase and decrease in the frequency shifts at the anchor and the midspan could be correlated to the hangers that are steel beams at those locations (the first hanger and the three central hangers at the midspan). Regarding the third mode, similar patterns are observed for damage closer to the anchor and the towers, but an inverse phenomenon is noticed at the midspan (Figures 5.14e and 5.14f). For this mode, the midspan experiences a sudden decrease in the natural frequencies, even more significant than the minima at the anchor.

Throughout these different Figures (5.14), it is interesting to notice that most damage levels have a similar patterns, with a more remarkable change in magnitudes as the damage level increases. This is consistent with the results observed in the All_ and Quad cases for the vertical symmetric modes. Furthermore, the curves are almost identical between the XSym and the DSym symmetry patterns, but the change ratio is approximately twice as important for the DSym. This ratio is considerably more significant than the one observed for the All_ and Quad cases and is consistent throughout Sing cases. Finally, it should be noted that the changes in natural frequencies are substantially less significant in the third mode, which would make them more difficult to distinguish from environmental factors in a real scenario. As discussed in Section 4.4.2, the equivalent results for the FREQ34 are available only for damage location indices 1, 6, 7, and 13. These results were primarily computed to validate the critical points, namely damage near the anchor, the tower, and the midspan. The results are presented in Figure I.4 in Appendix I, and the main takeaway is that the natural frequency shifts are mostly consistent between the FREQ11 and FREQ34 models.

The symmetrical modes can then be compared to the asymmetrical ones in in Figure 5.15.



(a) 1st Vertical Asymmetric Modes - XSym (b) 1st Vertical Asymmetric Modes - DSym

(c) 2nd Vertical Asymmetric Modes - XSym (d) 2nd Vertical Asymmetric Modes - DSym

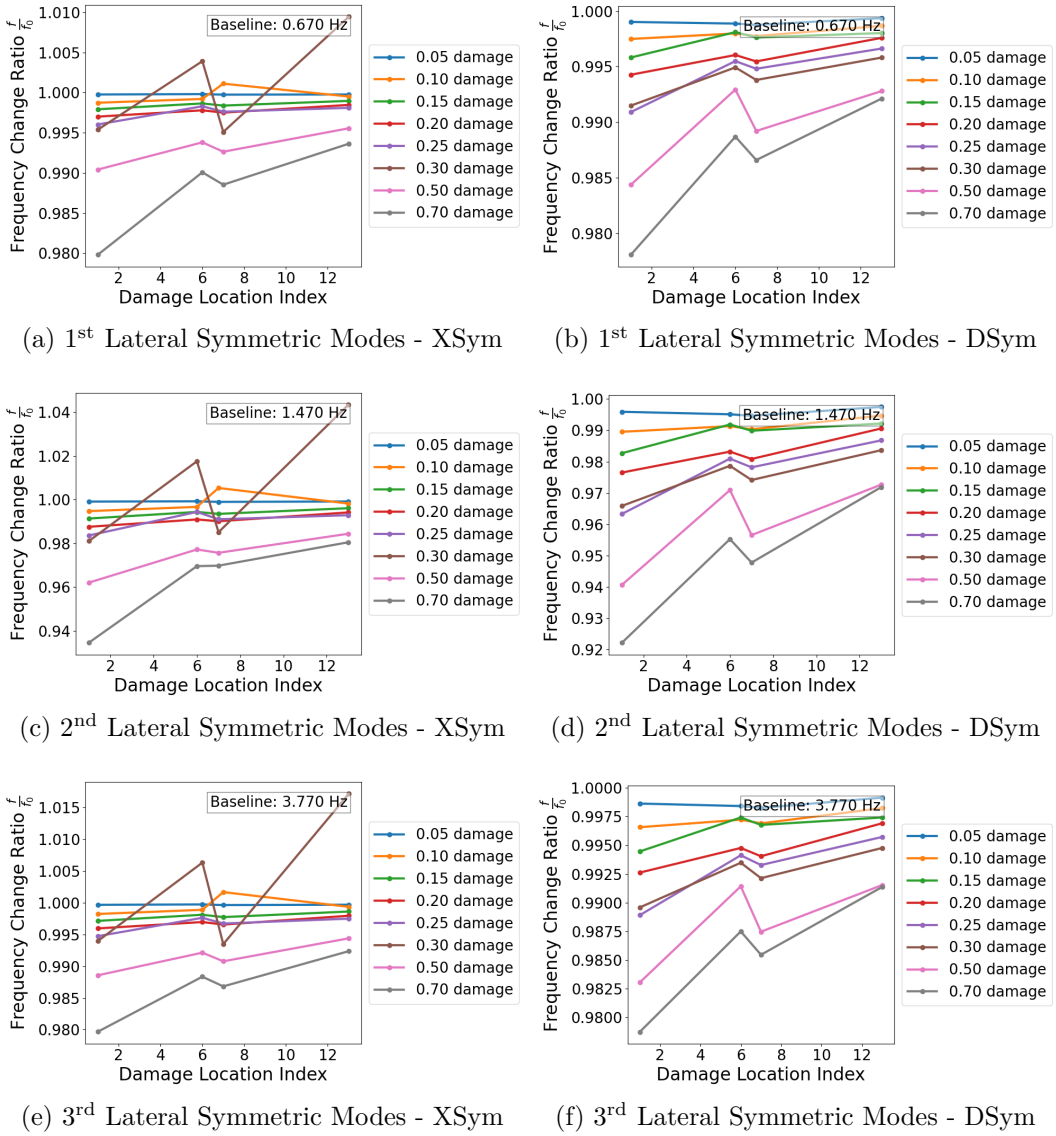
(e) 3rd Vertical Asymmetric Modes - XSym (f) 3rd Vertical Asymmetric Modes - DSym

Figure 5.15: FREQ11 Sing Asymmetrical Vertical Modes of Vibration

These results reveal repeating patterns from the symmetric vertical modes. First, all modes have a local minimum in the frequencies at the damage location index 6. Second, applying damage to the span closest to the anchor (location index 1) has a significant impact on the natural frequencies, with a more pronounced decrease than in the symmetrical modes. Third, the difference between XSym and DSym patterns is also approximately twice the shift in the natural frequencies. Fourth, except for the expected anomalies, increased damage level results in decreased natural frequencies. However, it can be determined that the first mode does not display a peak in the results for the damage at the midspan, whereas the second and third modes do. This is an essential contrast with the symmetrical modes. Another difference with the symmetrical modes is that out of the first three modes, the first one is the one in which the changes in the natural frequencies

are the least important. Finally, these results for the FREQ11 model are generally consistent with those obtained for the FREQ34 model, reinforcing the validation of using the FREQ11 model for this analysis. These findings are presented in Figure I.5 in Appendix I.

The following vibration modes would be the lateral symmetric and asymmetric vibration modes for model FREQ11, but as per Figures I.6 and I.7 in Appendix I, no correlation could be observed. These lateral vibration mode results seem to indicate that the changes in natural frequencies are independent of the changes in the damage level and location. On the other hand, the FREQ34 results are slightly more consistent in appearance, as demonstrated in Figure 5.16 for the symmetrical modes.



(a) 1st Lateral Symmetric Modes - XSym (b) 1st Lateral Symmetric Modes - DSym

(c) 2nd Lateral Symmetric Modes - XSym (d) 2nd Lateral Symmetric Modes - DSym

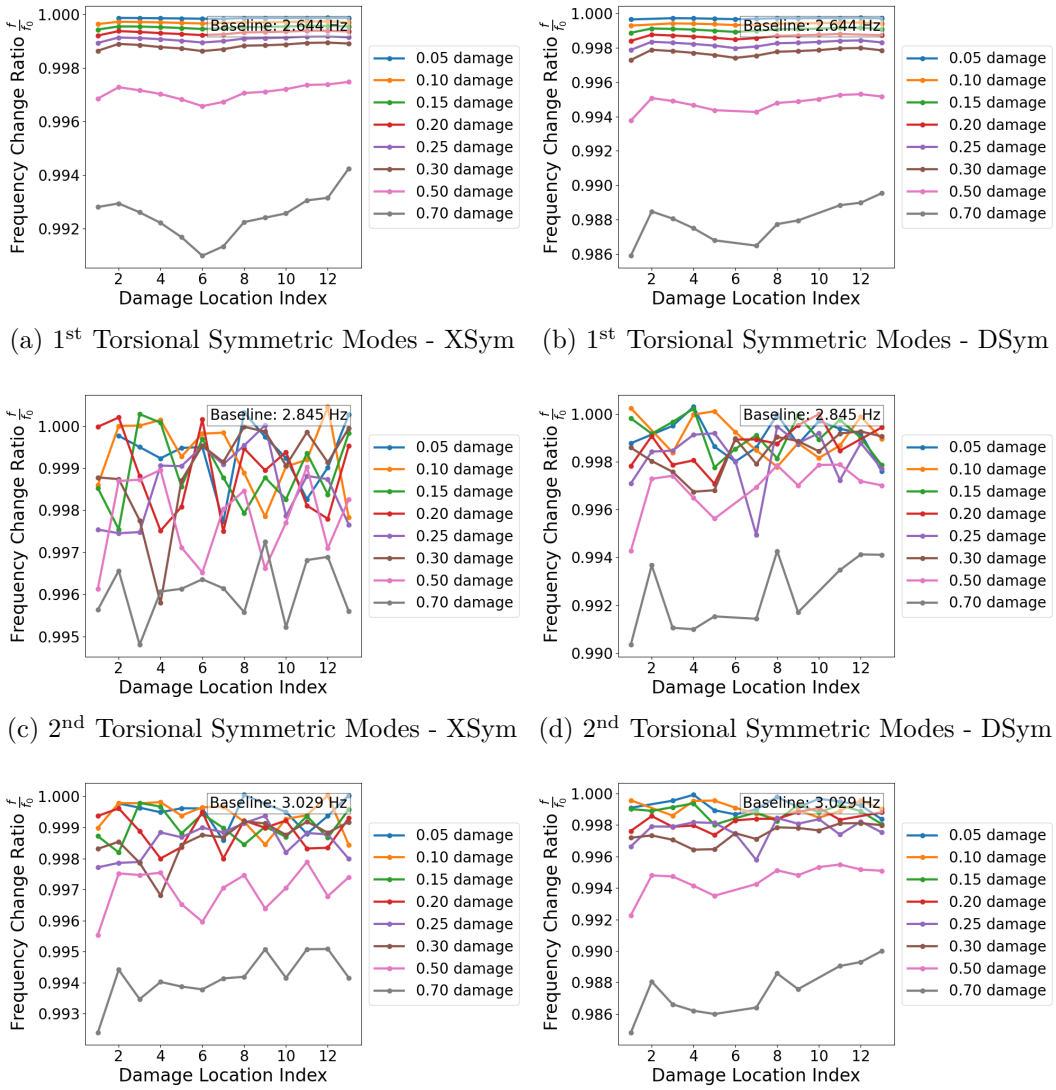
(e) 3rd Lateral Symmetric Modes - XSym (f) 3rd Lateral Symmetric Modes - DSym

Figure 5.16: FREQ34 Sing Symmetrical Lateral Modes of Vibration

These Figures (5.16) indicate that the natural frequencies decrease as the damage level in-

creases. This aligns with the FREQ34 All_ results for the lateral modes, for which the FREQ11 model also performed poorly. Otherwise, a more noticeable decrease in the natural frequencies is observed when the damage is applied to the wire closest to the anchor, consistent with the results from the vertical modes. However, significant overlapping of damage level lines can be observed in the results, making them challenging to interpret. More cases would be required to reach more conclusive correlations. A similar scenario is encountered for the lateral symmetric modes of the FREQ34 model, which are presented in Figure I.8 in Appendix I. Again, no further generalized tendencies are observed in these results, except for those aforementioned for the symmetrical modes. Due to the lack of consistency in the results, no further analysis is performed, as more cases are required to draw conclusions.

Next, the torsional modes are examined, with the symmetrical vibration modes for the FREQ11 model presented in Figure 5.17.



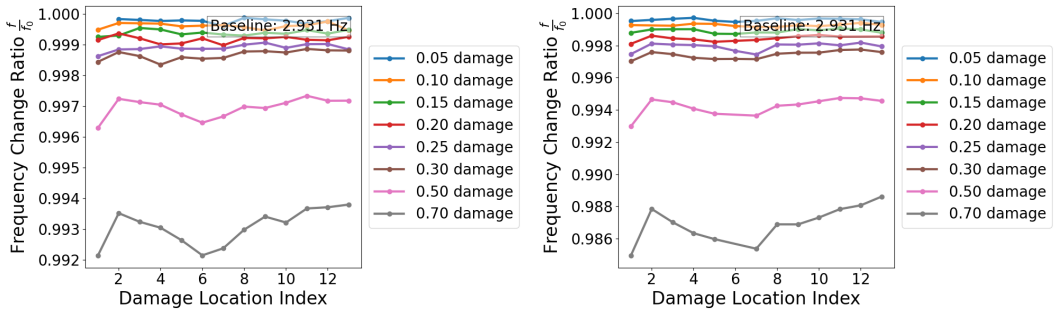
(a) 1st Torsional Symmetric Modes - XSym (b) 1st Torsional Symmetric Modes - DSym

(c) 2nd Torsional Symmetric Modes - XSym (d) 2nd Torsional Symmetric Modes - DSym

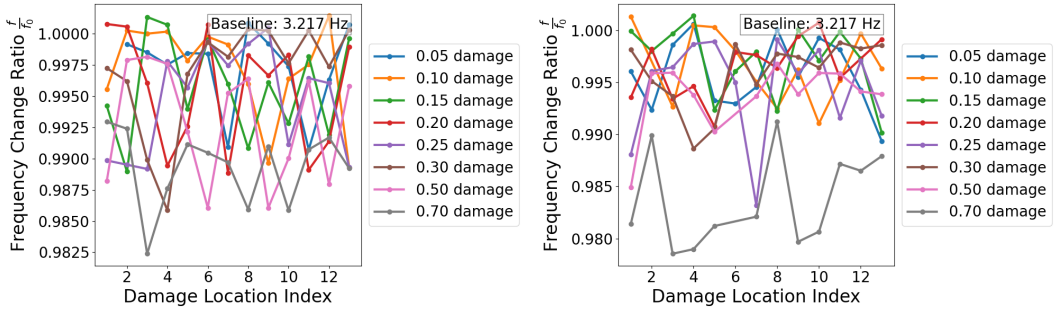
(e) 3rd Torsional Symmetric Modes - XSym (f) 3rd Torsional Symmetric Modes - DSym

Figure 5.17: FREQ11 Sing Symmetrical Torsional Modes of Vibration

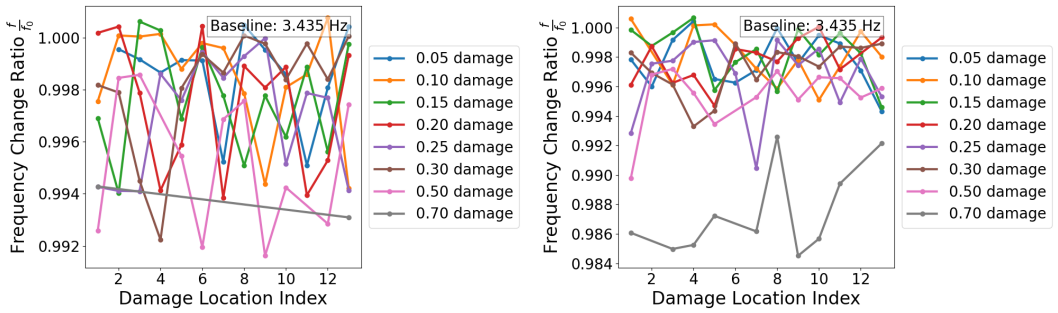
While the second and third modes are inconclusive due to their high variations, the first mode displays some previously observed patterns. For this mode, applying damage to the wire closest to the anchor (index 1) and the tower-adjacent (index 6 and 7) wires has the most significant effect on the natural frequencies. However, it is unclear if the lack of accentuated decrease at the anchor for the XSym case (Figure 5.17 is due to an inaccuracy in the model. A slight increase is noticed as damage reaches the midpsan of the deck. For damage levels of 0.50 and less, the results are far less correlated to location. These conclusions also hold for the first asymmetrical torsional mode (Figure 5.17a), for which the second and third modes seem stochastic regarding damage location.



(a) 1st Torsional Asymmetric Modes - XSym (b) 1st Torsional Asymmetric Modes - DSym



(c) 2nd Torsional Asymmetric Modes - XSym (d) 2nd Torsional Asymmetric Modes - DSym

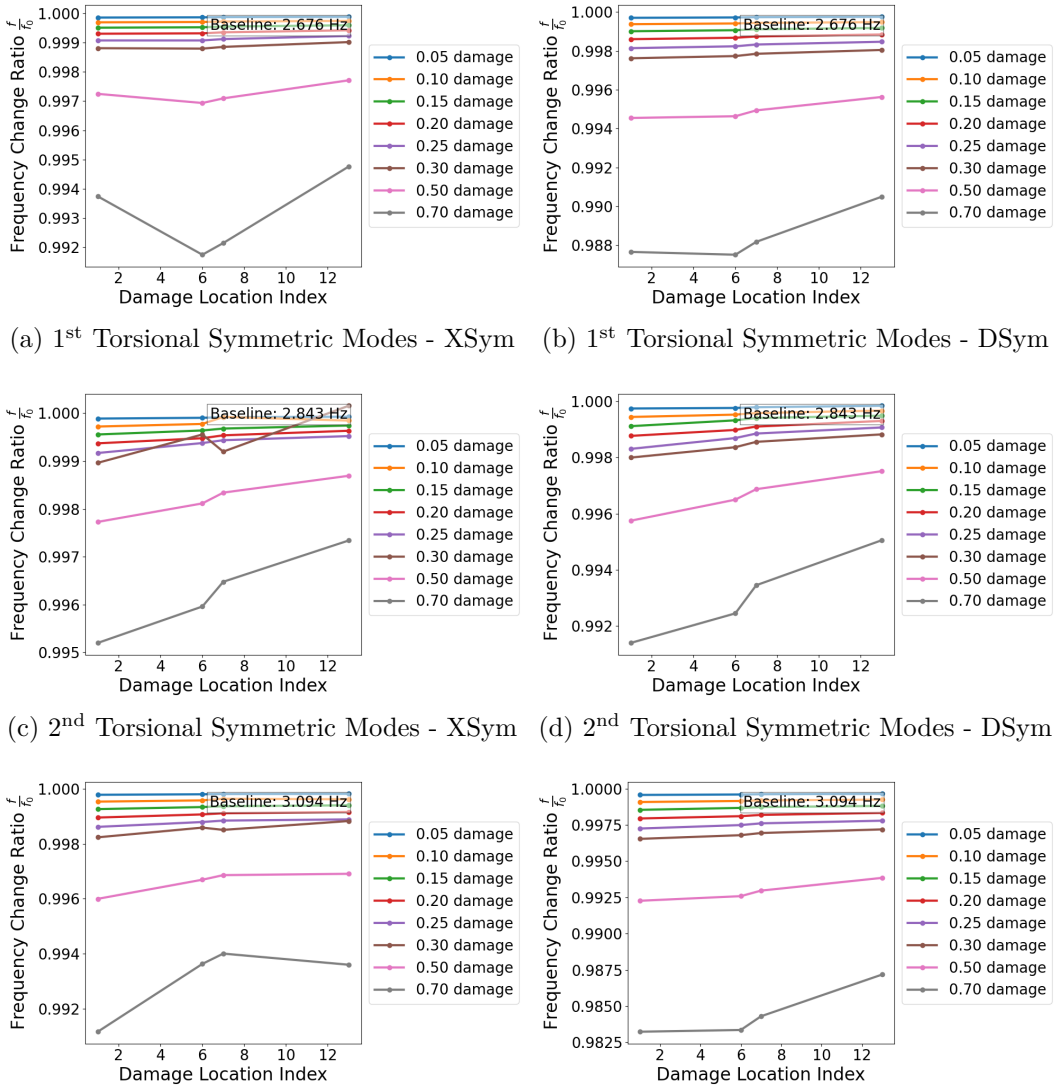


(e) 3rd Torsional Asymmetric Modes - XSym (f) 3rd Torsional Asymmetric Modes - DSym

Figure 5.18: FREQ11 Sing Asymmetrical Torsional Modes of Vibration

As per the lateral modes, the FREQ34 has more consistent results, as presented in Figures 5.19 and 5.20 for the symmetrical and asymmetrical versions. However, these results only consider four of the 13 possible damage locations, limiting the analysis. For both first modes (Figures 5.18a and 5.18b), damage levels of 0.50 and below seem unaffected by location. Moreover, the lowest natural frequencies are at the tower for the XSym pattern. This differs from the DSym trends, which are horizontal between the anchor and the tower. For the second mode, the natural frequencies decrease as the damage is applied in the lower location index, which again could become erroneous if the remaining points were added to the plot. For the third mode, the symmetrical mode with the XSym pattern has a more significant shift in natural frequencies at the midspan than at the towers, which is unusual compared to other modes. As a result, no additional patterns can be identified

from these results due to the limited data points and the discrepancies between the FREQ11 and FREQ34 models.

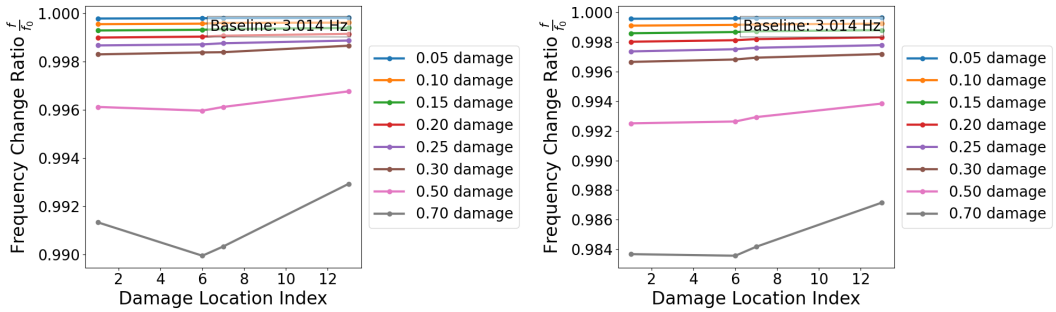


(a) 1st Torsional Symmetric Modes - XSym (b) 1st Torsional Symmetric Modes - DSym

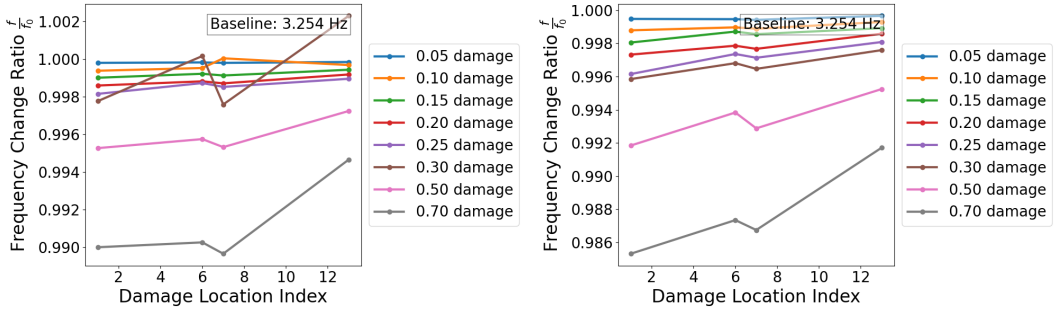
(c) 2nd Torsional Symmetric Modes - XSym (d) 2nd Torsional Symmetric Modes - DSym

(e) 3rd Torsional Symmetric Modes - XSym (f) 3rd Torsional Symmetric Modes - DSym

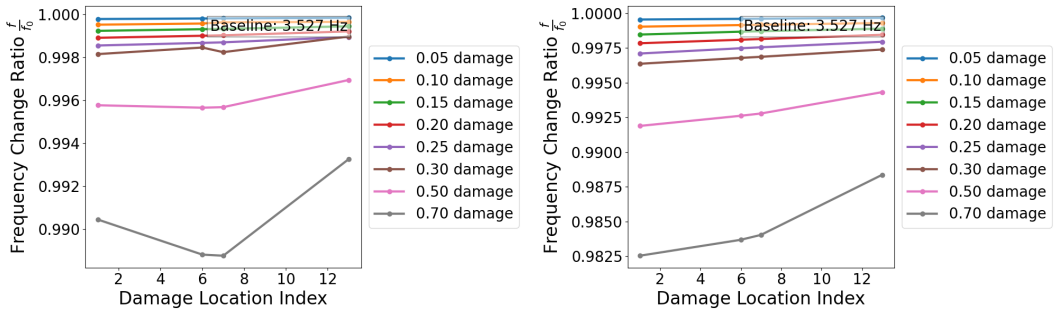
Figure 5.19: FREQ34 Sing Symmetrical Torsional Modes of Vibration



(a) 1st Torsional Asymmetric Modes - XSym (b) 1st Torsional Asymmetric Modes - DSym



(c) 2nd Torsional Asymmetric Modes - XSym (d) 2nd Torsional Asymmetric Modes - DSym



(e) 3rd Torsional Asymmetric Modes - XSym (f) 3rd Torsional Asymmetric Modes - DSym

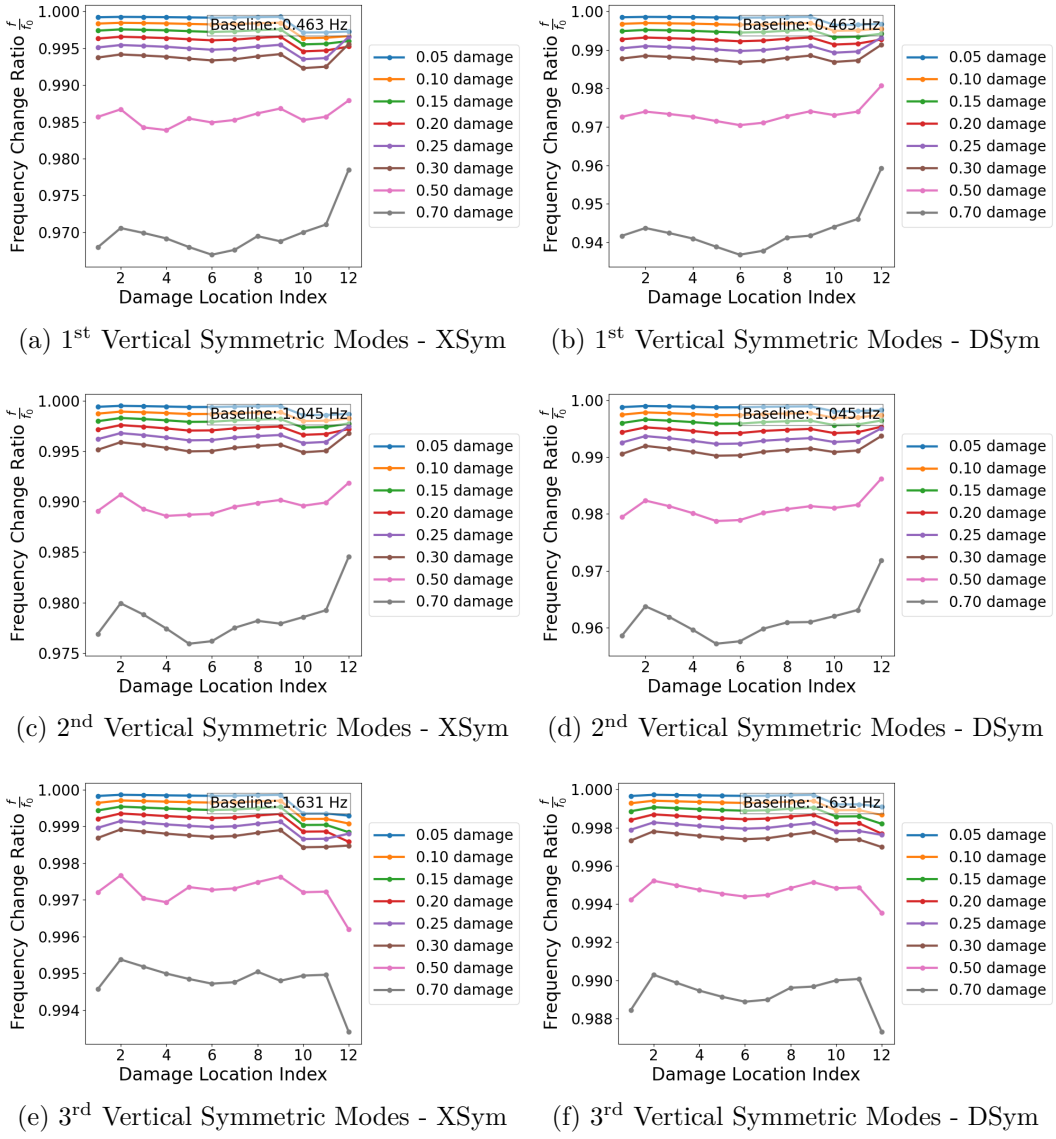
Figure 5.20: FREQ34 Sing Asymmetrical Torsional Modes of Vibration

5.1.4 Doub WiresNumber

With the results for the Sing cases presented, an equivalent analysis can be made for the Doub cases, in which damage is applied to two adjacent main cable spans, resulting in either four or eight damaged spans depending on the case symmetry pattern. The results obtained from this analysis are expected to provide insight into how the proximity and interaction between two damaged zones influence the overall dynamic response of the structure. An eventual machine learning model would require this sort of result accessible through an established database to determine how natural frequency shifts accumulate with damage in different locations. Given their location-based similar nature, the results for the Doub cases are expected to be similar to those of the Sing

cases in this analysis. Therefore, the main purpose of this section is to highlight the similarities and differences between Sing and Doub cases on significant observations. Based on the data sampled in the previous sections, the natural frequencies for the Doub cases are expected to be consistently lower than those observed for the corresponding Sing cases.

The first set of findings is for the vertical symmetrical modes, presented in Figure 5.21.



(a) 1st Vertical Symmetric Modes - XSym (b) 1st Vertical Symmetric Modes - DSym

(c) 2nd Vertical Symmetric Modes - XSym (d) 2nd Vertical Symmetric Modes - DSym

(e) 3rd Vertical Symmetric Modes - XSym (f) 3rd Vertical Symmetric Modes - DSym

Figure 5.21: FREQ11 Doub Symmetrical Vertical Modes of Vibration

For the most part, the results are as expected, with the overall trends closely resembling those observed in Figure 5.14. Moreover, the more extreme values in the Sing cases results, such as at the anchor, the tower, and the midspan, are attenuated in the equivalent Doub values. For example, while the location index 1 in Sing is generally associated with a more extreme value, the location index 2 is not (Figures 5.14 and 5.15). Therefore, combining these two damage cases results in a

proportionally smaller change in the Doub case compared to the sum of the changes observed in the Sing cases. However, these results make it difficult to determine the comparison between the Doub cases and the Sing cases. To solve this, an additional comparison is provided, which depicts the frequency change ratio of the Doub cases against the combination of the two individual Sing cases encompassed within the Doub case. The combination method is presented in Equation 5.1, a simple addition of the differences from the two Sing cases, denoted left and right. This equation is proposed to examine the effect of combined damage in the cables on the natural frequencies of the deck. With this equation, Figure 5.22 displays every damage location index, every damage level, and both XSym and DSym patterns of the first three vertical symmetric modes.

$$\frac{f_{Comb}}{f_{Base}} = 1 - \left(1 - \frac{f_{Sing_{left}}}{f_{Base}}\right) - \left(1 - \frac{f_{Sing_{right}}}{f_{Base}}\right); f_{Comb} = f_{Sing_{left}} + f_{Sing_{right}} - f_{Base} \tag{5.1}$$

Where $\frac{f_{Comb}}{f_{Base}}$ is the combined frequency change ratio, $\frac{f_{Sing_{left}}}{f_{Base}}$ is the left Sing frequency change ratio, and $\frac{f_{Sing_{right}}}{f_{Base}}$ is the right Sing frequency change ratio. To clarify, left is the case with the lowest damage location index of the two Sing cases involved, while right corresponds to the Sing case with the highest damage location index. The linear regression on the Doub results in Figure 5.22 present a slope close to unity, indicating that the combination method of adjacent Sing cases adequately approximates the Doub cases. With the slope of 0.9193, it can be noticed that the frequency change ratio of the Doub cases is almost always greater than the equivalent combined Sing cases. This implies that combining different damage levels reduces the total change that would occur in both individual cases. Furthermore, the high correlation coefficients indicate a robust relation between the combined and Doub shifts. This demonstrates that a machine learning model could be built to obtain other damage distribution relations.

In contrast, the FREQ34 frequency change ratio figures are provided in Figure I.9 in Appendix I, as they display the same combination pattern as the FREQ11 model. However, due to computation time limitations, there is insufficient variation in the number of damage location indices to establish a similar combined analysis.

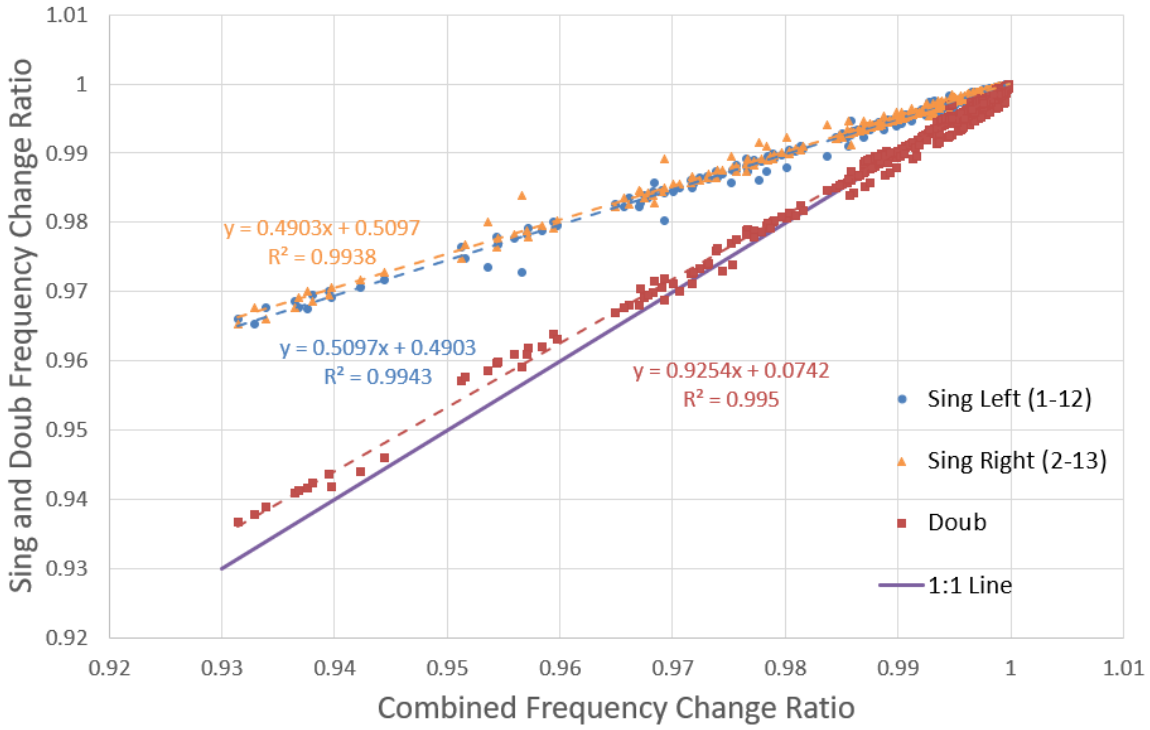
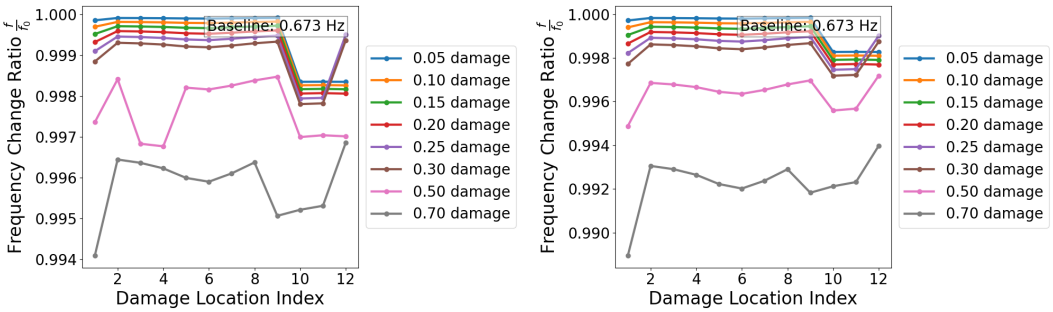
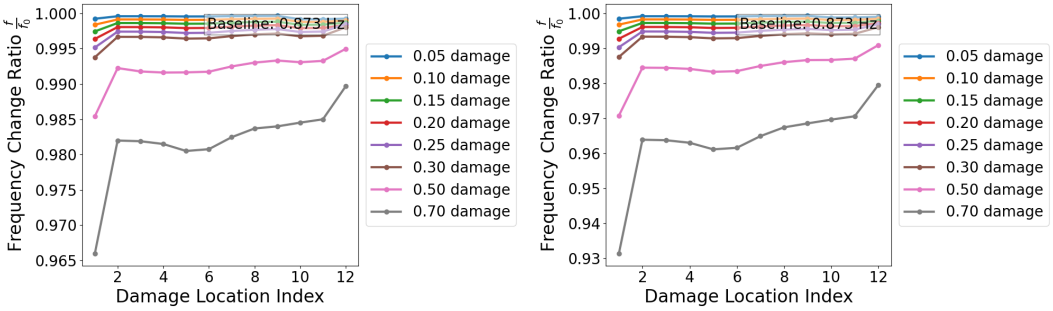


Figure 5.22: FREQ11 Symmetrical Vertical Modes of Vibration - Combined Analysis

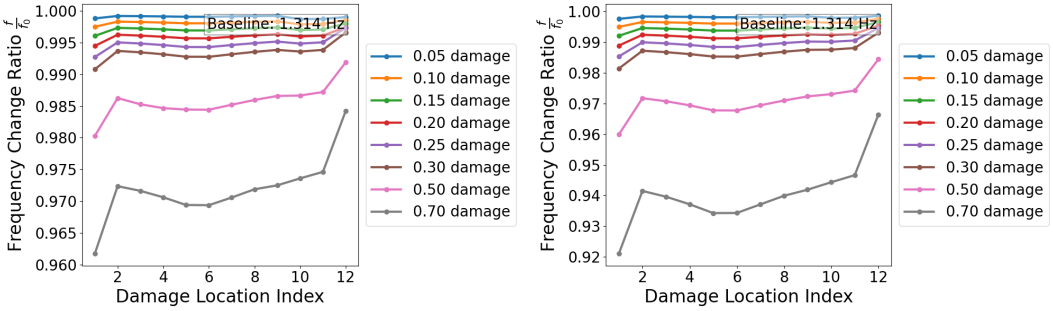
With a similar presentation, Figure 5.23 displays the vertical asymmetrical vibration modes with Doub cases. The same conclusions can be drawn for the vertical symmetrical modes when compared to the Sing cases. Additionally, the equivalent combined analysis is available in Figure 5.24, which only reinforces the previous findings about damage combinations with comparable slope and coefficient of correlation. The FREQ34 results are presented in Figure I.10 in Appendix I, but the combined analysis is not possible for the same reasons previously stated.



(a) 1st Vertical Asymmetric Modes - XSym (b) 1st Vertical Asymmetric Modes - DSym



(c) 2nd Vertical Asymmetric Modes - XSym (d) 2nd Vertical Asymmetric Modes - DSym



(e) 3rd Vertical Asymmetric Modes - XSym (f) 3rd Vertical Asymmetric Modes - DSym

Figure 5.23: FREQ11 Doub Asymmetrical Vertical Modes of Vibration

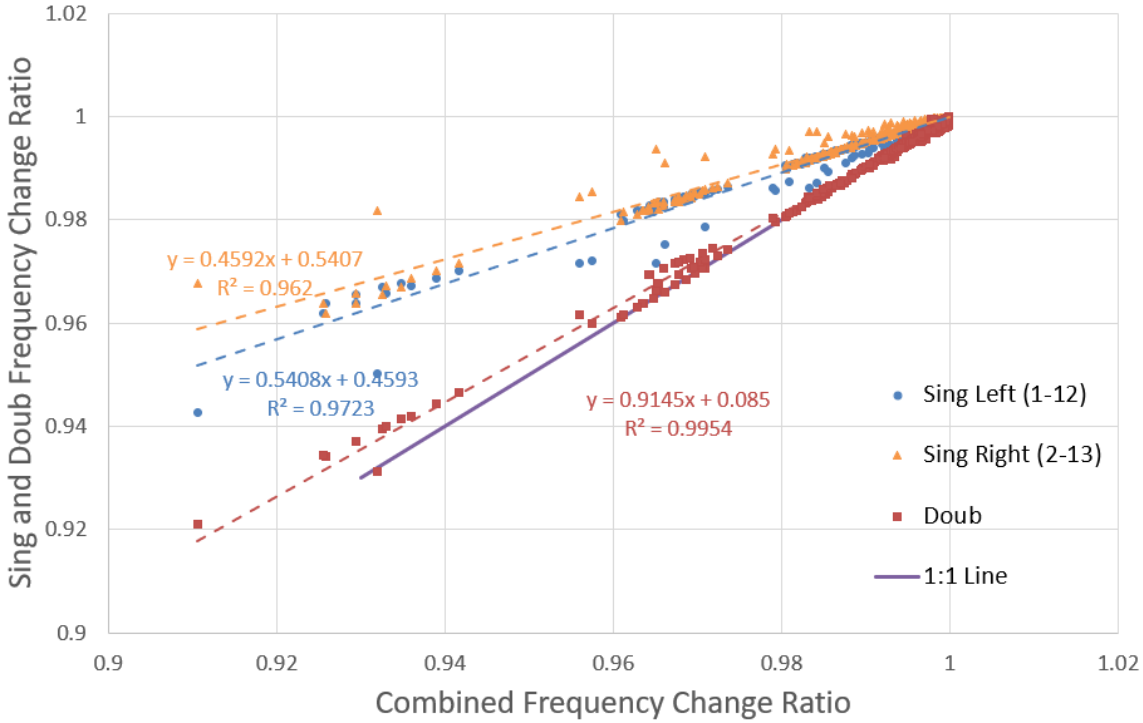


Figure 5.24: FREQ11 Asymmetrical Vertical Modes of Vibration - Combined Analysis

Inversely, the lateral modes demonstrate no association between the combined frequency change ratio and the Doub frequency change ratio. This can be observed in Figure 5.25, which presents the combined analysis of the FREQ11 lateral symmetrical vibration modes based on the data from Figure I.11 in Appendix I. As per the Sing cases, the lateral symmetrical modes are not promising to establish a statistical relationship with damage application. This is significantly reinforced by the coefficient of correlation of almost 0 between the Doub and the combined frequency change ratio in Figure 5.25. Therefore, these results reaffirms the doubts previously raised about the lateral modes. Finally, the equivalent FREQ34 data is available in Figure I.12 in Appendix I.

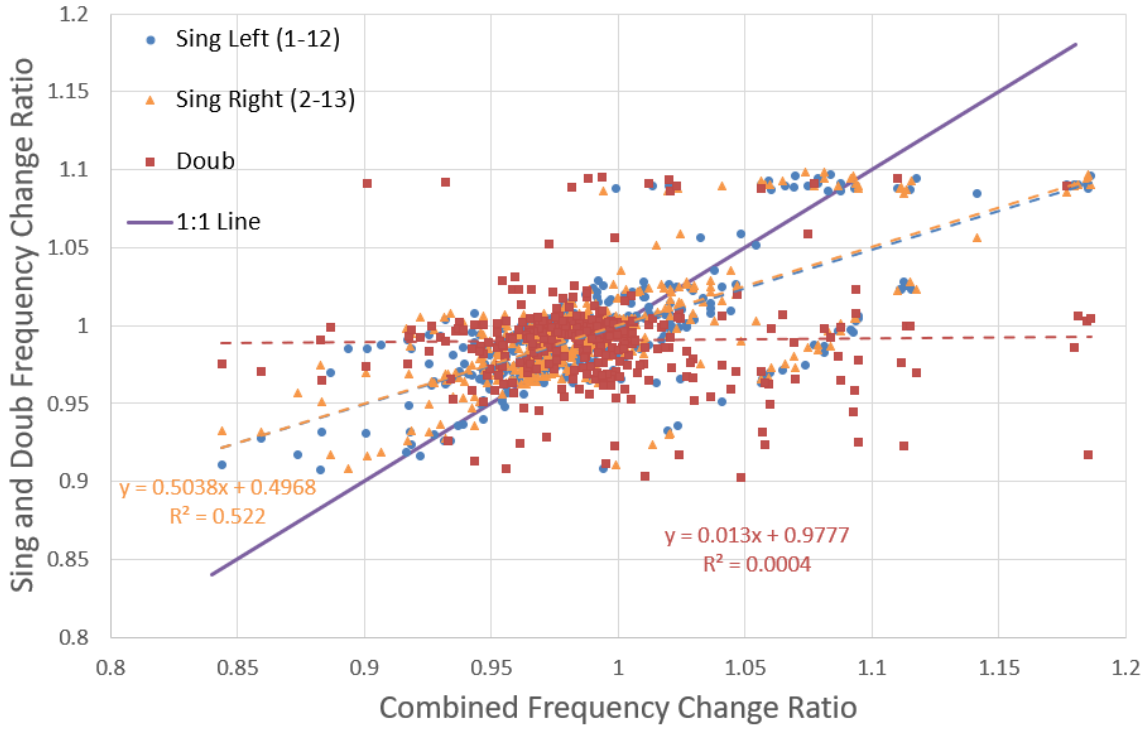


Figure 5.25: FREQ11 Symmetrical Lateral Modes of Vibration - Combined Analysis

Since the lateral asymmetrical modes have fewer points due to the filtering for the mode identification algorithm errors, the combined analysis has fewer points, as presented in Figure 5.26. This figure is based on the data available in Figure I.13 in Appendix I, which demonstrates that the shifts in the lateral modes' natural frequencies do not correlate with damage locations. The FREQ34 equivalent data is available in Figure I.14 in Appendix I.

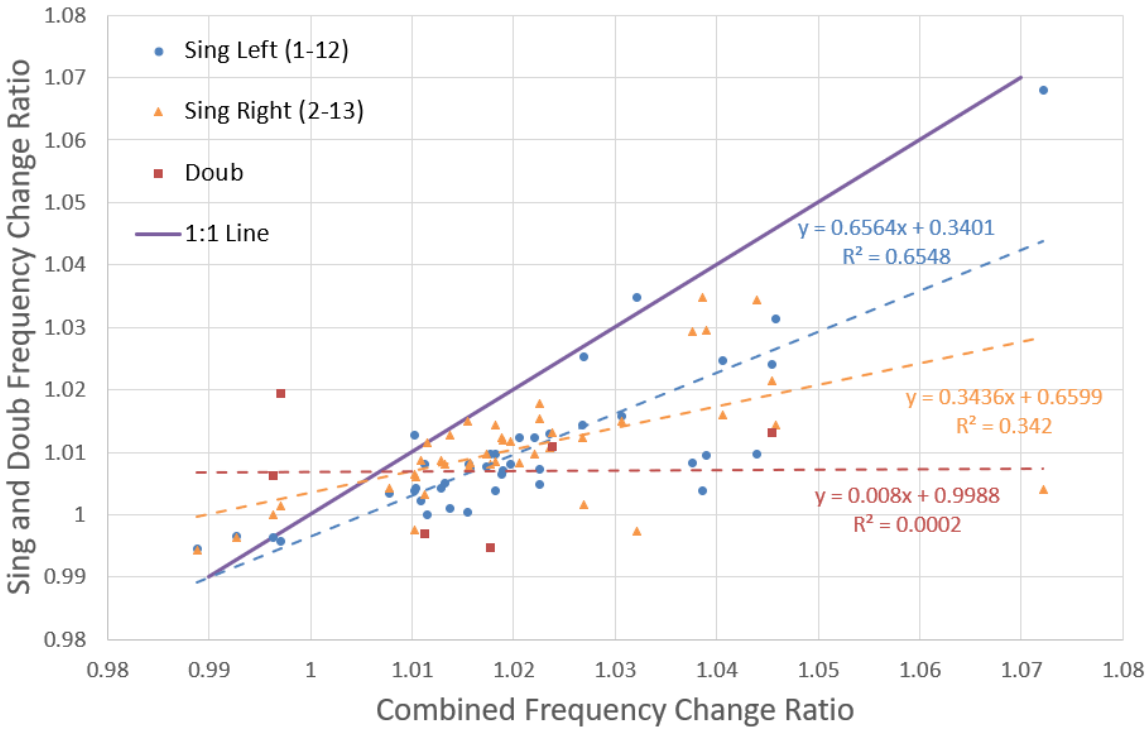
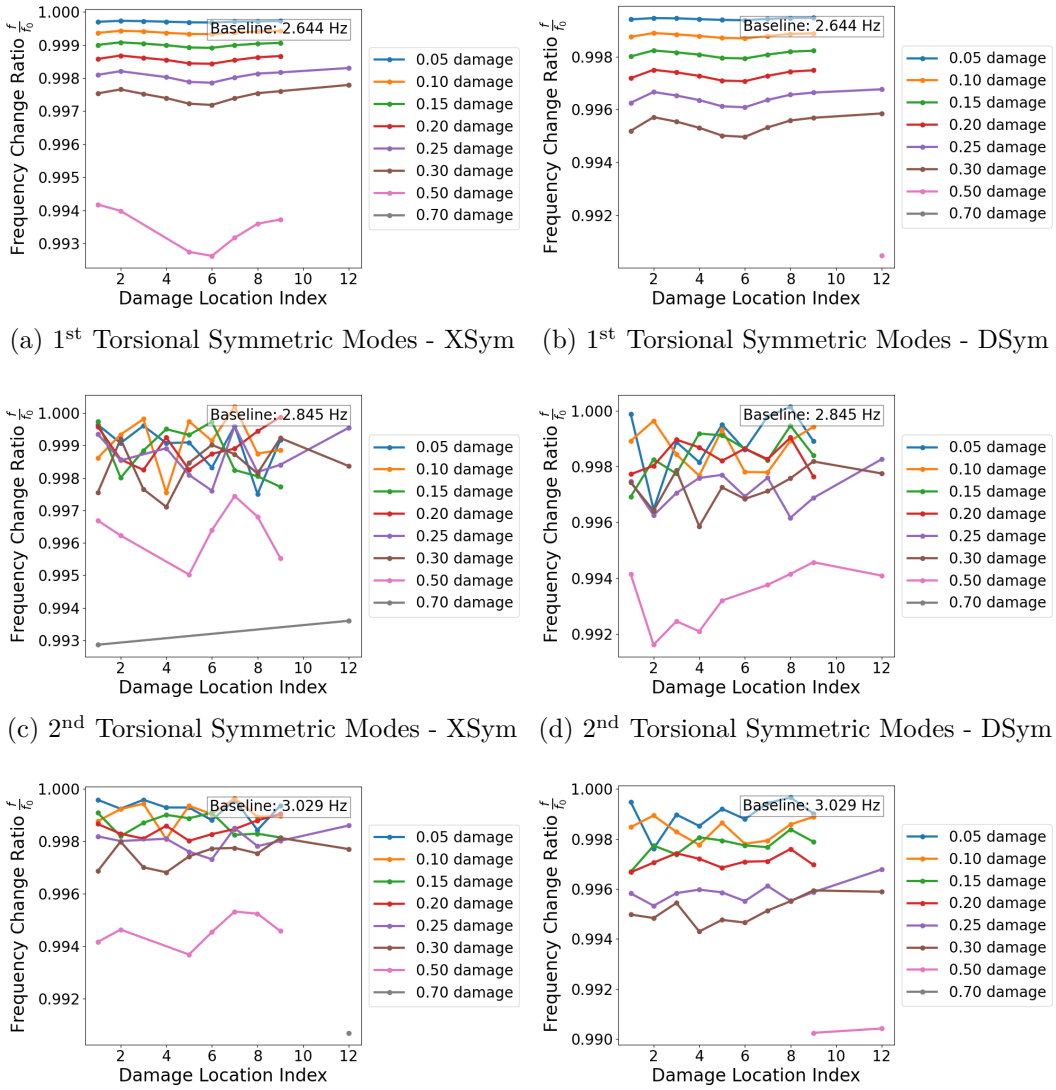


Figure 5.26: FREQ11 Asymmetrical Lateral Modes of Vibration - Combined Analysis

As observed in the previous cases, the torsional modes are an in-between of the vertical and the lateral modes in correlation with the applied damage. For the Sing cases, only the first torsional symmetrical mode presents consistent results for FREQ11, whereas the second and third modes are highly inconsistent. This is the pattern observed in Figure 5.27 below.



(a) 1st Torsional Symmetric Modes - XSym (b) 1st Torsional Symmetric Modes - DSym

(c) 2nd Torsional Symmetric Modes - XSym (d) 2nd Torsional Symmetric Modes - DSym

(e) 3rd Torsional Symmetric Modes - XSym (f) 3rd Torsional Symmetric Modes - DSym

Figure 5.27: FREQ11 Doub Symmetrical Torsional Modes of Vibration

Isolating only the first mode for a combined analysis, the results present a slope of just above one and a very high coefficient of correlation, presented in Figure 5.28. These results indicate that the first torsional symmetrical mode could be used as an additional indicator of combined damage locations by combining natural frequency shifts calculated through Equation 5.1. The FREQ34 figures display notable differences between the Sing and Doub cases, as presented in Figure I.15 in Appendix I; however, the limited number of damage location indices prevents a thorough analysis of the cause.

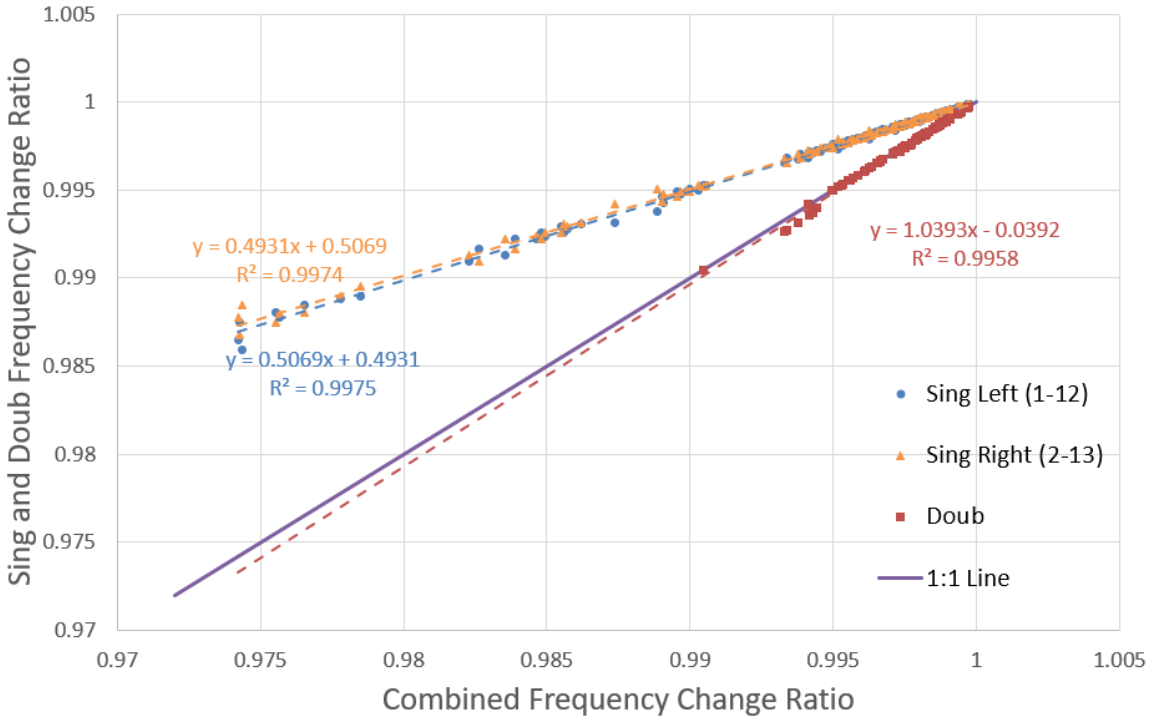
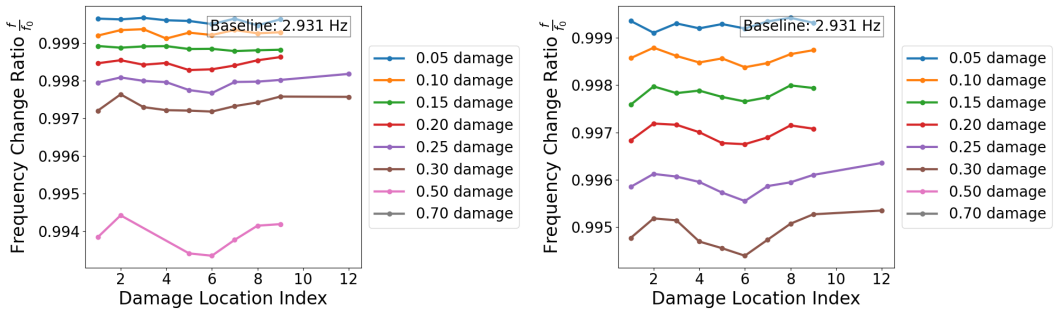
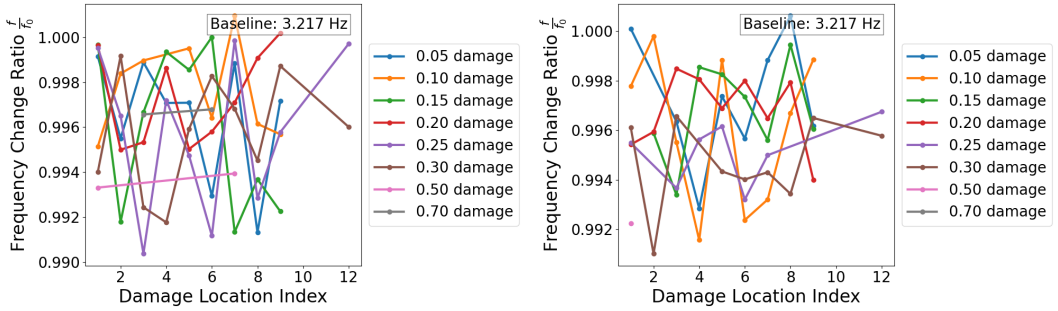


Figure 5.28: FREQ11 Symmetrical Torsional Modes of Vibration - Combined Analysis

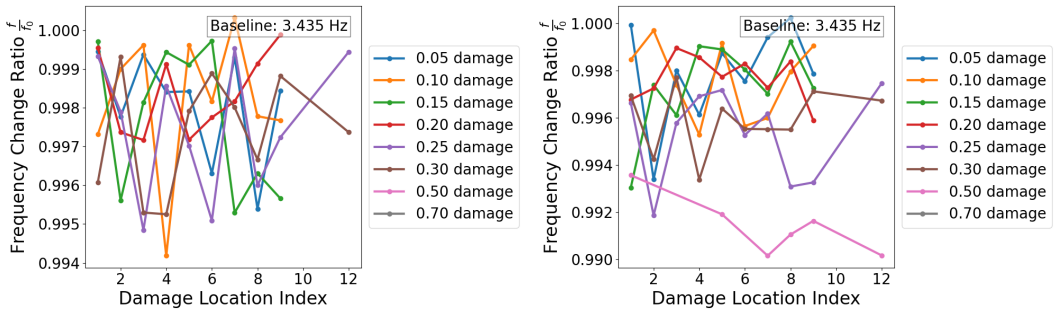
For the torsional asymmetrical mode, presented in Figure 5.29, only the first mode has a correlation between the changes in natural frequencies and the applied damage (location and level) as per the symmetrical results (Figure 5.27). Performing the equivalent combined analysis on that single mode reveals a very high coefficient of correlation with a slope of almost unity, as observed in Figure 5.30. These high coefficients imply that the combination of multiple damage cases could be analyzed against a case with all damage as a single case to isolate the individual contributions of each damage location. The FREQ34 results are available in Figure I.16 in Appendix I.



(a) 1st Torsional Asymmetric Modes - XSym (b) 1st Torsional Asymmetric Modes - DSym



(c) 2nd Torsional Asymmetric Modes - XSym (d) 2nd Torsional Asymmetric Modes - DSym



(e) 3rd Torsional Asymmetric Modes - XSym (f) 3rd Torsional Asymmetric Modes - DSym

Figure 5.29: FREQ11 Doub Asymmetrical Torsional Modes of Vibration

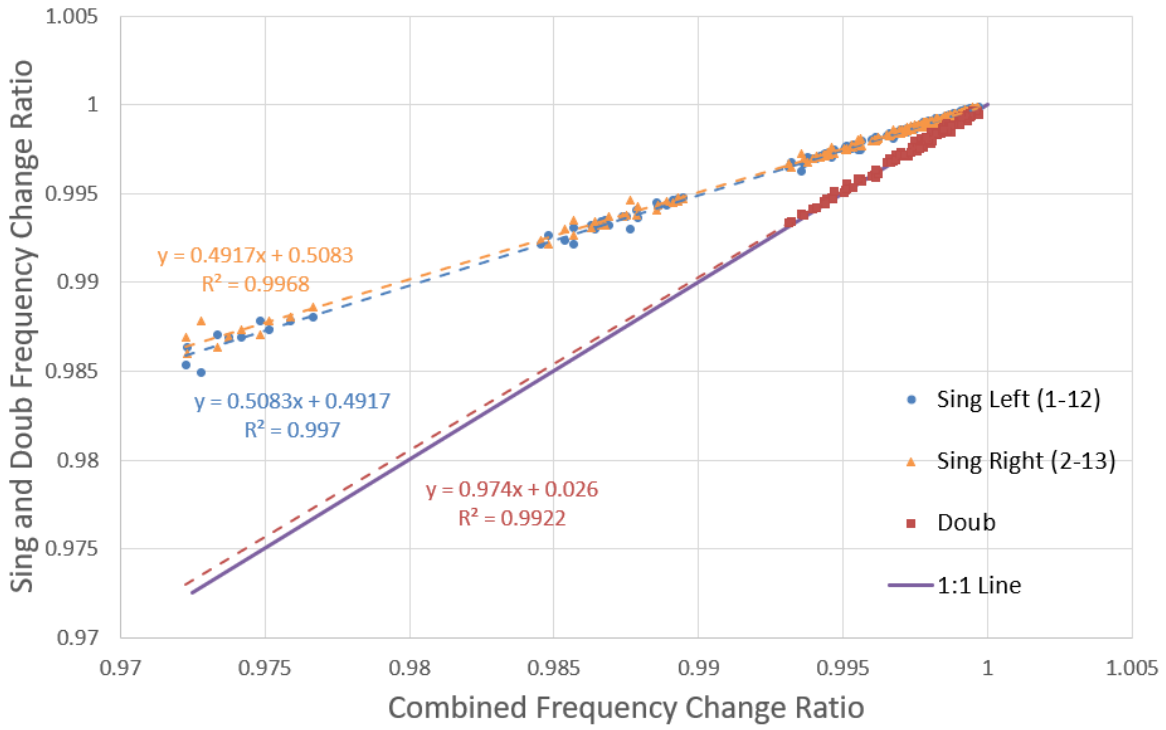


Figure 5.30: FREQ11 Asymmetrical Torsional Modes of Vibration - Combined Analysis

5.1.5 Discussion of Natural Frequencies Results

In general, the results in this section clearly demonstrate that, as expected, the application of damage to the wires forming the main cables creates noticeable changes in the natural frequencies associated with the deck's vibration modes. Consistently throughout all the investigated cases, the vertical vibration modes present the best correlations between shifts in natural frequencies and damage location and intensity. On the other hand, most of the lateral modes of vibration of the deck do not present coherent results when compared against damage location or intensity. Ultimately, the torsional modes can be adequate indicators of damage application, provided that only the first symmetric and first asymmetric modes are considered.

Overall, it is persistently noticed that increases in damage intensity always results in a decrease of the natural frequencies for the deck vibration. Regarding damage location, more specifically for the Doub and Sing cases, there are often some visual discontinuities in the natural frequencies for damage applied at the anchor, near the towers, and at the midspan. These locations are critical, as they could reveal if damage exists there more easily through a machine learning algorithm. However, the most significant finding in this section pertains to comparing the natural frequencies between the Sing and Doub cases. More specifically, this involves comparing the natural frequencies of two individual Sing cases, where the damage indices are adjacent to each other, to the Doub

case counterpart that accounts for these two damage locations. Combining the natural frequencies of the Sing cases results in an apparent correlation to the natural frequencies in the Doub case. The relation between these natural frequencies is linear with a slope close to unity (Equation 5.1 for the combination method of natural frequencies). This is the case for the vertical and the first torsional (symmetric and asymmetric) modes, but the same correlation is not found in the lateral mode results. While the results are only available for neighbouring damage locations for Sing cases to Doub cases, similar dependencies are expected to be obtained for other location arrangements.

Overall, the changes in the natural frequencies for the tested cases are relatively low, but significant enough for a numerical model to detect the changes over time. Considering, for example, the Sing cases presented, the most significant decreases in the natural frequencies are 3.47% for the vertical symmetric modes, 5.72% for the vertical asymmetric modes, 9.22% for the lateral symmetric modes, 0.57% for the lateral asymmetric modes, 1.52% for the torsional symmetric modes, and 2.15% for the torsional asymmetric modes. Therefore, similar environmental conditions would be necessary to detect these changes, as damage shifts could otherwise cause the changes to go unnoticed. However, further research in this field could motivate the installation of accelerometer sensors along the length of the deck to monitor the changes in the natural frequencies of the modes of vibration, providing a potential indication of the damage distribution in the bridge. Naturally, a machine learning model must be established before providing any accurate analysis. This model should be able to run damage application trials on the defined model, similar to the calibrating algorithms defined throughout Chapter 4. By adding each performed case to a database on which it can base its next prediction, the goal would be to establish patterns that allow it to determine what combination of damage locations and intensities are behind a set of shifts in natural frequencies. The connection to the model would also allow it to test the effect of its predictions to converge to a solution. At this stage, it should first be determined if other patterns can be established between Sing cases and other damage layouts (i.e. Doub with a spacing between damage locations or Triple continuous damage locations). The current program used throughout this research can be directly used for this purpose, making the development of further test series trivial. If the relationships obtained are consistent for other damage patterns, then the development of a learning model would be justified. Moreover, a key advantage of the current program is its flexibility, as it can be applied directly to various suspension bridges without requiring any code modifications. By using the inputs described in Section 4.2.1, the geometry and properties can be adjusted to represent different suspension bridges. Therefore, it would be interesting to determine if the correlations obtained here hold in other bridges.

Regarding the comparison between the FREQ11 and FREQ34 models, it is determined that for the modes in which strong relationships are obtained between damage and natural frequency shifts are obtained, the improvement in stability and precision in the FREQ34 model does not justify its increase in computing time. It should be mentioned here that the latter takes approximately five times longer to compute and provides very similar results to the FREQ11 model. Overall, this implies that for the purposes encompassed in this research, the simplification of the main cable bundle can be approximated as an equivalent single wire main cable.

Another critical point revealed through the analysis of these results is the considerable limitation posed by instabilities in the finite element model. When test cases were aborted, the step stabilization parameters had to be adjusted, which likely explains some of the discontinuities observed in the results. In Abaqus, identifying the source of instabilities in larger model is particularly challenging, and the solutions for individual cases are often counterintuitive. For instance, the Frequency step, as defined in Section 3.4, requests the first 50 natural frequencies of the model, causing many failures across test cases. To address this issue, failed models were relaunched with a request for the first 51 natural frequencies. If this also failed, the request was incremented to 52 frequencies, and so on. This iterative process was one of the primary methods used to recover aborted cases. Another approach involved making a minuscule adjustment to the applied damage magnitude. For example, a failed case with a damage of 0.70 in some cables could be relaunched with the damage set to 0.700001 or 0.699999 (using variations of 1e-6, which matches the decimal precision in Abaqus). Although structurally insignificant, this change successfully resolved many failures. A combination of these methods was applied to the most problematic cases, while the remaining cases were addressed through sweeping adjustments of the step stabilization parameters. While these fixes are not particularly complex from a technical standpoint, they quickly become tedious in practice. In total, over 4,000 individual trials were required to resolve all 1,459 cases in the FREQ11 series. As a potential improvement, an interesting alternative would be to model the cables as beams with low bending stiffness instead of trusses, which could provide better model stability by allowing temporary residual moments in the nodes between cable elements during the model stabilization.

Nonetheless, considering the results presented in this section, it may not be concluded that the damage locations could be identified as being on a specific side of the bridge in any of the two directions. For example, if a learning model is developed and determines that the damage is located near one of the towers based on shifts in the natural frequencies, it would be impossible to identify which specific tower is affected. The double symmetry within this type of infrastructure

is the reason for this behaviour. This is the primary motivation behind the analysis of the deck vertical displacement in the next section. By comparing deck displacements across different damage symmetry patterns, it is expected that the affected area of the bridge (in terms of quarters) can be identified.

5.2 Deck Vertical Displacement Analysis

The vertical displacement from the starting model position is examined for this next set of results. In reference to the steps explained in Section 3.4, this is the total displacement at the end of the Vertical Loading step. This is investigated for a subset of test cases presented throughout Section 5.1, with some limitations. The purpose of presenting and analyzing these measurements is to demonstrate that the location of the applied damage, whether on the north or south side of the midspan, has a noticeable influence on the vertical displacements along the deck. Then, if a damage location index (index 1-13 on a quarter of the bridge) is identified through the natural frequency shifts, the exact location on either side of the bridge could be determined with the changes in the elevation of the deck. Because the deck is calibrated with pretension in the hangers (Section 4.3.2), the displacement should remain relatively small in the baseline undamaged model, as presented in Figure 5.31. This calibration ensures that the final bridge coordinates can be used for modelling. Note that in that same figure, the deck elevation corresponds to the Initial step and is shifted by 1.6567 m so that the elevation at the anchor is 0 m instead of the actual -1.6567 m in the model, which accounts for the anchor bearing height.

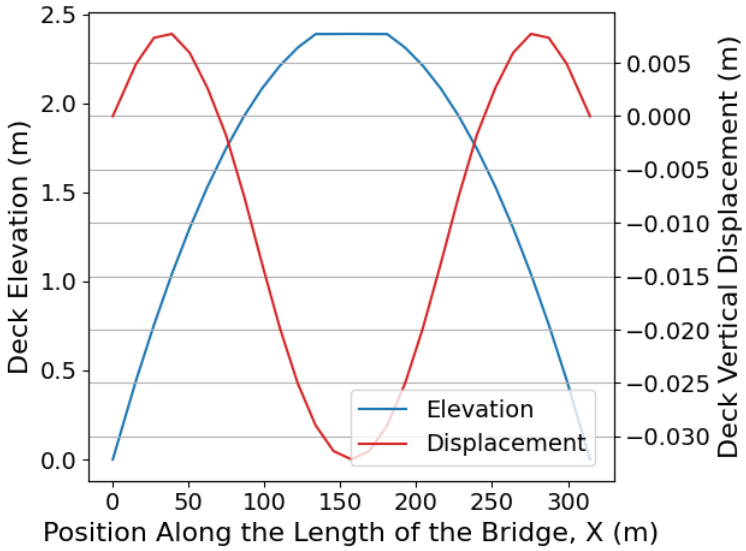
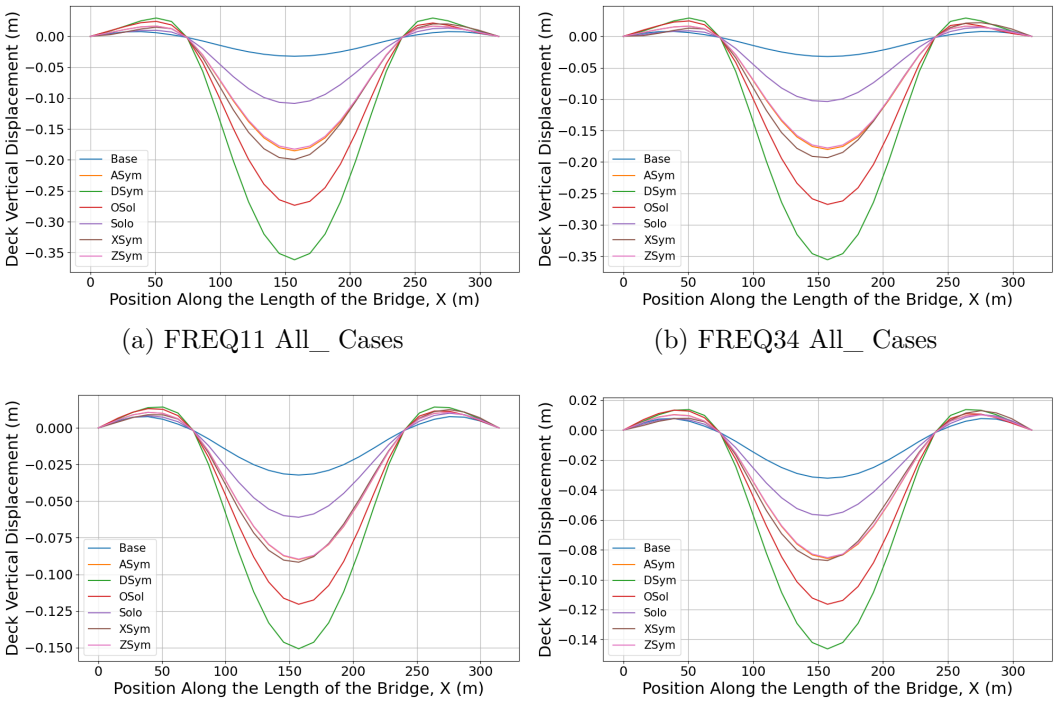


Figure 5.31: FREQ11 Baseline Deck Elevation and Displacement

However, since the displacements are not zero, a better method of displaying results for the displacement should be by scaling the obtained displacements per damage case with the baseline displacements. This is done as a simple ratio of the damage case over the baseline case on a point-by-point basis. These results, on their own, serve merely as proof of concept for the changes that occur with variations in the damage location index. While general patterns can be observed, detecting complex behaviours is not expected, as that aspect is better suited for analysis by a machine learning model. The main goal in this section is only to present differences between the different symmetry patterns for the damage (LocationPatterns defined in Section 4.4.1). The analysis is separated into two result subsets: 1) a damage location index-independent set of results (All_ and Quad cases) and 2) a damage location index-dependent set of results (Sing and Doub). Only the 0.30 damage level cases are presented for all analyses, as the other results are scaled versions of the same curves.

For the damage location index-independent results, all the symmetry patterns are presented in Figure 5.32, including the FREQ11 and the FREQ34 models. Note that on these figures, the location of the towers are at the point where the deck vertical displacement crosses zero, and the global shape of the deck can be attributed to a simply supported beam with intermediate roller support and a distributed load. One of the first observations that can be made from these figures is that, as expected, the vertical displacement at the midspan becomes progressively more significant as the number of affected cables increases. This behaviour occurs because the introduction of damage reduces the cable's stiffness, leading to an increase in displacements. The deck displacement for the ASym and ZSym cases overlaps in most figures, suggesting that if the damage is balanced on both sides (north/south), the specific side (east/west) on which the damage is located does not appear to have an effect. However, this considers the centre of the deck and not the ends of the girder members, which could also be investigated to determine the effect as a rotation of the deck. Nevertheless, the main conclusion consistent within all presented figures (5.32) is that the downward vertical displacements in the main span are always more significant on the side on which damage is applied. Considering that the Solo case has damage on the first half of the bridge (south - leftmost on the figures), the Solo and the XSym cases have more important displacements on the left side, whereas the OSol case is the opposite. This again confirms that applying damage on a side of the bridge causes more significant displacements on that side of the deck.



(a) FREQ11 All_ Cases

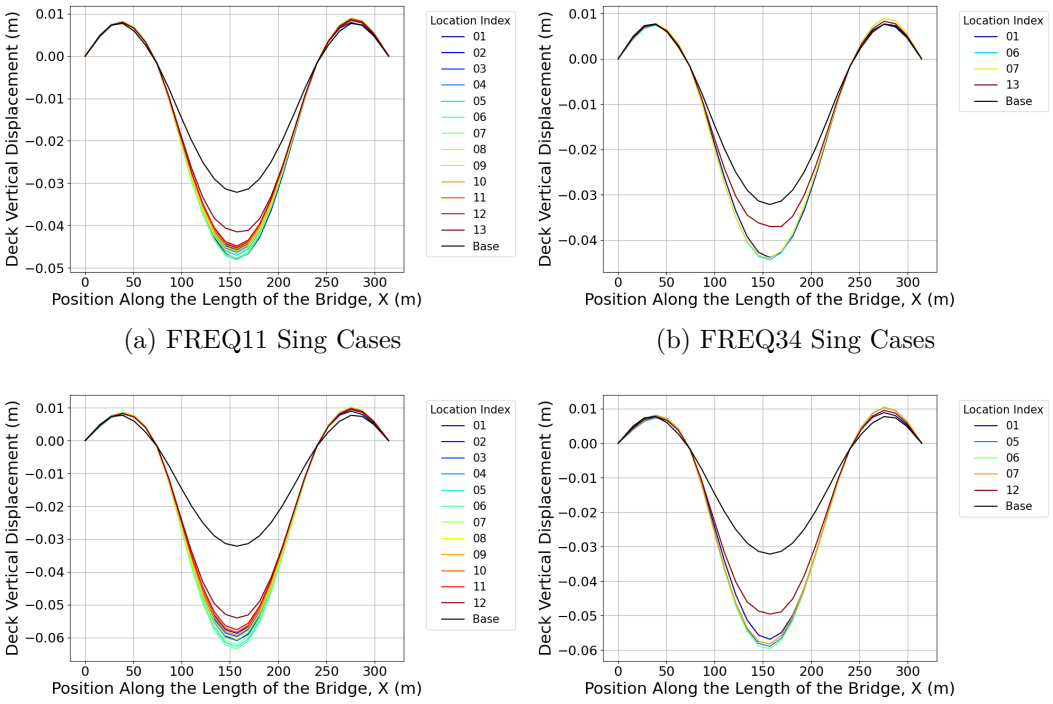
(b) FREQ34 All_ Cases

(c) FREQ11 Quad Cases

(d) FREQ34 Quad Cases

Figure 5.32: Deck Vertical Displacement for Location Index Independent Cases with 0.30 Damage

Thereafter, the damage location index dependent results present only the XSym and the DSym symmetry patterns, available in Figures 5.33 and 5.34, respectively. The aforementioned observations regarding the more significant displacements on the affected side of the bridge are also noticed in the XSym results. While most location index curves are consistently spaced, the midspan behaviour of the Sing location index 13 and the Doub location index 12 curves display an inverted tendency. Furthermore, applying damage solely to the main cable span closest to the anchor (Sing) creates a slightly different displacement behaviour, demonstrating symmetry around the midspan, which contrasts with other locations.



(a) FREQ11 Sing Cases

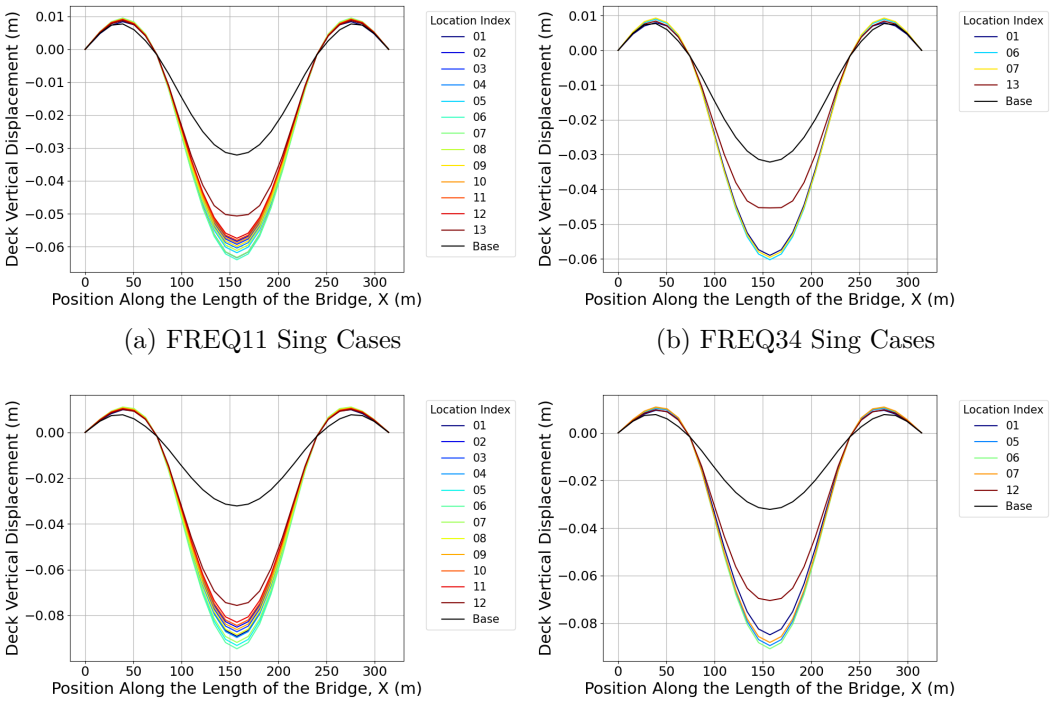
(b) FREQ34 Sing Cases

(c) FREQ11 Doub Cases

(d) FREQ34 Doub Cases

Figure 5.33: Deck Vertical Displacement for Location Index Dependent Cases with 0.30 Damage - XSym

The following conclusion from these results is determining which damage location has the most significant impact on the deck displacement, with a clearer representation available in Figure 5.34, which presents the DSym symmetry pattern. This allows for a better comparison between the curves since there is less overlapping due to symmetry. It can be observed that damage applied to the spans closer to the tower cause the largest displacements, while damage in the spans nearer to the midspan result in the least displacement. This can be easily explained by the fact that in this type of structure, the cable's tension is indicative of its angle, with sections closer to the tower experiencing the greatest tension. Therefore, a reduction of stiffness in these wires has a proportionally more significant impact on the displacement of the cables, leading to a more considerable sag in the deck.



(a) FREQ11 Sing Cases

(b) FREQ34 Sing Cases

(c) FREQ11 Doub Cases

(d) FREQ34 Doub Cases

Figure 5.34: Deck Vertical Displacement for Location Index Dependent Cases with 0.30 Damage - DSym

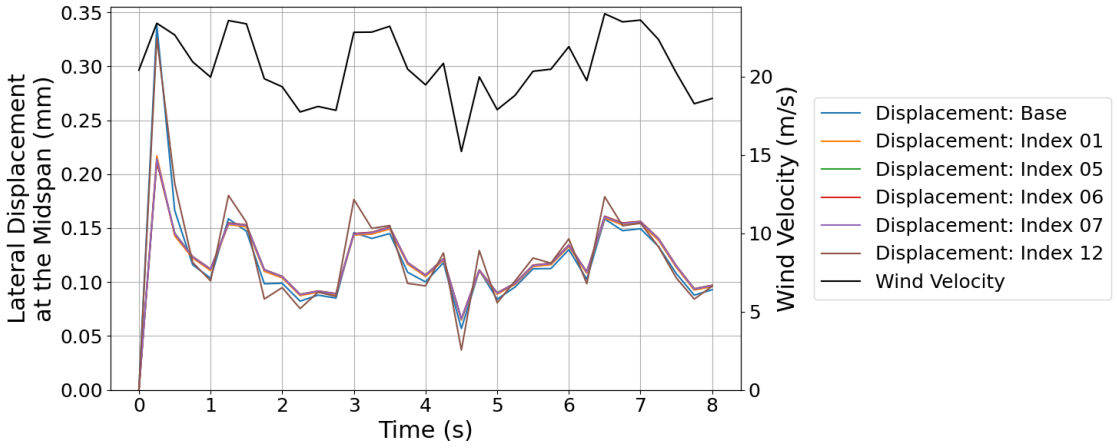
Overall, this section demonstrates that the level of vertical displacement along the length of the deck could indicate the general location of the applied damage. Building on the discussion in the previous subsection, this data demonstrates that it is possible to identify the damaged side of the bridge based on these results. Again, installing accelerometer sensors with displacement reporting along the length of the bridge would be required to accomplish this determination. However, when damage is identified from shifts in the natural frequencies, then the displacement of the deck could be examined to determine which side of the bridge the damage is applied on. Additionally, some of the results presented in this section highlight that the location of the applied damage directly influences the vertical displacement of the deck. This makes it a viable option for validating the findings of a machine learning model trained on shifts in natural frequencies.

5.3 Wind Scenarios Output Analysis

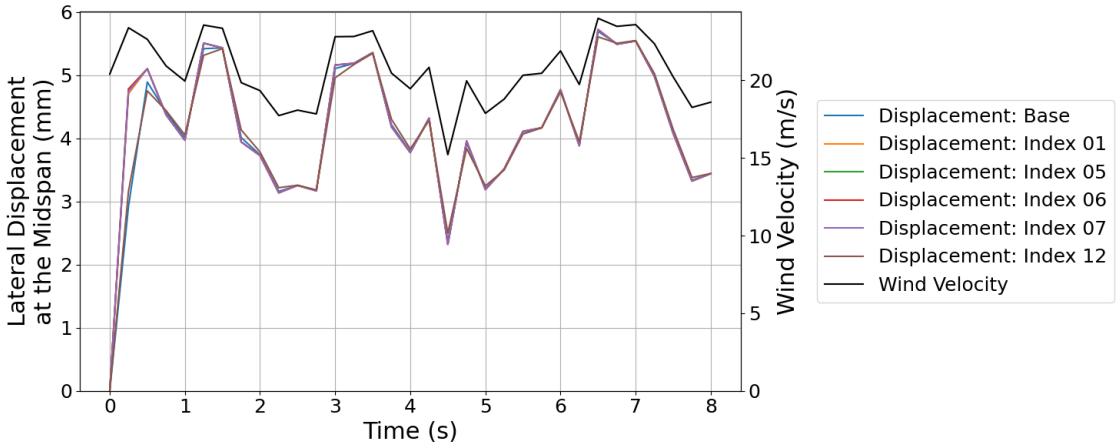
As mentioned in Section 4.4.3, the wind scenarios incur a high computational cost due to the time-varying nature of the loading. Consequently, the number of cases available for analysis is limited. Therefore, this short analysis only aims to demonstrate whether a relation exists between

the location of the applied damage and the displacement of the deck in response to a wind signal, more specifically considering the lateral translation and rotation of the deck at the midspan of the main span and of a side span. The lateral displacement is selected as the best representation since the vertical displacement is negligible due to the high gravity loads that maintain the deck in place in this dimension. It is important to note that all the cases considered for this test series are Doub DSym cases with a damage level of 0.30 to limit the volume of the results. This case selection is justified by the fact that these parameters produced stable models, enabling the computation of results that are sensitive to changes in the step stabilization parameters without requiring modifications to these parameters. Furthermore, as mentioned in the modelling explanations, the aerodynamic coefficients are approximations of the expected values, and the wind signal is artificially generated (Section 3.7). The purpose of this section is solely to highlight the differences between various damage location indices under wind loading; the magnitude of the obtained values should not be interpreted directly.

Starting with the lateral translation of the deck, the results for the displacement time-history at the midspan of the main and side span are presented in Figure 5.35. In the side span, presented in Figure 5.35a, the displacement curve associated with damage at the midspan (Index 12) stands out, exhibiting more pronounced variations in its values. On the other hand, the values for all other locations behave more or less identically throughout the signal, with less extreme values. This indicates that the lateral displacement of the deck in the side spans is more sensitive to the damage applied at the bridge midspan. However, the behaviour is the opposite in the main span figure (5.35b), which illustrates the smallest variations for the Index 12 response. Again, the rest of the locations similarly affect the deck response, making them hard to differentiate. Overall, while there is a vague variation in the results based on the location of the damage, the current data can only distinguish the damage at the midspan of the main span from the locations.



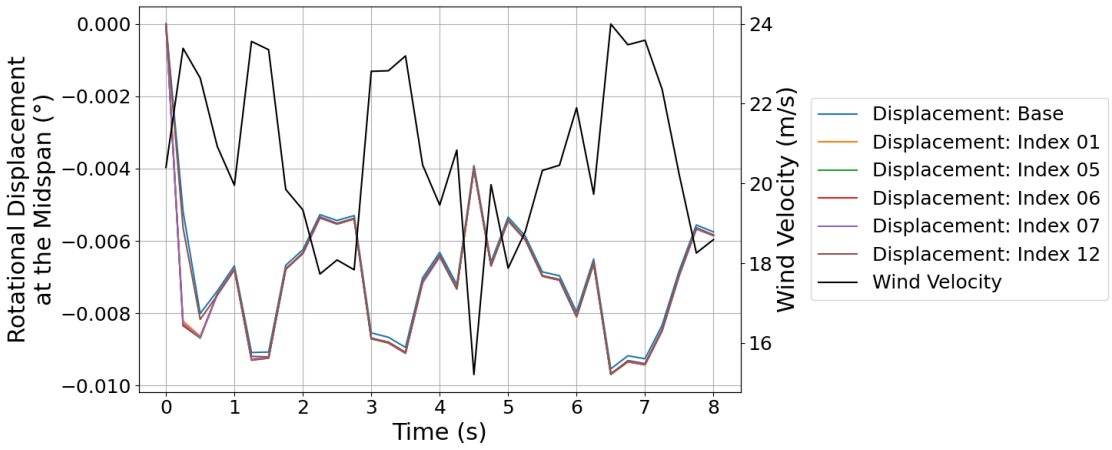
(a) Midspan of Side Span



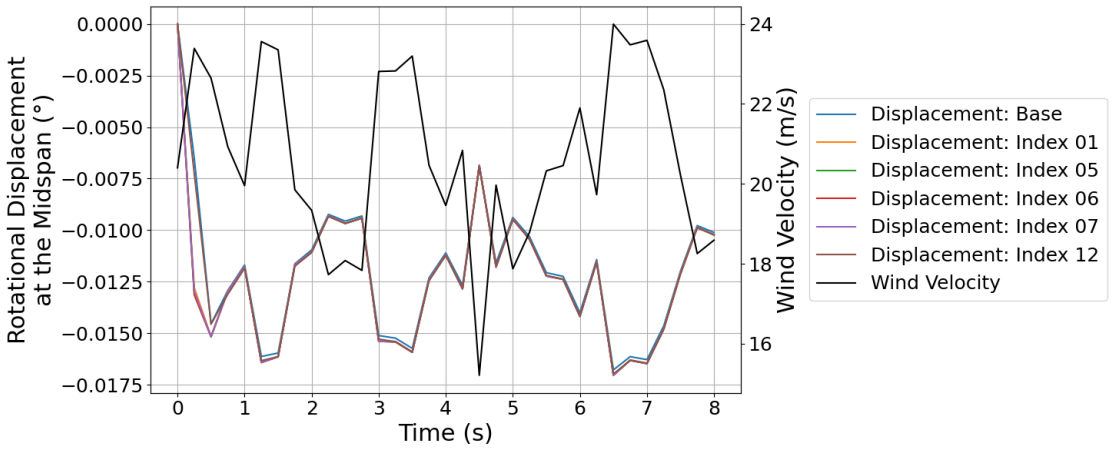
(b) Midspan of Main Span

Figure 5.35: WIND11 Deck Lateral Translation for Doub DSym 0.30 Damage Cases

The results for the deck rotation, available in Figure 5.36, suggest a very different conclusion since all the lines are overlapping. This behaviour is observed in both the main span and the side span, and indicates that the rotational response of the deck only has minimal changes with the variation of the damage location. On the other hand, the line for the baseline model (undamaged) is distinct from the rest, indicating that the application of damage affects the deck's rotation under wind loading. The only exception is for the Index 12 response at the start, which is the only time where it is closer to matching the baseline results than the other damage location index lines.



(a) Midspan of Side Span



(b) Midspan of Main Span

Figure 5.36: WIND11 Deck Rotation for Doub DSym 0.30 Damage Cases

While this section is brief, it generally outlines that the damage location in the main cables impacts the bridge's response when subjected to wind loading. However, it is found in the studied cases that the changes in lateral translation and rotation about the bridge length are minimal. It is highly unlikely that any clear correlation could be established between the damage location index and the lateral or rotational displacement of the deck under wind loading. Moreover, this type of testing is impossible to reproduce in reality, as it would require the same wind signal to occur multiple times over time for a comparable analysis to be performed. Nonetheless, this is also a demonstration of the strength of the created program, which would allow a user to perform a complex dynamic wind analysis on a defined suspension bridge. This can be executed with the inputs presented in Section 4.2.1 without requiring any modification to the program's source code. However, access to a computer with greater processing power would be needed for this type of analysis due to the limitations encountered in this current research.

Chapter 6

Conclusion

This current research consists of two main components: the development of an automated tool to perform the modelling and the automated structural assessment of a conventional suspension bridge and the investigation of the impact of damage applied to the main cables of the modelled bridge, using the developed tool. More specifically, Abaqus's detailed model creation methodology is presented, followed by the modelling process automation through Python. This last item forms the program used throughout this research, which is a flexible tool that can model most suspension bridges through Abaqus. Different automated analyses are performed on the output, such as the natural frequencies extraction, the deck modes of vibration identification through a calibrated algorithm, the bridge key displacements under gravity loading, and the response of the bridge under wind loading. While this program can now be used for various tasks, its use in this research is to model the Lower Liard River Bridge and compute over 2,000 test cases with various damage locations and intensities in the main cables. By defining four test case types (All_, Quad, Sing, and Doub), the changes in the natural frequencies and deck displacement under gravity can be evaluated, and correlations with the applied damage can be deducted. Furthermore, symmetry patterns are defined to generate different damage patterns to remove redundancy while increasing the available damage scenarios, which are critical for further work on this project. Also, a feature was added to the program specifically for the Lower Liard River Bridge to allow users to create rectangular main cable wire patterns, as found in the actual bridge. The aforementioned analyses were performed on the single wire simplified model and the more accurate 3x4 main cable bundle model. An algorithm using the Kaimal spectrum with added noise is created for the wind analysis to generate a high-frequency wind signal of the velocity. This wind signal is applied on the bridge to observe the response under a few critical damage scenarios.

6.1 Main Research Findings and Contributions

One of the main findings obtained based on the results in this research is that the vertical modes and the first torsional modes (symmetric and asymmetric) can be accurate indicators of the applied damage location and intensity. Some significant correlations were observed in the results, which made it possible to state that a machine-learning model could be trained using the developed tool from this research to identify damage locations and intensities in the main cables. These correlations may seem vague for raw examination, but key patterns observed at the anchor, the tower, and the midspan was noticed in many vertical and torsional modes. Another important correlation was found between the analysis of the natural frequencies for individually damaged main cable spans (Sing cases) against cases with a combination of the individual damages within a single case (Doub cases). There is a strong relationship between the natural frequencies of the corresponding cases for vertical modes of vibration and some torsional modes of vibration. This observed relationship indicates that the results from independent adjacent damage locations can be combined with a linear scaling ratio to obtain the results of a case combining both damage. For vertical modes, the obtained linear relation developed indicates that simply adding the changes in the natural frequency change ratio associated to a specific vibration mode for two Sing cases yields a frequency change ratio that is lower that the associated Doub case. To convert the obtained frequency change ratio to the expected Doub case result, the following relationship may be used: $\frac{f_{Comb}}{f_{Base}} = \alpha \left[1 - \left(1 - \frac{f_{Sing_{left}}}{f_{Base}}\right) - \left(1 - \frac{f_{Sing_{right}}}{f_{Base}}\right)\right] + \beta$, which can be simplified to $f_{Comb} = \alpha \left(f_{Sing_{left}} + f_{Sing_{right}}\right) + (\beta - \alpha) f_{Base}$. For the vertical modes, α was found to be 0.9254 for symmetric modes and 0.9145 for asymmetric modes, whereas β was 0.0742 symmetric modes and 0.0850 for asymmetric modes, both displaying a similar behaviour. For the torsional modes, the factors change considerably based on the symmetry: $\alpha = 1.0393$ and $\beta = -0.0392$ for the symmetry modes and $\alpha = 0.9740$ and $\beta = -0.0260$ for the asymmetric modes. This indicates a different behaviour for the symmetric modes, in which the simple addition of the change in the natural frequencies for two Sing cases yields a frequency change ratio that is greater than the associated Doub case. Out of the conclusive modes, the torsional symmetric modes are the only instances in which combining individual cases results proportionally intensifies the shifts in natural frequencies. However, these values are still very close to $\alpha = 1$ and $\beta = 0$, which would be the case where natural frequency shifts are purely additive. This is a strong argument that the behaviour of the bridge can be characterized based on Sing cases to predict extended damage cases. It is expected that this is also scalable to more damage cases, such as combining Doub cases to predict the results of Quad cases.

Extending this theory to further damage patterns would enforce the theory that a machine learning model could detect damage based on shifts in the structure's natural frequencies. The changes in the natural frequencies observed in the defined test cases are relatively low, with a maximum decrease in the frequencies per mode type ranging from 1.5% to 5.7% for basic Sing cases. It remains to be determined if these shifts are sufficient to be noticed in a real suspension bridge through an experimental study with accelerometer sensors installed along the length of the deck. Environmental factors, such as temperature and humidity, could be enough to offset the observed shifts in the natural frequencies, making the analysis considerably more challenging.

However, the objective of this study was to establish a proof of concept for applying damage in the main cables in a theoretical environment and studying the shifts in the natural frequencies of the deck's vibration modes. Not only has there been clear evidence of this correlation, linking damage location and intensity to the shifts, but the process has been automated from end to end. While the development of this program has been a long process, the tool obtained accomplishes everything that would be expected from it within the scope of this research, allowing future research in this field to start from a completed model. This also means that 2,000 individual models were created, computed, and processed for this research alone, which could easily be replicated for any other suspension bridge with minimal additional effort. Apart from natural frequencies and displacement analyses, supplementary features were added to the program to perform basic dynamic wind loading analysis efficiently, extracting the relevant response in the bridge, which comes with a joint feature: the high-frequency wind velocity signal generator.

For the study of the displacement of the deck under gravity loading, it was found that it could be an appropriate tool to distinguish between symmetrical damage locations identified by a machine learning model. As mentioned, the shifts in the natural frequencies hold no indication of the orientation of the bridge relative to the damage location due to the double symmetry of the suspension bridge. Therefore, this research suggests that a complementary analysis of the changes in the displacement of the deck can be used to identify which region of the bridge is damaged. The monitoring of both the shifts in the natural frequencies and the changes in the displacement of the deck will be required for a machine learning model to attempt to determine the exact location of damage components in a suspension bridge.

Regarding the wind loading analysis, while the wind loading application and analysis has been completed, the results do not indicate any conclusive difference between the tested damaged cases and the baseline undamaged model. Based on the obtained results, it is not expected for this branch of the research to be usable in the automatic monitoring of the structural health of a

suspension bridge to identify damage location and intensity in components.

Ultimately, the primary contribution of this thesis is the development of a program that automatically generates numerical models of suspension bridges. By automating both the modelling and the analysis of damage cases, the program greatly facilitates further research on the LLRB and enables the creation of additional bridge models with minimal manual effort. In alignment with the larger conceptual vision (Section 1.5) of developing a generalized tool for real-time, remote damage identification in suspension bridges, this program represents a significant step forward. The program, along with the post-analysis algorithms developed as part of this thesis, enabled the computation of the numerous damage cases analyzed in this study, and they now provide a strong basis for future advancements in this branch of research. Finally, strong conclusions are obtained within this research, clearly demonstrating that a high correlation exists between the applied damage to the main cable (both in location and intensity) and the natural frequencies of the deck's vibration modes, illustrating the potential for future machine learning applications. This is also justified by the fact that the correlations observed indicate that the behaviour of the bridge could be characterized using computed cases with damage localized to a single main cable segment, allowing the prediction of results for more extensive damage scenarios.

6.2 Research Limitations

Experimental data would be required to draw more accurate conclusions and validate the model created. Therefore, the results presented in this document should be considered with reservations to the specific values obtained, as they are highly dependent on the assumptions made. In addition, many model simplifications are employed within the model creation to reduce the computational effort required. While these simplifications allowed the computation of thousands of cases, it does come with a potential loss of accuracy. Then, multiple methods were needed to stabilize some of the steps in many tested cases. Unfortunately, these are the source of some of the discontinuities observed in the results. Combined with the rudimentary mode identification algorithm, which highly depends on its calibration, many anomalies are observed in the natural frequencies identified. While some filtering work has been done on these values, it should still be noted that some anomalies are still within the presented results. In retrospective, these stability issues could be partially mitigated by the use of beams with low bending stiffness to represent the cables, as covered in Section 2.1.2. It is considered that the use of truss is the source of most of the encountered instabilities. The other limiting component in this research is that damage is only considered in the main cables, with a simplified stiffness and mass reduction approach along main cable spans.

Finally, the tensions obtained from the field investigation were not achieved in the model in this research, but it is expected that these values are either imprecise or are associated to a damaged bridge in which the load path is severely affected. In both cases, further field studies should be conducted to enhance the model validation.

6.3 Future Research

Future work following this research could be to develop a machine learning model that could be trained to identify damage location and intensity within the main cables based on the data provided by the automated tool developed. As mentioned in this research, this model should be integrated into the created program and into a database to allow for the test of predicted cases. Furthermore, the theoretical side of this project is missing an experimental counterpart to validate the defined model. This experimental part, which could simply be installing a series of accelerometer sensors along the length of the bridge deck, would make it possible to determine if the environmental factors are too significant compared to the recorded natural frequency shifts due to damage in the main cables. Finally, using the developed automated tool, investigations should be carried out to analyze the effect of damage applied to other components in the bridge and evaluate the shifts of natural frequencies. Without the evaluation of the effect of damage on other components, it would be impossible to validate any findings in a real scenario since the effect of damage in the other bridge components would be unknown. Therefore, the current study provides a strong tool which can be employed for further research in several other topics, such as finite element modelling techniques for suspension bridges, machine learning algorithms for damage identification, and effect evaluation of dynamic loading on structural bridge components.

Bibliography

[1] Qiu, W. L., Jiang, M. & Zhang, Z. Nonlinear dynamic responses of suspension bridge to sudden breakage of hangers. *Journal of Physics: Conference Series* **628**, 012041 (2015). URL https://dx.doi.org/10.1088/1742-6596/628/1/012041. Publisher: IOP Publishing.

[2] Abdel-aziz, A. & Attia, W. INVESTIGATION OF FLUTTER BY TWO NUMERICAL SIMULATIONS ANALYSIS FOR SUSPENSION BRIDGES (2015). URL https://www.semanticscholar.org/paper/INVESTIGATION-OF-FLUTTER-BY-TWO-NUMERICAL-ANALYSIS-Abdel-aziz-Attia/2d28b17104f790a58f7b611e1ebe7cad428910ee.

[3] Fujino, Y. & Siringoringo, D. Vibration Mechanisms and Controls of Long-Span Bridges: A Review. *Structural Engineering International* **23**, 248–268 (2013). URL https://doi.org/10.2749/101686613X13439149156886. Publisher: Taylor & Francis _eprint: https://doi.org/10.2749/101686613X13439149156886.

[4] Seshu, P. *Textbook of Finite Element Analysis* (Prentice-Hall of India Pvt.Ltd, 2012), tenth edn.

[5] Ding, Q. & Lee, P. K. K. Computer simulation of buffeting actions of suspension bridges under turbulent wind. *Computers & Structures* **76**, 787–797 (2000). URL https://www.sciencedirect.com/science/article/pii/S0045794999001972.

[6] Brown, T. L. *et al.* Electrochemistry. In *Chemistry - The Central Science*, p. 1313–1379 (2014). URL https://chem.libretexts.org/Bookshelves/General_Chemistry/Map%3A_Chemistry_-_The_Central_Science_(Brown_et_al.)/20%3A_Electrochemistry.

[7] Kang, Z. *et al.* Effect of pitting corrosion on the mechanical properties and fracture model of steel wires for bridge cable. *Anti-Corrosion Methods and Materials* **70**, 173–181 (2023). URL https://www.emerald.com/insight/content/doi/10.1108/acmm-02-2023-2763/full/html. Publisher: Emerald Publishing Limited.

[8] Jeon, C.-H., Lee, J.-B., Lon, S. & Shim, C.-S. Equivalent material model of corroded prestressing steel strand. *Journal of Materials Research and Technology* **8**, 2450–2460 (2019). URL https://www.sciencedirect.com/science/article/pii/S2238785418309724.

[9] Reimer, S. & Waldie, G. Lower Liard River Bridge Alaska Highway km 763.3 (2011). URL https://buyandsell.gc.ca/cds/public/2016/03/11/f14f7343806fb5aaa324772aeeac473d/sample_of_lower_liard_bridge_report_reduced_size.pdf.

[10] Pétrowski, A. & Ben-Hamida, S. *Evolutionary Algorithms*, vol. 9 (ISTE Ltd, 2017).

[11] AHCS. Lower Liard River Bridge (2012). URL https://ouralaskahighway.com/?portfolio_item=lower-liard-river-bridge.

[12] CSiBridge. CSI Section Designer (2025).

[13] CISC. *Handbook of Steel Construction* (Canadian Institude of Steel Construction, Toronto, Ontario, Canada, 2017), 11th edn.

[14] List of longest cantilever bridge spans (2024). URL https://en.wikipedia.org/w/index.php?title=List_of_longest_cantilever_bridge_spans&oldid=1259188648. Page Version ID: 1259188648.

[15] List of longest continuous truss bridge spans (2024). URL https://en.wikipedia.org/w/index.php?title=List_of_longest_continuous_truss_bridge_spans&oldid=1238364348. Page Version ID: 1238364348.

[16] List of longest arch bridge spans (2025). URL https://en.wikipedia.org/w/index.php?title=List_of_longest_arch_bridge_spans&oldid=1272041304. Page Version ID: 1272041304.

[17] List of longest cable-stayed bridge spans (2025). URL https://en.wikipedia.org/w/index.php?title=List_of_longest_cable-stayed_bridge_spans&oldid=1272268292. Page Version ID: 1272268292.

[18] List of longest suspension bridge spans (2025). URL https://en.wikipedia.org/w/index.php?title=List_of_longest_suspension_bridge_spans&oldid=1271636305. Page Version ID: 1271636305.

[19] Impacts, A. Historic trends in bridge span length (2020). URL https://aiimpacts.org/historic-trends-in-bridge-span-length/. Section: Continuity of progress.

[20] Structurae. List | Cantilever truss bridges. URL https://structurae.net/en/structures/bridges/cantilever-truss-bridges/list.

[21] Bridgemeister. Bridgemeister - Suspension Bridges of Canada (2025). URL https://www.bridgemeister.com/list.php?type=country&country=Canada.

[22] MIDAS. Solution | Suspension Bridges. URL https://resource.midasuser.com/en/solution/suspension-bridges.

[23] Wang, D., Ye, J., Wang, B. & Wahab, M. A. Review on the Service Safety Assessment of Main Cable of Long Span Multi-Tower Suspension Bridge. *Applied Sciences* **11**, 5920 (2021). URL https://www.mdpi.com/2076-3417/11/13/5920. Number: 13 Publisher: Multidisciplinary Digital Publishing Institute.

[24] Harish, A. Why the Tacoma Narrows Bridge Collapsed (2018). URL https://www.simscale.com/blog/tacoma-narrows-bridge-collapse/.

[25] Jun, H. Probabilistic Fatigue Damage Analysis of Long Span Suspension Bridge Due to Wind Induced Buffeting. *Materials Science Forum* **980**, 275–281 (2020). URL https://www-scientific-net.proxy.bib.uottawa.ca/MSF.980.275. Publisher: Trans Tech Publications Ltd.

[26] Cai, C. S. & Montens, S. Chapter 57: Wind Effects on Long-Span Bridges. In *Bridge Engineering Handbook* (CRC Press, 2000). URL https://ingenieriasismica.utpl.edu.ec/sites/default/files/publicaciones/UCG-ES-00580.pdf.

[27] Zhao, C. *et al.* Considering frictional slippage at saddle-cable interface in seismic behavior of a suspension bridge. *Advances in Structural Engineering* **24**, 995–1008 (2021). URL https://doi.org/10.1177/1369433220974778. Publisher: SAGE Publications Ltd STM.

[28] Simulia, D. S. Beams and Trusses - 2020 - SOLIDWORKS Help (2020). URL https://help.solidworks.com/2020/english/SolidWorks/cworks/c_Beams_Trusses.htm?id=d7300bafccdb4d39acaea7af0ff2cfae#Pg0.

[29] Ltd., E. 3D Frame Analysis Library Technical Notes & Examples (2018). URL https://www.engissol.com/Downloads/Technical%20Notes%20and%20examples.pdf.

[30] LaValle, S. M. Yaw, pitch, and roll rotations (2012). URL https://msl.cs.uiuc.edu/planning/node102.html.

[31] Haber, A. Direction Cosine Matrix Tutorial – Correct Mathematical Explanation - Important for Aerospace and Robotics Engineering – Fusion of Engineering, Control, Coding, Machine Learning, and Science (2025). URL https://aleksandarhaber.com/direction-cosine-matrix-tutorial-important-for-aerospace-and-robotics-engineering/.

[32] Hearn, E. J. CHAPTER 1 - SIMPLE STRESS AND STRAIN. In Hearn, E. J. (ed.) *Mechanics of Materials 1 (Third Edition)*, 1–26 (Butterworth-Heinemann, Oxford, 1997). URL https://www.sciencedirect.com/science/article/pii/B9780750632652500025.

[33] FEMA. Structural Dynamics of Linear Elastic Single-Degree-of-Freedom (SDOF) Systems. URL http://www.ce.memphis.edu/7119/PDFs/FEAM_Notes/Topic03-StructuralDynamicsofSDOFSystems.pdf.

[34] Moore, J., Chatsaz, M., d'Entremont, A., Kowalski, J. & Miller, D. Viscous Damped Free Vibrations. URL https://mechanicsmap.psu.edu/websites/16_one_dof_vibrations/16-2_viscous_damped_free/16-2_viscous_damped_free.html.

[35] Raymond, D. J. 12.4: Complex Exponential Solutions (2021). URL https://phys.libretexts.org/Bookshelves/University_Physics/Radically_Modern_Introductory_Physics_Text_I_(Raymond)/12%3A_Harmonic_Oscillator/12.04%3A_Complex_Exponential_Solutions.

[36] OpenStax. 15.6: Damped Oscillations (2016). URL https://phys.libretexts.org/Bookshelves/University_Physics/University_Physics_(OpenStax)/Book%3A_University_Physics_I_-_Mechanics_Sound_Oscillations_and_Waves_(OpenStax)/15%3A_Oscillations/15.06%3A_Damped_Oscillations.

[37] Oasys. Participation Factor And Effective Mass | Oasys GSA Documentation (2024). URL https://docs.oasys-software.com/structural/gsa/references-theory/participation-factor-and-effective-mass/.

[38] Richardson, M. H. Modal Mass, Stiffness and Damping (2000). URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c9b52d7642c1fe455dd0d1dde94fc328f245ca8a.

[39] Gazzola, F. A Fish-Bone Beam Model. In Gazzola, F. (ed.) *Mathematical Models for Suspension Bridges: Nonlinear Structural Instability*, 105–147 (Springer International Publishing, Cham, 2015). URL https://doi.org/10.1007/978-3-319-15434-3_3.

[40] Wang, D. *et al.* Contact and slip behaviors of main cable of the long-span suspension bridge. *Engineering Failure Analysis* **136**, 106232 (2022). URL https://www.sciencedirect.com/science/article/pii/S1350630722002060.

[41] Zhang, W.-m., Lu, X.-f., Wang, Z.-w. & Liu, Z. Effect of the main cable bending stiffness on flexural and torsional vibrations of suspension bridges: Analytical approach. *Engineering Structures* **240**, 112393 (2021). URL https://www.sciencedirect.com/science/article/pii/S0141029621005435.

[42] Cheng, J., Xu, M. & Xu, H. Mechanical performance analysis and parametric study of double-deck plate-truss composite steel girders of a three-tower four-span suspension bridge. *Engineering Structures* **199**, 109648 (2019). URL https://www.sciencedirect.com/science/article/pii/S0141029619310879.

[43] Opsahl Steigen, R. *Modeling and analyzing a suspension bridge in light of deterioration of the main cable wires.* Master's thesis, University of Stavanger, Norway (2011). URL https://uis.brage.unit.no/uis-xmlui/handle/11250/182758. Accepted: 2011-10-24T15:29:06Z.

[44] Revie, R. W. & Uhlig, H. H. *Corrosion and Corrosion Control - An Introduction to Corrosion Science and Engineering* (John Wiley & Sons, Inc, Hoboken New Jersey, 2008), fourth edn.

[45] Persson, D., Thierry, D. & Karlsson, O. Corrosion and corrosion products of hot dipped galvanized steel during long term atmospheric exposure at different sites world-wide. *Corrosion Science* **126**, 152–165 (2017). URL https://www.sciencedirect.com/science/article/pii/S0010938X1631469X.

[46] Li, R., Miao, C. & Wei, T. Effect of environmental factors on electrochemical corrosion of galvanized steel wires for bridge cables. *Anti-Corrosion Methods and Materials* **69**, 111–118 (2022). URL https://www.emerald.com/insight/content/doi/10.1108/ACMM-07-2021-2522/full/html.

[47] Zhang, Z. *et al.* Effect of Coating on Stress Corrosion Performance of Bridge Cable Steel Wire. *Coatings (Basel)* **13**, NA–NA (2023). URL https://go-gale-com.proxy.bib.

uottawa.ca/ps/i.do?p=AONE&sw=w&issn=20796412&v=2.1&it=r&id=GALE%7CA762476583&sid=googleScholar&linkaccess=abs. Publisher: MDPI AG.

[48] Nakamura, S.-I., Suzumura, K. & Tarui, T. Mechanical Properties and Remaining Strength of Corroded Bridge Wires. *Structural Engineering International* **14**, 50–54 (2004). URL https://doi.org/10.2749/101686604777964305. Publisher: Taylor & Francis _eprint: https://doi.org/10.2749/101686604777964305.

[49] Hopwood, T. & Havens, J. Corrosion of Cable Suspension Bridges. *Kentucky Transportation Center Research Report* (1984). URL https://uknowledge.uky.edu/ktc_researchreports/660.

[50] Solutions, F. Structural Health Monitoring for Bridge Structures (2019). URL https://fprimec.com/structural-health-monitoring-for-bridge-structures/. Section: Structural Health Monitoring.

[51] Farrar, C. R. & Worden, K. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **365**, 303–315 (2006). URL https://royalsocietypublishing.org/doi/10.1098/rsta.2006.1928. Publisher: Royal Society.

[52] Rytter, A. Vibrational Based Inspection of Civil Engineering Structures (1993). Place: Aalborg Publisher: Dept. of Building Technology and Structural Engineering, Aalborg University.

[53] Wickramasinghe, W. R. *Damage detection in suspension bridges using vibration characteristics*. phd, Queensland University of Technology (2015). URL https://eprints.qut.edu.au/87435/.

[54] Worden, K. & Dulieu-Barton, J. M. An Overview of Intelligent Fault Detection in Systems and Structures. *Structural Health Monitoring* **3**, 85–98 (2004). URL https://doi.org/10.1177/1475921704041866. Publisher: SAGE Publications.

[55] Feroz, S. & Abu Dabous, S. UAV-Based Remote Sensing Applications for Bridge Condition Assessment. *Remote Sensing* **13**, 1809 (2021). URL https://www.mdpi.com/2072-4292/13/9/1809. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[56] He, D. Surface corrosion evaluation of steel cable for suspension bridge using electromagnetic method. *Surface Science and Technology* **1**, 16 (2023). URL https://doi.org/10.1007/s44251-023-00018-1.

[57] Paulson, P. O. Continuous acoustic monitoring of suspension bridges and cable stays. In *Structural Materials Technology III: An NDT Conference*, vol. 3400, 205–213 (SPIE, 1998). URL https://www.spiedigitallibrary.org/conference-proceedings-of-spie/3400/0000/Continuous-acoustic-monitoring-of-suspension-bridges-and-cable-stays/10.1117/12.300092.full.

[58] Wei Fan & Pizhong Qiao. Vibration-based Damage Identification Methods: A Review and Comparative Study. *Structural Health Monitoring* **10**, 83–111 (2011). URL https://journals.scholarsportal.info/details/14759217/v10i0001/83_vdimaracs.xml. Publisher: SAGE Publications.

[59] Nguyen, C.-K. *et al.* Damage Detection in Beam-Like Structures Using Mode Shape Curvature Change-Based Method. In Ha-Minh, C., Pham, C. H., Vu, H. T. H. & Huynh, D. V. K. (eds.) *Proceedings of the 7th International Conference on Geotechnics, Civil Engineering and Structures, CIGOS 2024, 4-5 April, Ho Chi Minh City, Vietnam*, 279–286 (Springer Nature, Singapore, 2024).

[60] Kim, J. T. & Stubbs, N. CRACK DETECTION IN BEAM-TYPE STRUCTURES USING FREQUENCY DATA. *Journal of Sound and Vibration* **259**, 145–160 (2003). URL https://www.sciencedirect.com/science/article/pii/S0022460X02951321.

[61] Zhang, Y., Wang, L., Lie, S. T. & Xiang, Z. Damage detection in plates structures based on frequency shift surface curvature. *Journal of Sound and Vibration* **332**, 6665–6684 (2013). URL https://www.sciencedirect.com/science/article/pii/S0022460X13006743.

[62] Salawu, O. S. Detection of structural damage through changes in frequency: a review. *Engineering Structures* **19**, 718–723 (1997). URL https://www.sciencedirect.com/science/article/pii/S0141029696001496.

[63] Ju, F. D. & Mimovich, M. E. Experimental Diagnosis of Fracture Damage in Structures by the Modal Frequency Method. *Journal of Vibration, Acoustics, Stress, and Reliability in Design* **110**, 456–463 (1988). URL https://doi.org/10.1115/1.3269550.

[64] Cawley, P. & Adams, R. D. The location of defects in structures from measurements of natural frequencies. *The Journal of Strain Analysis for Engineering Design* **14**, 49–57 (1979). URL https://journals.sagepub.com/doi/10.1243/03093247V142049.

[65] Askegaard, V. & Mossing, P. LONG TERM OBSERVATION OF RC-BRIDGE USING CHANGES IN NATURAL FREQUENCY. NORDIC CONCRETE RESEARCH. PUBLICATION NO 7. *Publication of: NORDIC CONCRETE FEDERATION* (1988). URL https://trid.trb.org/View/307574.

[66] Wickramasinghe, W. R., Thambiratnam, D. P., Chan, T. H. T. & Nguyen, T. Vibration characteristics and damage detection in a suspension bridge. *Journal of Sound and Vibration* **375**, 254–274 (2016). URL https://www.sciencedirect.com/science/article/pii/S0022460X16300785.

[67] Domaneschi, M., Limongelli, M. & Martinelli, L. Structural Damage Localization in a Suspension Bridge Under Seismic Excitation (2012). URL https://www.semanticscholar.org/paper/Structural-Damage-Localization-in-a-Suspension-Domaneschi-Limongelli/f4df40cdf7b5319d82aa3f7073a167eb837df4fa.

[68] Xiang, Y., Noël, É., Dragomirescu, E. & Martín-Pérez, B. The effect of cable corrosion on dynamic response of Stonecutters Cable-Stayed Bridge model. *Canadian Journal of Civil Engineering* **50**, 282–293 (2022). URL https://www.sciencedirect.com/org/science/article/pii/S0315146822000748.

[69] University, P. Bernoulli's Equation. URL https://www.princeton.edu/~asmits/Bicycle_web/Bernoulli.html.

[70] Nakayama, Y. Chapter 9 - Drag and Lift. In Nakayama, Y. (ed.) *Introduction to Fluid Mechanics (Second Edition)*, 177–201 (Butterworth-Heinemann, 2018). URL https://www.sciencedirect.com/science/article/pii/B9780081024379000097.

[71] Chandrasekaran, S. & Srivastava, G. Introduction. In Chandrasekaran, S. & Srivastava, G. (eds.) *Design Aids of Offshore Structures Under Special Environmental Loads including Fire Resistance*, 1–72 (Springer, Singapore, 2018). URL https://doi.org/10.1007/978-981-10-7608-4_1.

[72] Kaimal, J. C., Wyngaard, J. C., Izumi, Y. & Coté, O. R. Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society* **98**, 563–589 (1972). URL https://onlinelibrary.wiley.com/doi/abs/10.1002/qj.49709841707. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qj.49709841707.

[73] Patro, S. R., Banerjee, A., Adhikari, S. & Ramana, G. V. Kaimal spectrum based H2 optimization of tuned mass dampers for wind turbines. *Journal of Vibration and Control* **29**, 3175–3185 (2023). URL https://journals.scholarsportal.info/details/10775463/v29i13-14/3175_ksbhootmdfwt.xml. Publisher: SAGE Publications.

[74] Acquisition d'un signal analogique : bande passante, théorème d'échantillonnage de Nyquist et repliement (2024). URL https://www.ni.com/fr/shop/data-acquisition/measurement-fundamentals/analog-fundamentals/acquiring-an-analog-signal--bandwidth--nyquist-sampling-theorem-.html.

[75] Yan, Y. J., Cheng, L., Wu, Z. Y. & Yam, L. H. Development in vibration-based structural damage detection technique. *Mechanical Systems and Signal Processing* **21**, 2198–2211 (2007). URL https://www.sciencedirect.com/science/article/pii/S0888327006002226.

[76] Greco, A., Cannizzaro, F. & Pluchino, A. Seismic collapse prediction of frame structures by means of genetic algorithms. *Engineering Structures* **143**, 152–168 (2017). URL https://www.sciencedirect.com/science/article/pii/S0141029616305715.

[77] Greco, A., Cannizzaro, F., Bruno, R. & Pluchino, A. A nested genetic algorithm strategy for an optimal seismic design of frames. *Computational Optimization and Applications* **87**, 677–704 (2024). URL https://doi.org/10.1007/s10589-023-00523-x.

[78] SkyCiv. CSA G40.21 Steel Properties (2024). URL https://beamdimensions.com/materials/Steel/Canadian_Standards_Association_(CSA)/CSA_G40.21/.

[79] Google. Liard River Suspension Bridge (2022). URL https://www.google.com/maps/place/Liard+River+Suspension+Bridge/@59.4135831,-126.0977851,17z/data=!3m1!4b1!4m6!3m5!1s0x53e55d852888ef15:0x73e50bdbccbeb10c!8m2!3d59.4135804!4d-126.0952102!16s%2Fg%2F11f1bkng28?entry=ttu&g_ep=EgoyMDI1MDIwMi4wIKXMDSoASAFQAw%3D%3D.

[80] Hearn, E. J. CHAPTER 5 - SLOPE AND DEFLECTION OF BEAMS. In Hearn, E. J. (ed.) *Mechanics of Materials 1 (Third Edition)*, 92–139 (Butterworth-Heinemann, Oxford, 1997). URL https://www.sciencedirect.com/science/article/pii/B9780750632652500062.

[81] Hearn, E. J. CHAPTER 8 - TORSION. In Hearn, E. J. (ed.) *Mechanics of Materials 1 (Third Edition)*, 176–197 (Butterworth-Heinemann, Oxford, 1997). URL https://www.sciencedirect.com/science/article/pii/B9780750632652500098.

[82] Manufacturers, L. C. Item # ASTM A586-2, Galvanized Structural Strand ASTM A586. URL https://catalog.lexcocable.com/item/all-categories-strand-bridge-rope/dge-rope-galvanized-structural-strand-br-astm-a586/astm-a586-2.

[83] Alkharisi, M. K. & Heyliger, P. R. Free Vibration of Inclined Twisted Cables. *Structures* **50**, 1879–1896 (2023). URL https://www.sciencedirect.com/science/article/pii/S2352012423003016.

[84] Simulia, D. S. ABAQUS Analysis User's Manual (v6.6) (2009). URL https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/usb/default.htm.

[85] ABAQUS. ABAQUS Release Notes (v6.5-1) - Surface-to-surface tie constraint formulation. URL https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.5/books/rnb/default.htm?startat=abc10aqs01.html?

[86] Liu, X. Wind loads on multiple cylinders arranged in tandem with effects of turbulence and surface roughness. *LSU Master's Theses* (2003). URL https://repository.lsu.edu/gradschool_theses/2653.

Appendix A

Temperature Calculation Functions

Algorithm A.1 Calculate Temperature Difference Functions

```
def calc_temperature_hanger(pretension, bridge):
  area = bridge.cables.cable_area.hangers
  hanger_material = bridge.model.materials[bridge.components_material.hangers]
  young_mod = hanger_material.elastic.table[0][0]
  alpha = hanger_material.expansion.table[0][0]
  temperature = pretension / (young_mod * area * alpha)
  return temperature


def calc_temperature_mc(pretension, bridge):
  n_strands = bridge.cables.cable_bundle.n_width * bridge.cables.cable_bundle.n_height
  pretension = pretension / n_strands
  area = bridge.cables.cable_area.main_cables
  mc_material = bridge.model.materials[bridge.components_material.hangers]
  young_mod = mc_material.elastic.table[0][0]
  alpha = mc_material.expansion.table[0][0]
  temperature = pretension / (young_mod * area * alpha)
  return temperature
```

Appendix B

Input Specification for the Bridge Modelling Program

B.1 Steps JSON

The program requires inputs obtained in three stages. The first part comprises the Steps, provided as a JSON file. This Steps file contains a JSON object literal, composed of key and value pairs. The keys are strings that represent step names for the model in Abaqus, whereas the values are JSON objects that contain the parameters that define the step. The content of the JSON objects highly depends on the step type, which is a common key. The program can handle two types of steps for the bridge model creation: Static General and Frequency, as described in Section 3.4. The available keys are presented in Table B.1 for Static General steps, and those for Frequency steps are displayed in Table B.2.

Table B.1: Parameters in Steps JSON objects - Static, General

Key	Value Type	Description {suggested units}	Available Values (Blank for Any of Valid Type)
type	string	The type of Abaqus step	"StaticGeneral"
pre_step_name	string	The name of the step that precedes this step	
description	string	The description of the step	
nlgeom	boolean	The non-linear geometry settings, true to toggle on	
stab_method	string	The automatic stabilization method for the step	"None" for None; "Damping_Factor" for Specify damping factor; "Dissipated_Energy_Fraction" for Specify Dissipated Energy Fraction; "Damping_Factor_From_Prev" for Use damping factors from previous general step
stab_magn	number, null	The magnitude of the stabilization as a dissipated energy fraction or a damping factor {no units}	
adap_damp_ratio	number, null	The maximum ratio of stabilization to strain energy for the use of adaptive stabilization {no units}	
time_inc_method	string	The type of time incrementation	"Automatic"; "Fixed"
max_inc	number, null	The maximum increment size {s}	Any number greater than or equal to initial_inc
min_inc	number, null	The minimum increment size {s}	Any number greater than 0 and lower than or equal to initial_inc
initial_inc	number	The initial increment size {s}	Any number greater than 0
inc_number	number (integer)	The maximum number of increments	Any positive integer greater than 0
adiabatic	boolean	The adiabatic heating effects, true to toggle on.	

Table B.2: Parameters in Steps JSON objects - Frequency

Key	Value Type	Description	Available Values (Blank for Any of Valid Type)
type	string	The type of Abaqus step	"Frequency"
pre_step_name	string	The name of the step that precedes this step	
description	string	The description of the step	
num_eigen	number (integer)	The number of eigenvalues requested with the Subspace Eigensolver	Any positive integer greater than 0

For the specific LLRB model discussed in this thesis, the baseline Steps JSON object literal used is presented in Algorithm B.1.

Algorithm B.1 Steps JSON File

```
{
  "Pre-Displacement": {
    "stab_method": "Dissipated_Energy_Fraction",
    "description": "Pre-displacement of the cables",
    "pre_step_name": "Initial",
    "nlgeom": true,
    "stab_magn": 0.0002,
    "adap_damp_ratio": 0.05,
    "max_inc": 1,
    "min_inc": 1e-06,
    "initial_inc": 0.1,
    "type": "StaticGeneral",
    "time_inc_method": "Automatic",
    "inc_number": 100,
    "adiabatic": false
  },
  "Vertical Loading": {
    "stab_method": "Dissipated_Energy_Fraction",
    "description": "Gravity application",
    "pre_step_name": "Pre-Displacement",
    "nlgeom": true,
    "stab_magn": 0.0002,
    "adap_damp_ratio": 0.05,
    "max_inc": 1,
    "min_inc": 1e-6,
    "initial_inc": 0.1,
    "type": "StaticGeneral",
    "time_inc_method": "Automatic",
    "inc_number": 1000,
    "adiabatic": false
  },
  "Wind Loading": {
    "stab_method": "Dissipated_Energy_Fraction",
    "description": "Wind loading application",
    "pre_step_name": "Vertical Loading",
    "nlgeom": true,
    "stab_magn": 0.0002,
    "adap_damp_ratio": 0.05,
    "max_inc": 0.25,
    "min_inc": 1e-06,
    "initial_inc": 0.25,
    "type": "StaticGeneral",
    "time_inc_method": "Automatic",
    "inc_number": 10000,
    "adiabatic": false
  },
  "Frequency": {
    "description": "Frequency Analysis",
    "pre_step_name": "Vertical Loading",
    "num_eigen": 50,
    "type": "Frequency"
  }
}
```

B.2 Materials JSON

The second component of the inputs is the properties of the materials. A similar structure as the Steps inputs is used, with a JSON file containing a JSON object literal for which every key and value pair is a different material. A key represents the name of the material, whereas its value is a JSON object containing the material's properties. Similarly to the Steps, Table B.3 provides the available keys for the materials.

Table B.3: Parameters in Materials JSON objects

Key	Value Type	Description {suggested units}
type	string	The type of material in Abaqus - Only "Elastic" available
mass_density	number	The density of the material {kg/m^3}
young_mod	number	The Young's modulus (modulus of elasticity) of the material {Pa}
poisson_ratio	number	The Poisson's ratio of the material {no units}
no_compression (optional)	boolean, null	The no compression setting of the material, true to toggle on, defaults to false
expansion_coeff (optional)	number, null	The mechanical expansion coefficient of the material {m/mK}
specific_heat (optional)	number, null	The specific heat of the material {J/kg°C}
conductivity (optional)	number, null	The thermal conductivity of the material {W/mK}

The JSON object literal for the Material inputs is presented in Algorithm B.2. With the Steps and Materials inputs as a foundation, the last component is a collection of the remaining inputs, primarly geometric, required to create the model. These inputs are provided as an Excel file following a template that can be processed by the Python program afterwards. The file contains 14 sheets, each covering a specific scope of the bridge definition. The following sub-sections cover each sheet and the inputs described in Chapter 3.

Algorithm B.2 Materials JSON File

```
{
  "Cable Steel": {
    "type": "Elastic",
    "mass_density": 7850,
    "young_mod": 200e9,
    "poisson_ratio": 0.3,
    "expansion_coeff": 11.7e-6,
    "no_compression": true
  },
  "Structural Steel": {
    "type": "Elastic",
    "mass_density": 7850,
    "young_mod": 200e9,
    "poisson_ratio": 0.3,
    "no_compression": false
  },
  "Support Material": {
    "type": "Elastic",
    "mass_density": 0,
    "young_mod": 1,
    "poisson_ratio": 0.3
  },
  "Rigid Steel": {
    "type": "Elastic",
    "mass_density": 0,
    "young_mod": 200e9,
    "poisson_ratio": 0
  }
}
```

B.3 General Parameters Sheet

The first sheet provides the general bridge information, as presented in Table B.4. For testing purposes, features are available to model only specific regions or components of the bridge. The different components and regions can be activated (TRUE) or deactivated (FALSE) in the model to facilitate the design process or to create targeted analyses. An equivalent feature is available for modelling both sides (east and west) of the bridge, which can be set to FALSE to model only a single side. While this thesis does not provide a detailed explanation, the remainder of the modelling process compensates for missing components or regions by applying logical constraints and boundary conditions. These ensure that all generated models remain computationally viable. The name of the material of each component is specified, which must match the materials provided in the Materials inputs file. Furthermore, gravity can be activated (TRUE), its magnitude can be selected (m/s^2), and the step in which it is first applied. For the deactivation of specific components in the model, if the deck is deactivated, the option to manually load the hangers at their bottom

to simulate the deck mass is available. The step in which this loading is created must be specified if used. Wind can be activated from this sheet, along with its activation step, as an additional loading feature. The loading parameters for wind can be provided in the Wind Sheet if activated. Thereafter, the stability fix setting controls whether the cable supports are added to the model and if the pre-displacement is considered to avoid aborted computations for the model (Section 3.4). The material for the cable support is provided in the same table, followed by the step in which the stability fix is created and the pre-displacement magnitude (m). The following table pertains to the activation of pretension in the main cables and hangers. If activated, the corresponding values are provided in the Pretension Sheet. The last component of the General Sheet is the toggle for Natural Frequencies, which controls whether a Frequency step is created and if the frequency analysis is performed.

Table B.4: General Parameters Input Sheet

Components to Model	
Main Cables	TRUE
Hangers	TRUE
Towers	TRUE
Deck	TRUE

Regions to Model	
Start to Tower1	TRUE
Tower1 to Tower2	TRUE
Tower2 to End	TRUE

Model Both Sides	TRUE

Components	Material
Main Cables	Cable Steel
Hangers	Cable Steel
Towers	Structural Steel
Decks	Structural Steel
Girders	Rigid Steel
Hangers-Beam	Structural Steel
Braces	Structural Steel
Cross Girders	Structural Steel
Anchor Bearings	Structural Steel
Tower Bearings	Structural Steel
Cable Connections	Rigid Steel

Gravity	
Exist	TRUE
Value	9.81
Step	Vertical Loading

Manual Hanger Loading	
Exist	FALSE
Step	Vertical Loading

Wind Loading	
Exist	FALSE
Step	Wind Loading

Stability Fix	
Applied	TRUE
Material	Support Material
Step	Pre-Displacement
Pre-Displacement	-0.01

Pre-Tension	
Main Cables	TRUE
Hangers	TRUE

Natural Frequencies	TRUE

B.4 Deck Properties Sheet

In this sheet, the properties of the deck are provided as per Table B.5, which includes the profiles for the deck, the girder and the bearing elements, the deck width, the centroid and shear centre dimensions, the anchor bearing height, the anchor connections at the ends of the deck as well as the mesh size properties. This sheet is the first to use the profile inputs template, which is composed of seven cells covering any generalized profile in Abaqus, with the area (m^2), the second moments of area (m^4), the torsional constant (m^4), the sectoral moment (Gamma O - m^4), and the warping constant (Gamma W - m^6). While this model does not currently involve using the sectoral moment or the warping constant, these parameters are still available as a feature for future project adaptations. The specified areas, second moments of area and the torsional constants provided reflect the values from Section 3.2.2. The deck width (m) is the length along the Z axis between hanger pairs, and the deck centroid below hanger (m) is the height of the vertical arms of girders. The deck shear centre above the centroid (m) is simply the height difference between the centroid (deck level in the model) and the shear centre of the deck. The last dimension provided in this

sheet is the anchor bearing height (m).

The anchor connections for the anchor bearing's bottom (boundary condition) and top (constraint) can be modified or removed by specifying the fixed degrees of freedom with TRUE. Finally, the deck's meshing precision is controlled by the number of mesh elements per wire for every component.

Table B.5: Deck Properties Input Sheet

	Deck Profile
Area	0.689
Ix	0.168
Ixy	0
Iy	1.66
J	0.22
Gamma O	0
Gamma W	0

	Girder Profile
Area	1E+00
Ix	1E+02
Ixy	0
Iy	1E+02
J	1E+01
Gamma O	0
Gamma W	0

Deck Width
9.144

Deck Centroid Below Hanger (h)
1.6567

*origin to hanger - centre of mass

Deck Shear Centre Above Centroid
0.115

*centroid - shear centre

Anchor Bearing Height
2

	Anchor Connection	
	Bottom	Top
Exist	TRUE	TRUE
UX	TRUE	TRUE
UY	TRUE	TRUE
UZ	TRUE	TRUE
RX	TRUE	TRUE
RY	TRUE	TRUE
RZ	FALSE	FALSE

	Mesh Element
Deck	1
Girder	1
Anchor Bearing	1
Tower Bearing	1

	Anchor Bearing Profile
Area	0.0167
Ix	2.21E-04
Ixy	0
Iy	7.45E-05
J	3.12E-06
Gamma O	0
Gamma W	0

	Tower Bearing Profile
Area	0.0334
Ix	0.002187416
Ixy	0
Iy	0.000149
J	0.00000624
Gamma O	0
Gamma W	0

B.5 Tower Properties Sheet

The following sheet covers the tower properties and is presented in Tables B.6 and B.7. In the first part of the sheet, the generalized profile properties for the vertical tower parts and the tower cross girder are available. Then, each tower component's mesh values by size (m) are provided. This is followed by the connections between the tower cross girder and the girder at the tower (as part of the deck). This connection is made through two external bearings and the central connection (Section 3.5). The restrained degrees of freedom are identified, and an option is available to add springs (of the elasticity constant specified - N/m) at each tower bearing is available, which can be help model certain deck-to-tower connections in other bridges. Furthermore, any connections may be disabled to accommodate different designs.

Table B.6: Tower Properties Input Sheet - Main Profiles, Mesh and Constraints

	Tower Profile
Area	9.694E-02
Ix	6.027E-03
Ixy	0
Iy	2.830E-03
J	6.825E-05
Gamma O	0
Gamma W	0

	Tower Mesh Size
Tower	0.75
Cross Girder	0.75
Brace	0.75

	Girder Connection		
	Exterior Bot	Exterior Top	Centre
Exist	TRUE	TRUE	TRUE
UX	TRUE	TRUE	FALSE
UY	TRUE	TRUE	FALSE
UZ	TRUE	TRUE	TRUE
RX	TRUE	TRUE	FALSE
RY	TRUE	TRUE	FALSE
RZ	FALSE	FALSE	FALSE

	Girder Profile
Area	8.830E-02
Ix	1.360E-02
Ixy	0
Iy	9.768E-03
J	1.240E-02
Gamma O	0
Gamma W	0

Spring UY	0.00E+00

The next part of the tower sheet concerns the height of the vertical tower parts and the brace system. First, points of interest (POI) along the tower's height are defined as ID (integer) and Y coordinate pairs. These POI represent the bottom of the tower (defaults to the POI with the lowest coordinate), the top of the tower (defaults to the POI with the highest coordinate), and connections to the cross girder and the braces. The ID of the POI that connects to the cross girder is then specified, along with the distance along the Z axis between vertical tower part pairs at the bottom and the top of the tower. The values in Table B.7 for the dimensions are equivalent to those in Figure 3.5. Afterwards, the profiles for the different braces are supplied with the generalized profile input template. The available brace types are those used in the model (chevron, cross, and horizontal — Section 3.2.1) and inverted chevron, which could be used for another design. The connection points are defined as pairs of POI ID for the absolute bottom and top of the brace for chevron, inverted chevron, and cross braces. Logically, the top of the chevron and the bottom of the inverted chevron braces should match the ID of a horizontal brace or the cross girder, which the program will recognize as a connection point. The horizontal braces are specified as a single POI ID at the connection point with the vertical towers. This system of inputs for the towers allows for a wide range of tower designs. A critical point to mention regarding the Y coordinates of the towers is that they are shifted based on the change in the deck elevation at the towers, provided in Section B.7. As such, the values entered in the spreadsheet would be used if the elevation at the deck was the same at the anchor than at the towers. For the LLRB, the deck elevation difference between the anchors and the towers is 1.744 m, which means that all the Y coordinates of the towers are shifted in the Y-positive direction by 1.744 m. This ensures consistency in the connection between the deck and the towers.

Table B.7: Tower Properties Input Sheet - Coordinates and Braces

POI ID	y
1	-10.521
2	-9.544
3	-3.748
4	3.191
5	17.332

	POI	Width
Bottom	1	11.1125
Girder	3	
Top	5	9.144

	Brace Profile
Area	0.02110
Ix	1.110E-03
Ixy	0
Iy	1.140E-04
J	2.830E-06
Gamma O	0
Gamma W	0

Chevron	
2	3

	Brace Profile
Area	0.02110
Ix	1.110E-03
Ixy	0
Iy	1.140E-04
J	2.830E-06
Gamma O	0
Gamma W	0

Inverted Chevron	

	Brace Profile
Area	0.02110
Ix	1.110E-03
Ixy	0
Iy	1.140E-04
J	2.830E-06
Gamma O	0
Gamma W	0

Cross	
4	5

	Brace Profile
Area	0.03670
Ix	1.041E-03
Ixy	0
Iy	1.679E-03
J	1.921E-03
Gamma O	0
Gamma W	0

Horizontal
2
5

B.6 Cable Properties Sheet

The cable properties sheet in Table B.8 is used to specify the relevant general properties of the hangers and the main cables. First, the LLRB's main cables are separated into multiple wires in a 3x4 pattern. This can be changed in this section, allowing for rows (Height - integer greater than 0) and up to nine columns (Width - integer from 1 to 9) in the main cable pattern. The cable damage sheet setup causes the limit of nine rows but can easily be scaled for greater values. The pattern's centre-to-centre spacing (m) between consecutive wires is provided. Thereafter, the size of the cables is provided as a linear mass (kg/m). This value is considered in conjunction with the material's density when calculating the cable area. For the main cables, the value provided is for the combined bundled. The horizontal length of restraint (m) of the cable connectors is added as a constant for all main cable connections. The mesh size for the cables is by number of elements per edge, which can be specified within this sheet. This value could be increased to the desired value for studies in which each main cable wires are modelled as n elements. Lastly, the profile properties for the cable connection parts are provided.

Table B.8: Cable Properties Input Sheet

Cable Bundle	# Cables	Spacing
Width	3	0.10795
Height	4	0.09525

Length of Restraint
0.3048

	Mesh Element
Hanger	1
Main Cable	1

	Linear Cable Mass
Hanger	10.99
Total Main Cable	150

	Cable Connection Profile
Area	1E-01
Ix	1E-01
Ixy	0
Iy	1E-01
J	1E-01
Gamma O	0
Gamma W	0

B.7 Hanger Coordinates Sheet

Concerning the cables, the next sheet in Table B.9 provides the coordinates of the bottom and top of the hangers and the vertical length of the ties (the distance from the hanger top to the cable connector centre). These are provided per bridge region, with a column for the start to the first tower side span, another for the first tower to the second tower main span, and a last one for the second tower to the end side span. It should be noted that the last value for a region should always match the first value of the next region.

Table B.9: Hanger Coordinates Input Sheet

Hanger Bottom		
Start to Tower1	Tower1 to Tower2	Tower2 to End
0	1.744	1.744
0.446	1.928	1.534
0.757	2.082	1.301
1.042	2.210	1.042
1.301	2.312	0.757
1.534	2.388	0.446
1.744	2.388	0
	2.389	
	2.388	
	2.388	
	2.312	
	2.210	
	2.082	
	1.928	
	1.744	

Hanger Top		
Start to Tower1	Tower1 to Tower2	Tower2 to End
0	19.64	19.64
1.949	14.876	14.447
4.012	11.174	10.273
6.795	8.145	6.795
10.273	5.790	4.012
14.447	4.107	1.949
19.64	3.243	0
	2.907	
	3.243	
	4.107	
	5.790	
	8.145	
	11.174	
	14.876	
	19.64	

Ties		
Start to Tower1	Tower1 to Tower2	Tower2 to End
0.599	0	0
0.3302	0.3556	0.3556
0.3556	0.3556	0.3556
0.3556	0.3556	0.3556
0.3556	0.3556	0.3556
0.3556	0.3556	0.3302
0	0.2100	0.599
	0.2100	
	0.2100	
	0.3556	
	0.3556	
	0.3556	
	0.3556	
	0.3556	
	0	

B.8 Deck Spans Sheet

The lengths of the spans are then in the following sheet as Table B.10, which controls the distance along the X axis between consecutive deck spans. Naturally, this also governs the geometry of the cable system, which is attached to the girders. The position of the first tower can be obtained through the summation of the spans in the first region, and similarly, the position of the second tower can be obtained with the first two regions.

Table B.10: Deck Spans Input Sheet

Start to Tower1	Tower1 to Tower2	Tower2 to End
15.3990	12.0142	12.0142
11.7856	11.7856	11.7856
11.7856	11.7856	11.7856
11.7856	11.7856	11.7856
11.7856	11.7856	11.7856
12.0142	11.7856	15.3990
	11.7856	
	11.7856	
	11.7856	
	11.7856	
	11.7856	
	11.7856	
	11.7856	
	12.0142	

B.9 Main Cable Restraints Sheet

The main cable restraints are specified as the number of restraints per span per region in the bridge in Table B.11. For the LLRB, all spans have at least a single restraint, but values of zero may be entered to disable the feature. Furthermore, the feature is disabled automatically if there is a single main cable wire instead of a bundled pattern.

Table B.11: Main Cable Restraints Input Sheet

Start to Tower1	Tower1 to Tower2	Tower2 to End
2	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	2
	1	
	1	
	1	
	1	
	1	
	1	
	1	
	1	

B.10 Hanger Beams Sheet

As previously mentioned, some hanger locations do not have cables but rather steel beams. The locations where this occurs are specified as 1 in the sheet presented in Table B.12, whereas regular hangers are left as 0. It should be noted that this system considers the anchors and towers as a row in the table, which must be left as zero. Moreover, a region's last term must match the next region's first term, representing the same tower location. The profile of the hanger beam can then be defined.

Table B.12: Hanger Beams Input Sheet

Start to Tower1	Tower1 to Tower2	Tower2 to End
0	0	0
1	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	1	0
	1	
	1	
	0	
	0	
	0	
	0	
	0	
	0	

	Hanger Beam Profile
Area	0.01391
Ix	3.864E-05
Ixy	0
Iy	2.261E-04
J	5.862E-06
Gamma O	0
Gamma W	0

B.11 Cable Pretensions Sheet

The subsequent sheet, presented in Table B.13, relates to the pretension values in the hangers and the main cables. These pretensions are applied in the model at the start of the Vertical Loading step and are obtained through the process explained in Section 4.3.2. For the hangers, a region's last value must match the next region's first value. The anchors and the towers are considered in

this table, for which the pretension values should remain zero. The values defined for the main cables are for the entire main cable bundle.

Table B.13: Cable Pretensions Input Sheet

Hangers		
Start to Tower1	Tower1 to Tower2	Tower2 to End
0	0	0
0	311,323	288,241
320,320	310,563	320,167
320,156	311,273	320,156
320,167	311,777	320,320
288,241	312,122	0
0	0	0
	0	
	0	
	312,122	
	311,777	
	311,273	
	310,563	
	311,323	
	0	

Main Cables		
Start to Tower1	Tower1 to Tower2	Tower2 to End
5,778,489	6,136,070	6,207,840
5,839,124	6,028,980	6,098,572
5,907,495	5,938,676	5,994,186
5,994,186	5,865,323	5,907,495
6,098,572	5,809,615	5,839,124
6,207,840	5,772,085	5,778,489
	5,742,874	
	5,742,874	
	5,772,085	
	5,809,615	
	5,865,323	
	5,938,676	
	6,028,980	
	6,136,070	

B.12 Cable Damages Sheet

This sheet is where cable damage may be specified for the main cable wires. The version presented in Table B.14 is streamlined for the LLRB, but the actual template can accommodate up to nine columns. These columns represent the columns in the bundled wires pattern and are always alphabetical and based on the number of wires in width specified in the cable properties sheet in Section B.6. The other axis is the span indices in the bridge and the rows in the bundle, defined as height in the cable properties sheet. The sheet automatically readjusts itself based on the cable properties sheet. Within the allocated cells, the cable damage as a ratio of loss area may be specified for individual wires. These values should always be less than one, as this would cause a section to have a cross-sectional area of zero. These values may be provided independently for the east and the west sides.

Table B.14: Cable Damage Input Sheet

East		Columns			West		Columns		
Spans	Rows	A	B	C	Spans	Rows	A	B	C
1	1				1	1			
1	2				1	2			
1	3				1	3			
1	4				1	4			
2	1				2	1			
2	2				2	2			
2	3				2	3			
2	4				2	4			
3	1				3	1			
3	2				3	2			
3	3				3	3			
3	4				3	4			
4	1				4	1			
4	2				4	2			
4	3				4	3			
4	4				4	4			
5	1				5	1			

B.13 Wind Loading Sheet

The last sheet, presented in Table B.15, provides the parameters for the wind loading on the bridge. This sheet is considered by the program only when wind loading is activated in the general parameters sheet (Section B.3). The wind signal can be passed in this sheet as a series of time (s) and wind velocity (m/s) points. While the program has no limit on the signal length, the jobs in Abaqus may abort for longer signals. Then, the air density (kg/m^3) can be set in the wind loading parameters, followed by the appropriate components' drag, lift, and moment coefficients, along with the relevant dimensions (m) for these components.

Table B.15: Wind Loading Input Sheet

Time	Wind Velocity
0	20.40
0.25	23.39
0.5	22.64
0.75	20.93
1	19.96
1.25	23.56
1.5	23.35
1.75	19.85
2	19.34
2.25	17.73
2.5	18.08
2.75	17.84
3	22.81
3.25	22.82
3.5	23.20

	Wind Loading Parameters
Air Density	1.225
Deck Cd	1.6
Deck Cl	0.1
Deck Cm	0.05
Deck Width	9.489
Deck Height	0.572
Tower Cd	2.05
Tower Width	0.8865
Main Cable Cumulative Cd	1.5
Main Cable Width	0.180
Hanger Cd	1.2
Hanger Width	0.0422
Hanger Beam Cd	2
Hanger Beam Width	0.1826

B.14 Inputs Parsing and Processing

The first step accomplished by the program is the basic processing of the inputs in the Excel file. The inputs from the spreadsheet are parsed as an instance of the defined class Inputs. This class

contains a method called to serialize the instance's content into a JSON compatible dictionary. This object is then stored in the file bridge_inputs.json and read by the program when running into Abaqus. The Inputs class also has a method to deserialize a passed dictionary into an instance of Inputs. Therefore, the instance can be recreated within Abaqus. This transfer system is done for three reasons. The first reason is that this allows an intermediary script (after parsing but before the Abaqus launch) to modify the current inputs without affecting the original set of inputs. This is particularly useful for iterations over specific parameters, such as the hanger coordinates, the pretension of the cables, or the damage in the main cables. Each iteration can be launched from the same set of inputs, with different modifications added based on the case requirements. The second reason is related to the storage of case inputs/outputs. While the size difference between the Excel inputs file (100+ KB) and the JSON inputs file (30+ KB) is not significant when compared to the outputs file (500+ KB), the real advantage comes from the ease of storage, the ease of reading by a Python script for results analysis, and the ease of implementation of version control. The third and last reason is that since Abaqus (2022) is running on Python 2.7, many important Python libraries are not supported anymore. Therefore, it is much easier to handle as many tasks as possible, such as input parsing, using one of the latest Python versions (3.10 in this case) before the launch of Abaqus. These changes start with Abaqus 2024, where the scripting interface uses Python 3. However, this also means that a program written for Abaqus 2024 is not backward compatible with older versions. Most of the current program was written with the forward compatibility of Python 3 in mind, so minimal changes would be required to upgrade to a working version in Abaqus 2024. Overall, this method of passing the parameters is more appropriate to the current settings.

Appendix C

Cable Pretension Values

Table C.1: Pretension Values in the Hangers and the Main Cables

Location Index	Hanger Pretension (N)	Hanger Starting Temperature	Main Cable Wire Pretension (N)	Main Cable Wire Starting Temperature (K)
0	Anchor	-		
			5,778,489	422.234
1	Beam Hanger	-		
			5,839,124	423.590
2	320,320	390.778		
			5,907,495	425.119
3	320,156	390.728		
			5,994,186	427.058
4	320,167	390.731		
			6,098,572	429.393
5	288,241	380.986		
			6,207,840	431.836
T1	Tower	-		
			6,136,070	430.231
7	311,323	388.032		
			6,028,980	427.836
8	310,563	387.799		
			5,938,676	425.817
9	311,273	388.016		
			5,865,323	424.176
10	311,777	388.170		
			5,809,615	422.930
11	312,122	388.275		
			5,772,085	422.091
12	Beam Hanger	-		
			5,742,874	421.437
13	Beam Hanger	-		
↓↓↓ Symmetry about index 13 ↓↓↓				

Appendix D

Basic Python Commands for Abaqus

Algorithm D.1 Elastic Material Creation Commands

```
# Creates a material in Abaqus with the name from the name attribute ElasticMaterial
model.Material(name=name)
# Creates a reference to the Abaqus Material object
material = model.materials[name]
# Provides the density value to the material
material.Density(table=((mass_density,),))
# Provides the elastic modulus value and the Poisson's ratio value to the material
material.Elastic(table=((young_mod, poisson_ratio),))
# Provides the expansion coefficient value to the material, if available
if expansion_coeff is not None:
  material.Expansion(table=((expansion_coeff,),))
# Provides the conductivity value to the material, if available
if conductivity is not None:
  material.Conductivity(table=((conductivity,),))
# Provides the specific heat value to the material, if available
if specific_heat is not None:
  material.SpecificHeat(table=((specific_heat,),))
# Provides the appropriate no compression keyword to the material, if defined
if no_compression:
  material.elastic.setValues(noCompression=ON)
```

Algorithm D.2 Static General Step Creation Commands

```
# Creation of the StaticStep in Abaqus
model.StaticStep(name=name, previous=pre_name, description=description)
# Creation of the reference to the Abaqus Step object
step_ref = model.steps[name]

# Time incrementation method set to AUTOMATIC, triggered if the specified method is "Automatic"
time_inc_method = AUTOMATIC
# Use default values for increment parameters that are None (null in the inputs file)
initial_inc = 1 if initial_inc is None else initial_inc
min_inc = 1e-5 if min_inc is None else min_inc
max_inc = 1 if max_inc is None else max_inc
inc_number = 100 if inc_number is None else inc_number
# Set time increment values for the step
step_ref.setValues(
  timeIncrementationMethod=time_inc_method,
  initialInc=initial_inc,
  minInc=min_inc,
  maxInc=max_inc,
  maxNumInc=inc_number
)

# Determine the keyword to use to activate or deactivate the non linear geometry setting
nlgeom_keyword = ON if nlgeom == True else OFF
# Set the nlgeom parameter for the step
step_ref.setValues(nlgeom=nlgeom_keyword)

# Stabilization method set to DISSIPATED_ENERGY_FRACTION, triggered if the specified method is
# "Dissipated_Energy_Fraction". The continueDampingFactors parameter is automatically set to
# False for this method (True for the "Damping_Factor_From_Prev" method).
stab_method = DISSIPATED_ENERGY_FRACTION
continueDampingFactors = False
# Use default values for stabilization parameters when necessary
stab_magn = 0.0002 if stab_magn is None else stab_magn
adap_damp_ratio = None if adap_damp_ratio == "None" else adap_damp_ratio
adap_damp_ratio = adap_damp_ratio if isinstance(adap_damp_ratio, (float, int)) else None
# Set stabilization parameters for the step
step_ref.setValues(
  stabilizationMagnitude=stab_magn,
  stabilizationMethod=stab_method,
  continueDampingFactors=continueDampingFactors,
  adaptiveDampingRatio=adap_damp_ratio
)

# Set the adiabated heating effect to ON is specified as True in the inputs
if adiabatic:
  step_ref.setValues(adiabatic=ON)
```

Algorithm D.3 Frequency Step Creation Commands

```
# Creation of the FrequencyStep in Abaqus
model.FrequencyStep(
  eigensolver=SUBSPACE,
  numEigen=numEigen,
  name=name,
  previous=pre_name,
  description=description
)
# Creation of the reference to the Abaqus Step object
step_ref = model.steps[name]
```

Algorithm D.4 Vertex Set Creation Commands

```
# Create the standard name for the set
set_name = "set_{0}".format(set_name)
# Get the available part vertices
v = part_ref.vertices
# Find the vertex at the specified coordinates
myVert = v.findAt(coords,)
# Create a tuple with the found vertex
vertex = (v[myVert.index : myVert.index + 1],)
# Create a set with the found vertex
part_ref.Set(vertices=vertex, name=set_name)
```

Algorithm D.5 Mesh By Factor Creation Commands

```
# Select the region as the part
region = model.parts[part_name]
# Seed the region by edge by size
region.seedEdgeBySize(
  constraint=FINER,
  minSizeFactor=minSizeFactor,
  edges=region.sets['set_alledges'].edges,
  size=size,
  deviationFactor=deviation_factor
)
# Generate the mesh on the region
region.generateMesh()
# Assign the element type based on the part type (beam or truss)
if part_type == "WireBeam":
  elemType = ElemType(elemCode=B31, elemLibrary=STANDARD)
  picked_regions = (region.sets['set_alledges'].edges,)
  region.setElementType(
    regions=picked_regions,
    elemTypes=(elemType,)
  )
elif part_type == "WireTruss":
  elemType = ElemType(elemCode=T3D2, elemLibrary=STANDARD)
  picked_regions = (region.sets['set_alledges'].edges,)
  region.setElementType(
    regions=picked_regions,
    elemTypes=(elemType,)
  )
```

Algorithm D.6 Mesh By Value Creation Commands

```
# Seed the region by edge by size
region.seedEdgeByNumber(
  constraint=FINER,
  edges=region.sets['set_alledges'].edges,
  number=number
)
```

Algorithm D.7 Section Assignment Command

```
# Assign the section to the set_alledges by name
part_ref.SectionAssignment(
  region=part_ref.sets["set_alledges"],
  sectionName=section.name,
  offset=0.0,
  offsetType=MIDDLE_SURFACE,
  offsetField="",
  thicknessAssignment=FROM_SECTION
)
```

Algorithm D.8 Beam Orientation Assignment Command

```
# Assign the beam orientation to set_alledges by name
part_ref.assignBeamSectionOrientation(
  method=N1_COSINES,
  n1=normal_direction,
  region=part_ref.sets['set_alledges']
)
```

Algorithm D.9 Coupling Constraint Creation Command

```
# Define the type of multipoint constraint
mpcType = BEAM_MPC
# Select the master set for the constraint in the assembly
master = assembly.instances[master_instance_name].sets[master_set_name]
# Select the slave set for the constraint in the assembly
slave = assembly.instances[slave_instance_name].sets[slave_set_name]
# Create the multipoint constraint
model.MultipointConstraint(
  controlPoint=master,
  csys=None,
  mpcType=mpcType,
  name=name,
  surface=slave,
  userMode=DOF_MODE_MPC,
  userType=0
)
```

Algorithm D.10 Multipoint Constraint Creation Command

```
# Define the degrees of freedom
u1 = OFF if u1 not in [ON, OFF] else u1
u2 = OFF if u1 not in [ON, OFF] else u2
u3 = OFF if u1 not in [ON, OFF] else u3
ur1 = OFF if u1 not in [ON, OFF] else ur1
ur2 = OFF if u1 not in [ON, OFF] else ur2
ur3 = OFF if u1 not in [ON, OFF] else ur3
# Select the master set for the constraint in the assembly
master = assembly.instances[master_instance_name].sets[master_set_name]
# Select the slave set for the constraint in the assembly
slave = assembly.instances[slave_instance_name].sets[slave_set_name]
# Create the coupling constraint
model.Coupling(
  name=name,
  controlPoint=master,
  surface=slave,
  couplingType=KINEMATIC,
  influenceRadius=WHOLE_SURFACE,
  localCsys=None,
  u1=u1,
  u2=u2,
  u3=u3,
  ur1=ur1,
  ur2=ur2,
  ur3=ur3
)
```

Algorithm D.11 Pinned Boundary Condition Creation Command

```
# Select the region by the set
region = assembly.instances[instance_name].sets[set_name]
# Create the pinned boundary condition on the region
model.PinnedBC(
  name=name,
  createStepName=step_name,
  region=region,
  localCsys=None
)
```

Algorithm D.12 Encastre Boundary Condition Creation Command

```
# Select the region by the set
region = assembly.instances[instance_name].sets[set_name]
# Create the encastre boundary condition on the region
model.EncastreBC(
  name=name,
  createStepName=step_name,
  region=region,
  localCsys=None
)
```

Algorithm D.13 Displacement Boundary Condition Creation Command

```
# Select the region by the set
region = assembly.instances[instance_name].sets[set_name]
# Unset the components of the displacement / rotation vectors that are not specified as numbers
disp = [d if isinstance(d, (float, int)) else UNSET for d in disp]
# Create the displacement and rotation vectors
u = (disp[0], disp[1], disp[2])
ur = (disp[3], disp[4], disp[5])
# Create the displacement boundary condition on the region
model.DisplacementBC(
  name=name,
  createStepName=step_name,
  region=region,
  localCsys=None,
  amplitude=UNSET,
  distributionType=UNIFORM,
  fieldName='',
  fixed=OFF,
  u1=u[0],
  u2=u[1],
  u3=u[2],
  ur1=ur[0],
  ur2=ur[1],
  ur3=ur[2]
)
```

Algorithm D.14 Predefined Field Creation Command for Pretension

```
# Define the base temperature for the region
base_temp = 293.0
# Select the region by the set (set_alledges)
region = assembly.instances[instance_name].sets[set_name]
# Create the temperature in the Initial step
step_name = "Initial"
model.Temperature(
  createStepName=step_name,
  crossSectionDistribution=CONSTANT_THROUGH_THICKNESS,
  distributionType=UNIFORM,
  magnitudes=(base_temp + temp_diff, ),
  name=name,
  region=region
)
# Change the temperature in the Vertical Loading step
step_name = "Vertical Loading"
model.predefinedFields[name].setValuesInStep(
  magnitudes=(base_temp, ),
  stepName=step_name
)
```

Algorithm D.15 Job Creation and Submission Command

```
# Use basic parameters for no parallelization
numCpus = 1
numDomains = 1
numGPUs = 0
# Define the job in Abaqus
mdb.Job(
  name=name,
  model=model_name,
  description=description,
  type=ANALYSIS,
  atTime=None,
  waitMinutes=0,
  waitHours=0,
  queue=None,
  memory=90,
  memoryUnits=PERCENTAGE,
  getMemoryFromAnalysis=True,
  explicitPrecision=SINGLE,
  nodalOutputPrecision=SINGLE,
  echoPrint=OFF,
  modelPrint=OFF,
  contactPrint=OFF,
  historyPrint=OFF,
  userSubroutine="",
  scratch="",
  resultsFormat=ODB,
  numThreadsPerMpiProcess=1,
  multiprocessingMode=DEFAULT,
  numCpus=numCpus,
  numDomains=numDomains,
  numGPUs=numGPUs
)
# Regenerate the assembly
mdb.models[model_name].rootAssembly.regenerate()
# Submit the job in Abaqus
mdb.jobs[name].submit(consistencyChecking=OFF)
```

Appendix E

Polynomial Root Finding Algorithm

Algorithm E.1 Polynomial Root Finding Function

```
import numpy as np
import numpy.polynomial.polynomial as poly

def get_best_poly_root(x, y, max_degree=1, step_factor=0.1):
  coefs = poly.polyfit(x, y, deg=min(max_degree, len(x) - 1))
  roots = poly.polyroots(coefs)
  valid_roots = roots[(roots >= min(x)) & (roots <= max(x))]
  if len(valid_roots) > 0:
    return valid_roots[np.argmin(abs(valid_roots - np.mean(x)))]
  else:
    extrapolation_step = abs(max(x) - min(x)) * step_factor
    eval_min = poly.polyval(min(x) - extrapolation_step, coefs)
    eval_max = poly.polyval(max(x) + extrapolation_step, coefs)
    if abs(eval_min) < abs(eval_max):
      return min(x) - extrapolation_step
    else:
      return max(x) + extrapolation_step
```

Appendix F

Genetic Algorithm

Algorithm F.1 Genetic Algorithm Next Individual Creation

```python
import numpy as np
import random

def create_individual(
  current_generation, gene_bounds, sigma_factor=8, tournsize=7, tol=1e-5,
  mate_prob=0.5, mate_gene_prob=0.5, mutate_prob=0.3, mutate_gene_prob=0.2,
):
  if len(current_generation) < 2:
    return create_random_individual(gene_bounds)
  initial_population = Population()
  for case_values in current_generation.values():
    initial_population.add_individual(
      individual=Individual(genes=case_values["inputs"], score=case_values["score"])
    )
  tournsize = min(tournsize, len(initial_population.individuals))

  while True:
    offspring = tournament_selection(
      population=initial_population,
      k=len(initial_population.individuals),
      tournsize=tournsize,
    )
    for child1, child2 in zip(offspring[::2], offspring[1::2]):
      if random.random() < mate_prob:
        mate(child1, child2, mate_gene_prob)
    for mutant in offspring:
      if random.random() < mutate_prob:
        mutate(mutant, sigma_factor, mutate_gene_prob, gene_bounds)
    next_genes = list(random.choice(offspring).genes)
    if not any(
      np.allclose(next_genes, list(values["inputs"]), atol=tol)
      for _, values in current_generation.items()
    ):
      break
  return next_genes
```

Algorithm F.2 Genetic Algorithm Functions

```
import random
import copy

class Individual():
  def __init__(self, genes=None, score=None):
    self.genes = genes
    self.score = score
  def set_genes(self, genes):
    self.genes = genes
  def set_score(self, score):
    self.score = score

class Population():
  def __init__(self):
    self.individuals = []
  def add_individual(self, individual):
    self.individuals.append(individual)

def create_random_individual(gene_bounds):
  random_genes = []
  for gene_range in gene_bounds:
    random_genes.append(random.uniform(gene_range[0], gene_range[1]))
  return random_genes

def mate(individual1, individual2, gene_prob):
  for i in range(len(individual1.genes)):
    if random.random() < gene_prob:
      gene1 = individual1.genes[i]
      gene2 = individual2.genes[i]
      individual1.genes[i] = gene2
      individual2.genes[i] = gene1

def mutate(individual, sigma_factor, gene_prob, gene_bounds):
  for i in range(len(individual.genes)):
    if random.random() < gene_prob:
      min_bound, max_bound = gene_bounds[i]
      mu = 0
      sigma = (max_bound - min_bound) / sigma_factor
      while True:
        new_gen = individual.genes[i] + random.gauss(mu, sigma)
        if min_bound <= new_gen <= max_bound:
          individual.genes[i] = new_gen
          break

def tournament_selection(population, k, tournsize=7):
  winners = []
  for _ in range(k):
    aspirants = random.sample(population.individuals, tournsize)
    winner = max(aspirants, key=lambda ind: ind.score)
    winners.append(copy.deepcopy(winner))
  return winners
```

Appendix G

Wind Velocity Signal Generation

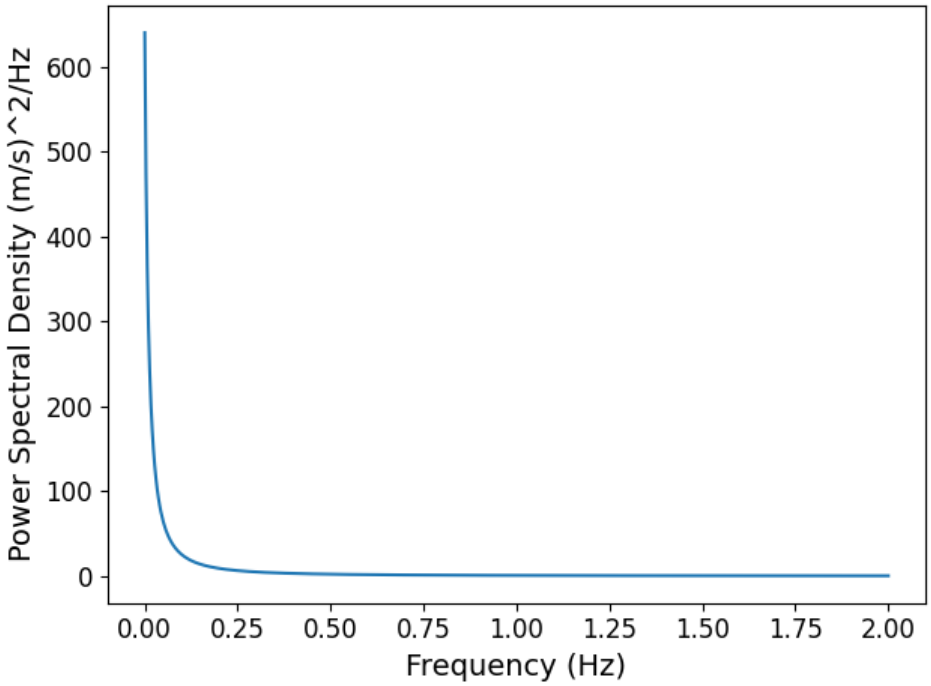


Figure G.1: Kaimal Spectrum

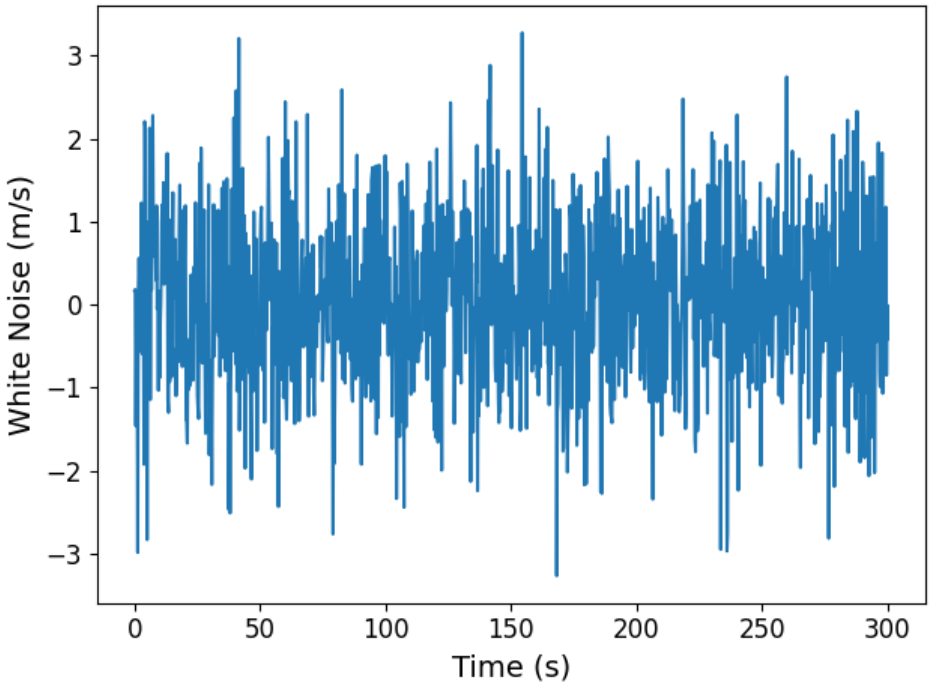


Figure G.2: White Noise in the Time-Domain

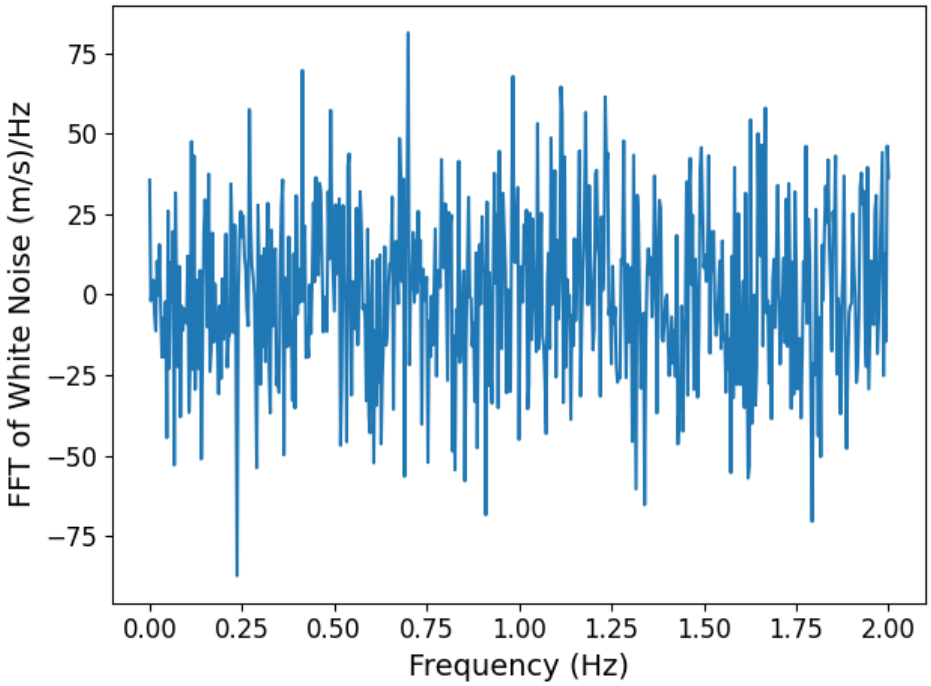


Figure G.3: White Noise in the Frequency-Domain

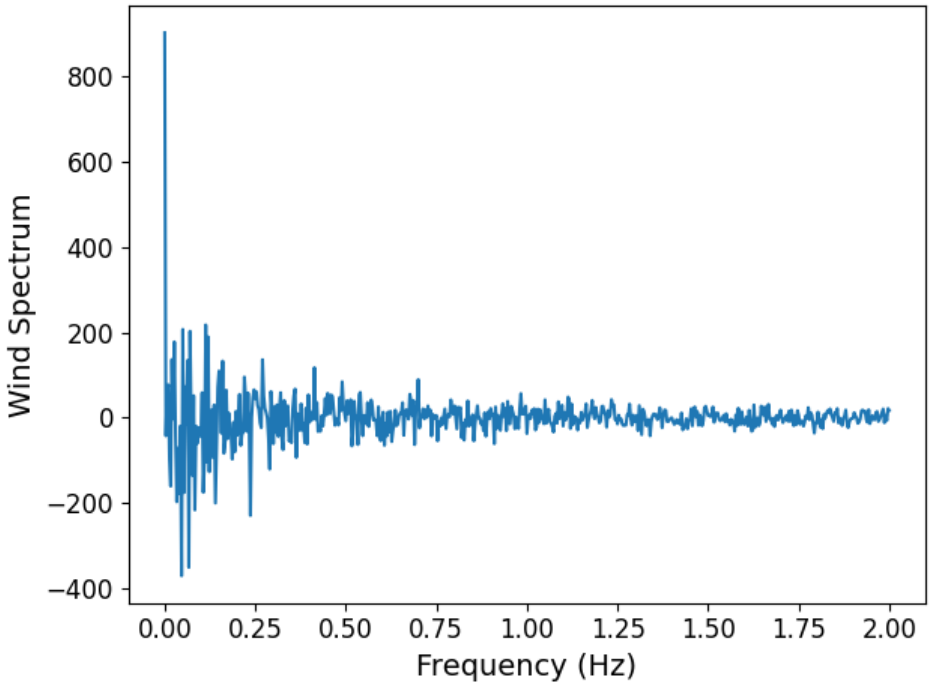


Figure G.4: Wind Spectrum

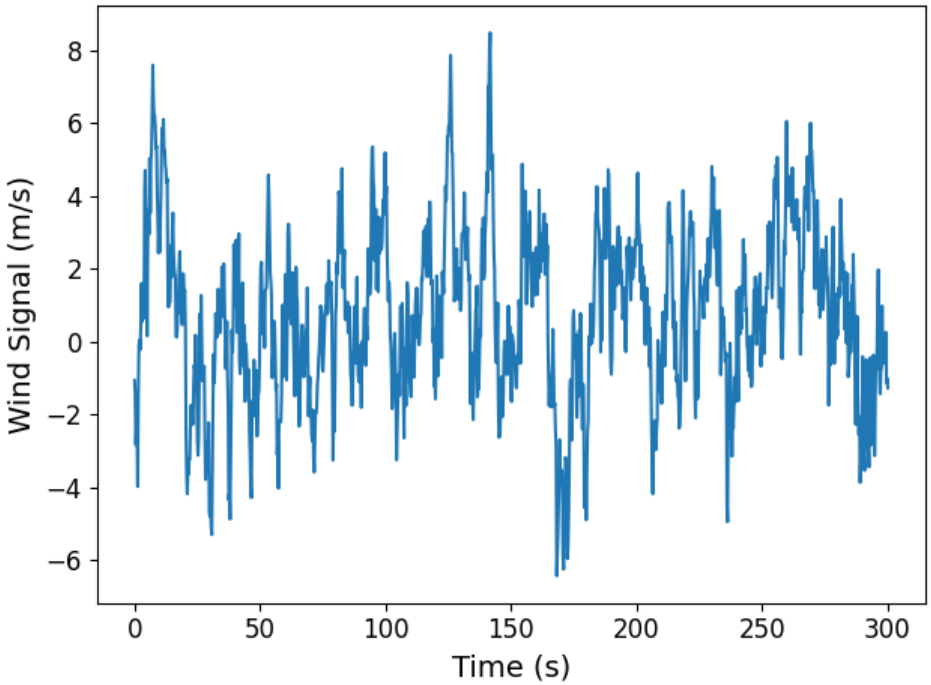


Figure G.5: Initial Wind Signal

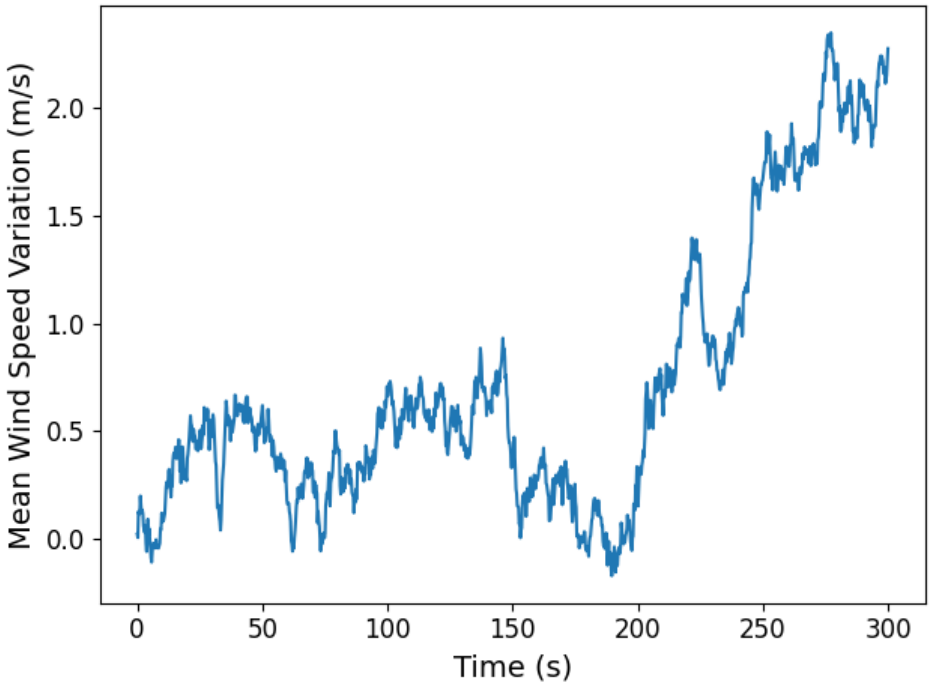


Figure G.6: Mean Wind Speed Variation

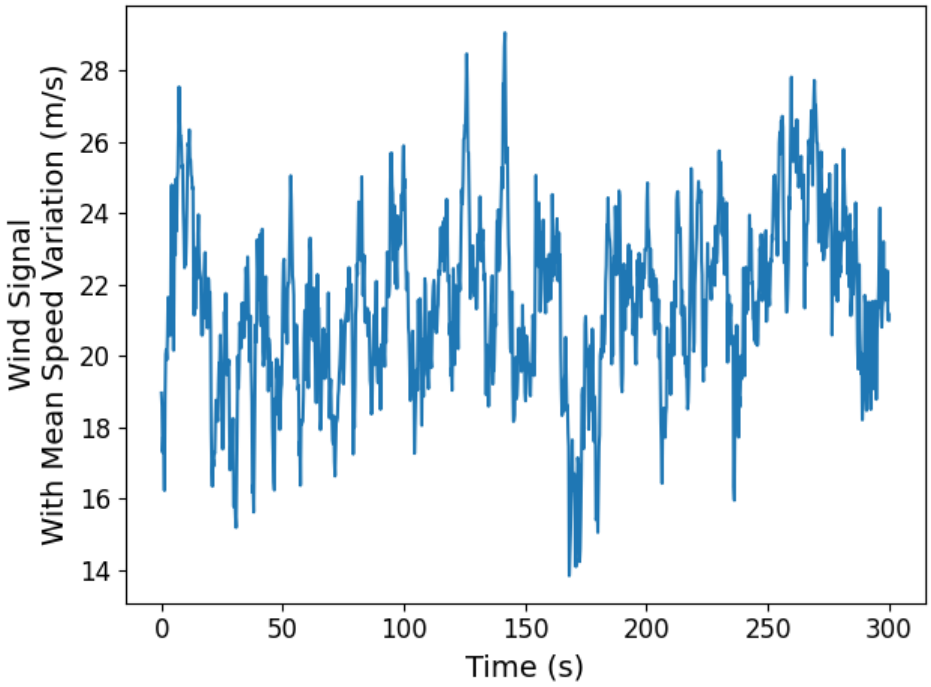


Figure G.7: Wind Signal With Mean Speed Variation

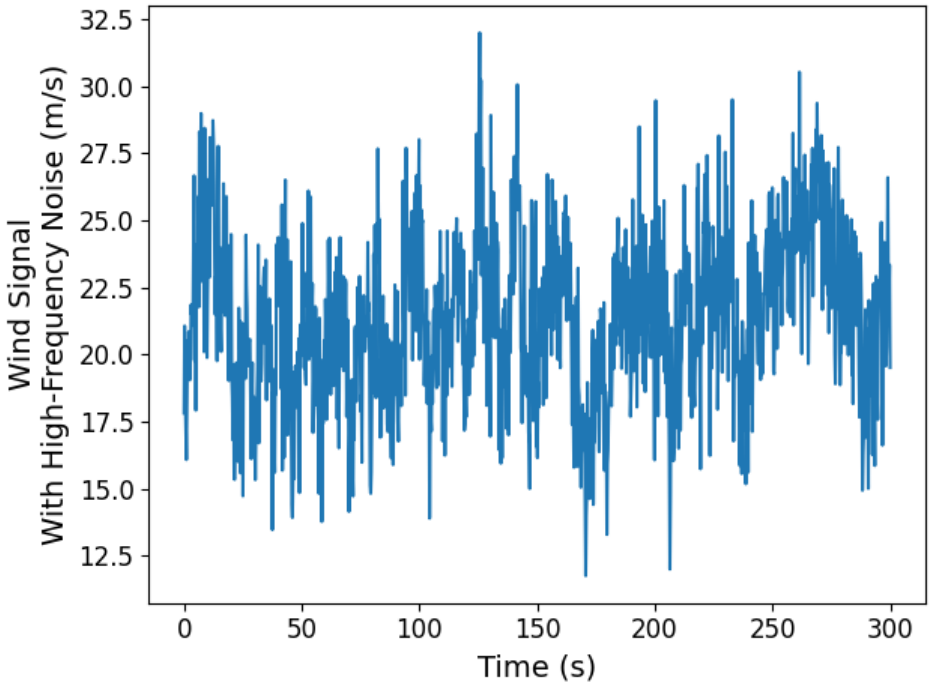


Figure G.8: Wind Signal With Mean Speed Variation and High-Frequency Noise

Appendix H

Equivalent Cable Damage Calculations

Throughout this research, damage in the main cable is presented as a simple reduction of the cross-sectional area. This assumption is made because it is considered that a strand that would break in a wire and that is unnoticed would not contribute to the cable stiffness within this span. However, this assumption becomes invalid when broken strands are clamped, as observed in Figure 2.12. In such cases, the cable experiences a loss of stiffness over a limited length along the cable, which has a lesser impact on the total stiffness of the cable. On the other hand, if the cable is not clamped and it is assumed that it has no stiffness contribution from the broken strands in the entire length of the cable, the loss in stiffness is only applicable to the sub-span, delimited by cable connector and cable restraint parts (Section 3.2.4). Since most spans have a single cable restraint along their length, the maximum length of loss of stiffness in a cable is generally half of its complete span. If a main cable span is considered as a series of springs, then the damaged and undamaged lengths' contribution to the equivalent cable stiffness can be calculated with Equation H.1.

$$\frac{1}{k_{eq}} = \frac{1}{k_u} + \frac{1}{k_d} \tag{H.1}$$

Where k_{eq} is the equivalent axial stiffness of the damaged cable (N/m), k_u is the stiffness of the undamaged length (N/m), and k_d is the stiffness of the damaged length. Here, the damaged length refers to the length on which the broken strands do not contribute to the stiffness. This length can be defined as αL, where α is the length factor of the damaged portion of the cable. Then, the axial stiffness of a cable can be defined through Equation H.2.

$$k = \frac{EA}{L} \tag{H.2}$$

Where E is the modulus of elasticity (Pa), A is the cross-sectional area (m^2), and L is the

length (m). Therefore, the cross-sectional area of the damage can be defined as $(1-\beta)\,A$, with β the damage ratio in the cable. Integrating these relations into Equation H.1 yields the following Equation H.3:

$$\begin{aligned}\frac{1}{k_{eq}} &= \frac{L-\alpha L}{EA} + \frac{\alpha L}{E\,(1-\beta)\,A} \\ \frac{1}{k_{eq}} &= \frac{(1-\beta+\alpha\beta)\,L}{(1-\beta)\,EA} \\ k_{eq} &= \frac{1-\beta}{1-\beta+\alpha\beta}\frac{EA}{L}\end{aligned} \tag{H.3}$$

Therefore, the reduced cross-sectional area applied on the cables along entire spans throughout this research can be converted to a more realistic damage level with this relation. This conversion can be obtained as a combination of Equation H.3 with Equation H.2 with a damage of γ over the entire cable's length, and solving for the damage ratio in the cable, β.

$$\begin{aligned}\frac{E\,(1-\gamma)\,A}{L} &= \frac{1-\beta}{1-\beta+\alpha\beta}\frac{EA}{L} \\ 1-\gamma &= \frac{1-\beta}{1-\beta+\alpha\beta} \\ \beta &= \frac{\gamma}{\alpha+\gamma-\alpha\gamma}\end{aligned} \tag{H.4}$$

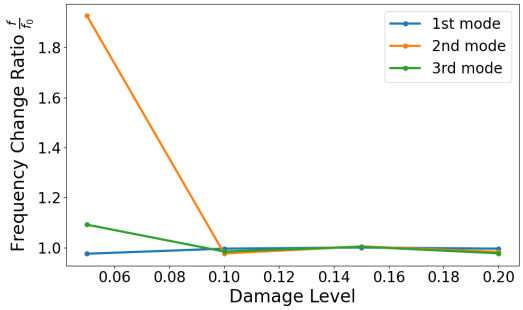
Therefore, the damage ratio presented throughout this study can be converted to a more realistic one by considering $\alpha = 0.5$. These damage equivalencies are presented in Table H.1 below, with DamageIntensity as referenced in 4.4.1. Note that for the cable spans closest to the anchors, α should be taken as $1/3$. Moreover, the real effect of clamped strands can be evaluated by adjusting α accordingly.

Table H.1: Cable Damage Equivalency

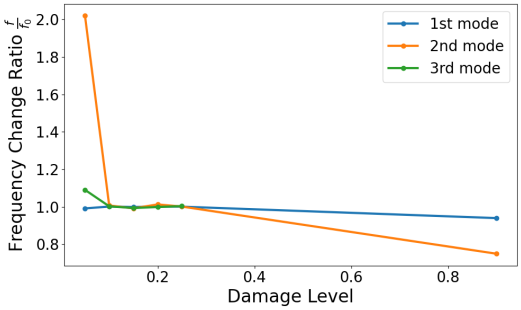
DamageIntensity	True Equivalent Damage ($\alpha = 0.5$)
0.05	0.0952
0.1	0.1818
0.15	0.2609
0.2	0.3333
0.25	0.4000
0.3	0.4615
0.5	0.6667
0.7	0.8235
0.9	0.9497

Appendix I

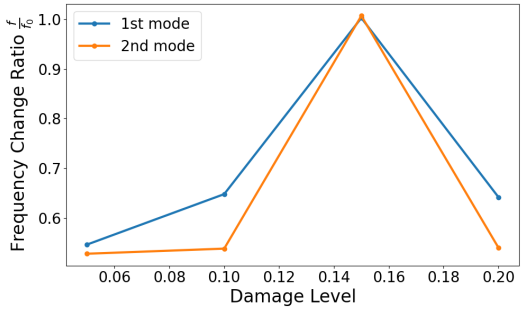
Additional Natural Frequencies Results



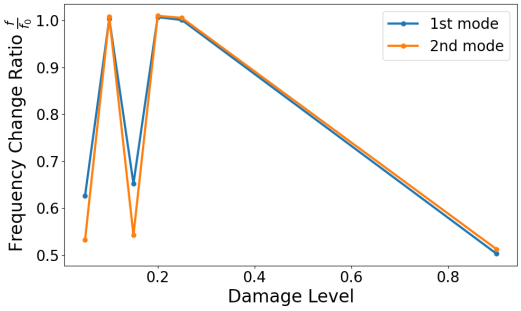
(a) Lateral Symmetric Modes - XSym



(b) Lateral Symmetric Modes - DSym

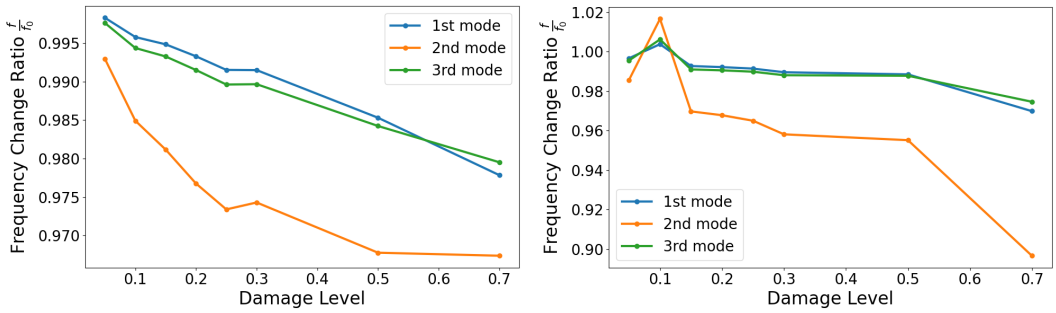


(c) Lateral Asymmetric Modes - XSym

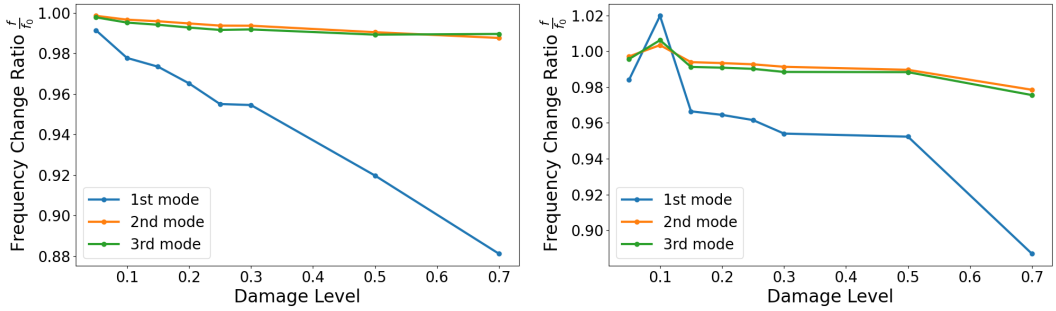


(d) Lateral Asymmetric Modes - DSym

Figure I.1: FREQ11 Quad Lateral Modes of Vibration

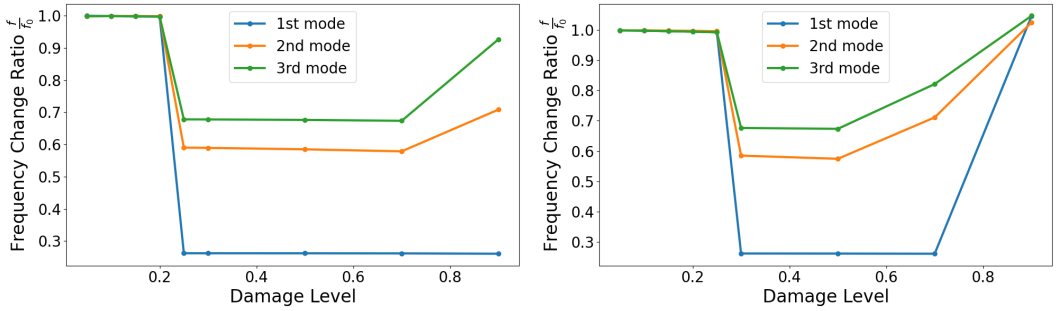


(a) Lateral Symmetric Modes - XSym (b) Lateral Symmetric Modes - DSym

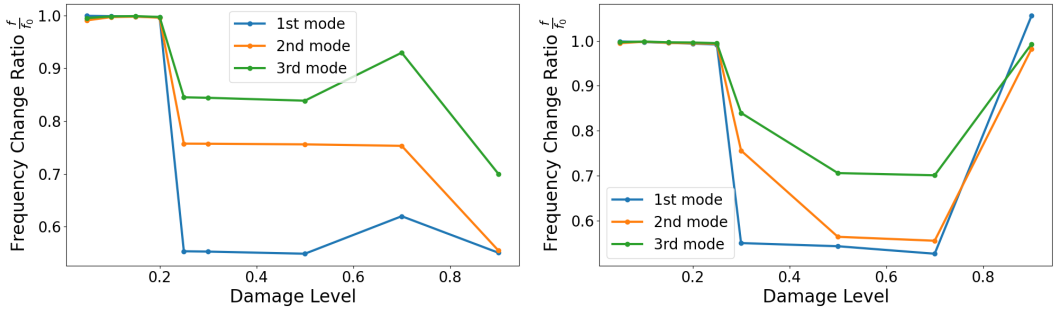


(c) Lateral Asymmetric Modes - XSym (d) Lateral Asymmetric Modes - DSym

Figure I.2: FREQ34 Quad Lateral Modes of Vibration

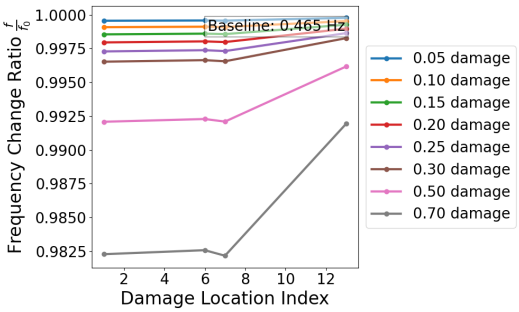


(a) Torsional Symmetric Modes - XSym (b) Torsional Symmetric Modes - DSym

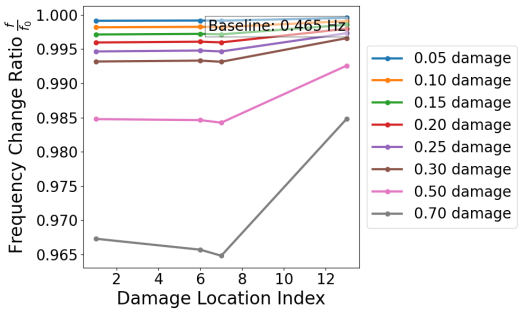


(c) Torsional Asymmetric Modes - XSym (d) Torsional Asymmetric Modes - DSym

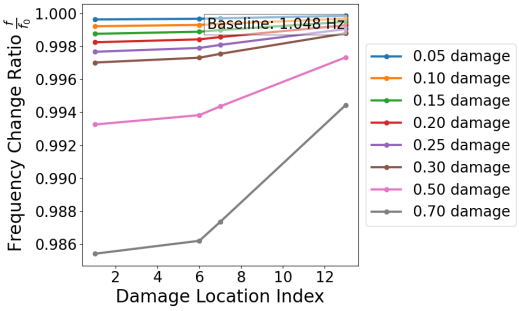
Figure I.3: FREQ11 Quad Torsional Modes of Vibration



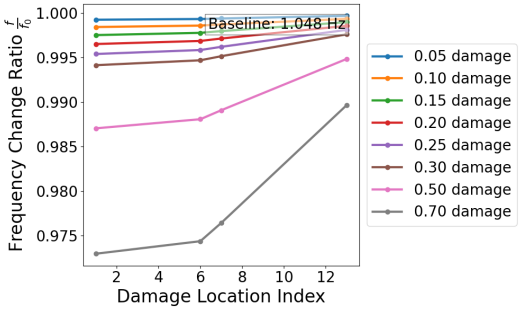
(a) 1st Vertical Symmetric Modes - XSym



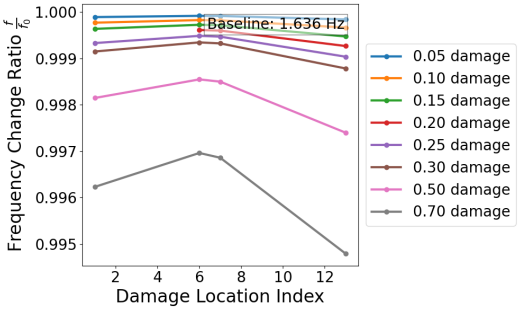
(b) 1st Vertical Symmetric Modes - DSym



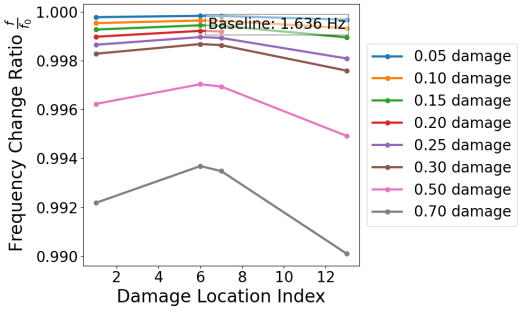
(c) 2nd Vertical Symmetric Modes - XSym



(d) 2nd Vertical Symmetric Modes - DSym

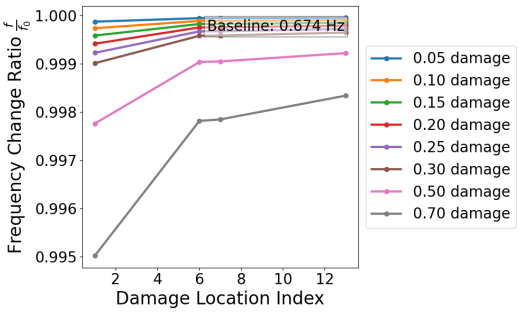


(e) 3rd Vertical Symmetric Modes - XSym

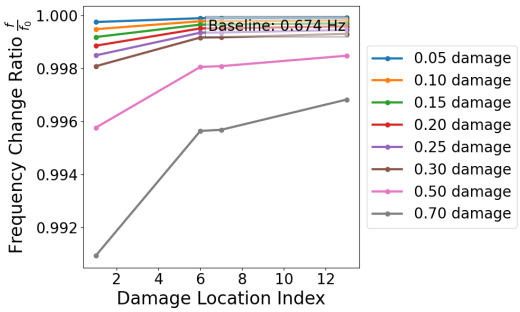


(f) 3rd Vertical Symmetric Modes - DSym

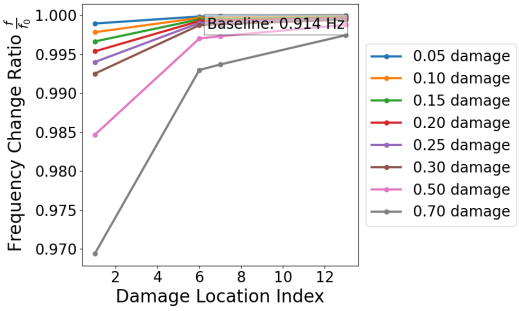
Figure I.4: FREQ34 Sing Symmetrical Vertical Modes of Vibration



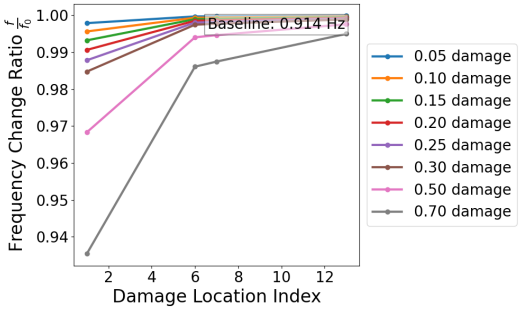
(a) 1st Vertical Asymmetric Modes - XSym



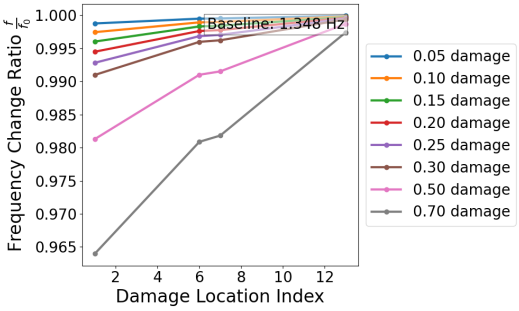
(b) 1st Vertical Asymmetric Modes - DSym



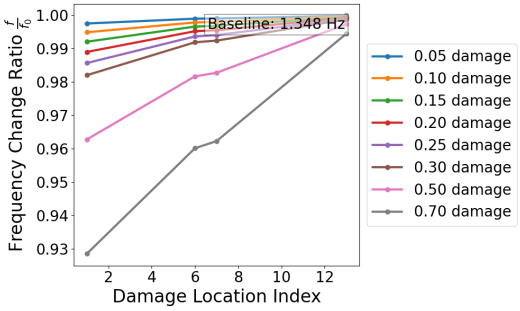
(c) 2nd Vertical Asymmetric Modes - XSym



(d) 2nd Vertical Asymmetric Modes - DSym

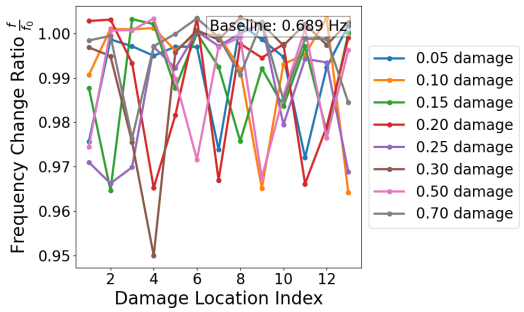


(e) 3rd Vertical Asymmetric Modes - XSym

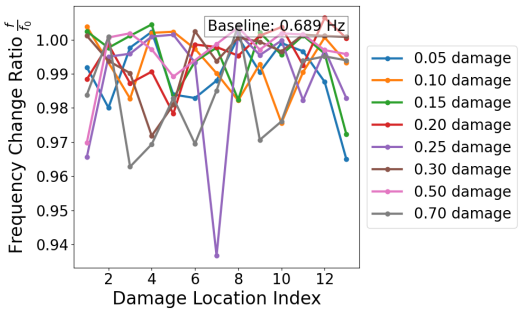


(f) 3rd Vertical Asymmetric Modes - DSym

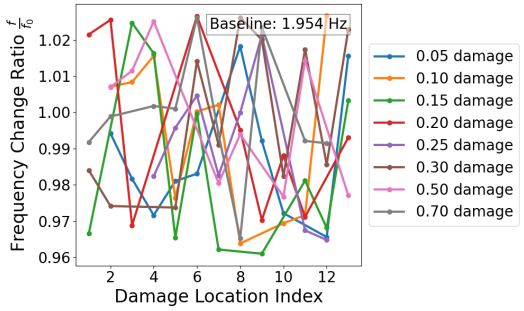
Figure I.5: FREQ34 Sing Asymmetrical Vertical Modes of Vibration



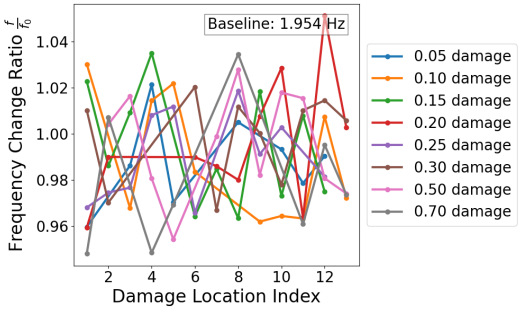
(a) 1st Lateral Symmetric Modes - XSym



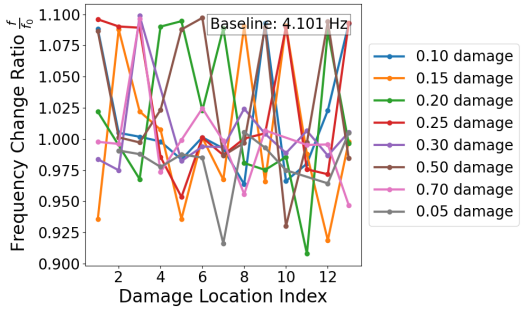
(b) 1st Lateral Symmetric Modes - DSym



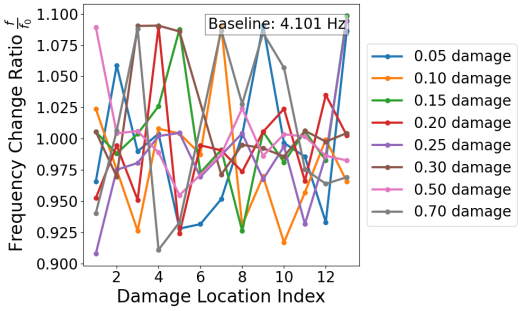
(c) 2nd Lateral Symmetric Modes - XSym



(d) 2nd Lateral Symmetric Modes - DSym

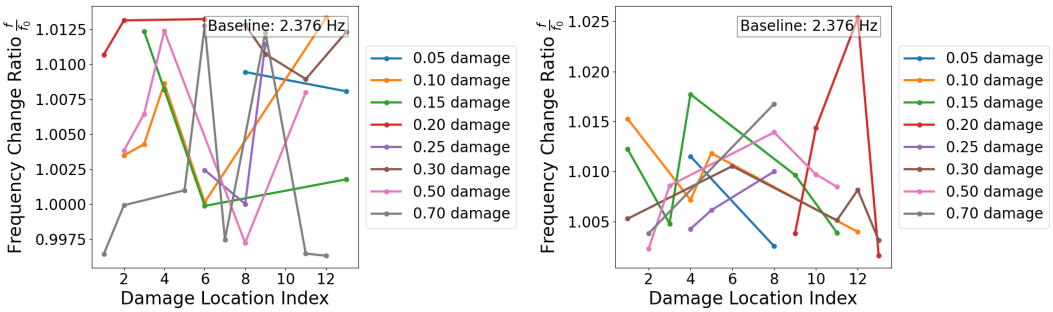


(e) 3rd Lateral Symmetric Modes - XSym

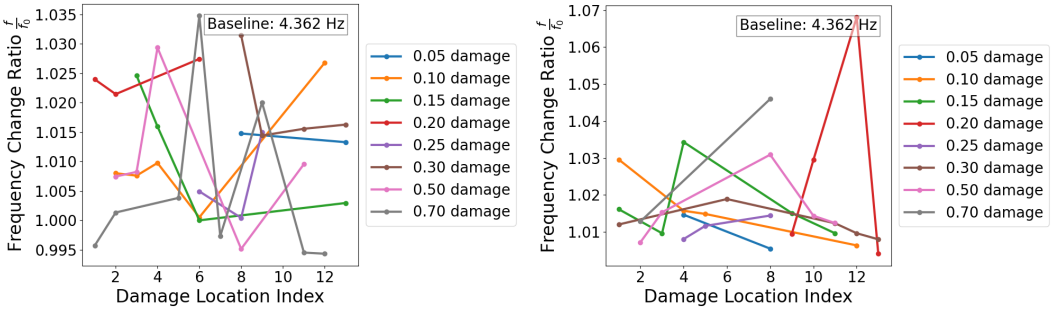


(f) 3rd Lateral Symmetric Modes - DSym

Figure I.6: FREQ11 Sing Symmetrical Lateral Modes of Vibration

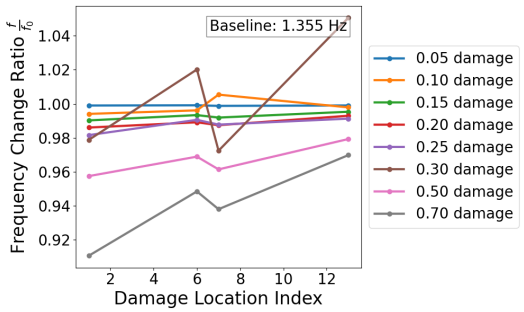


(a) 1st Lateral Asymmetric Modes - XSym (b) 1st Lateral Asymmetric Modes - DSym

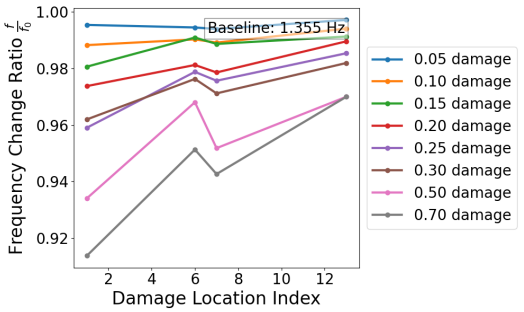


(c) 2nd Lateral Asymmetric Modes - XSym (d) 2nd Lateral Asymmetric Modes - DSym

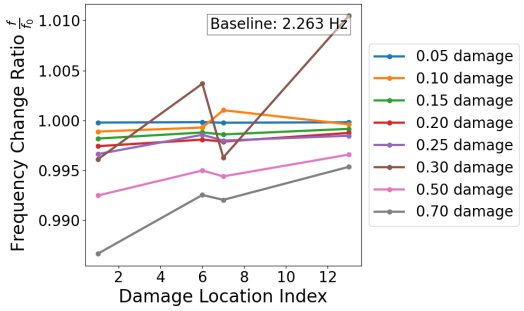
Figure I.7: FREQ11 Sing Asymmetrical Lateral Modes of Vibration



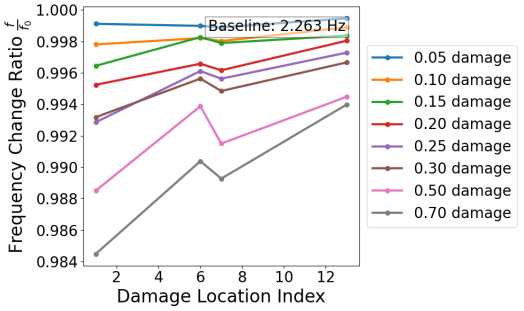
(a) 1st Lateral Asymmetric Modes - XSym



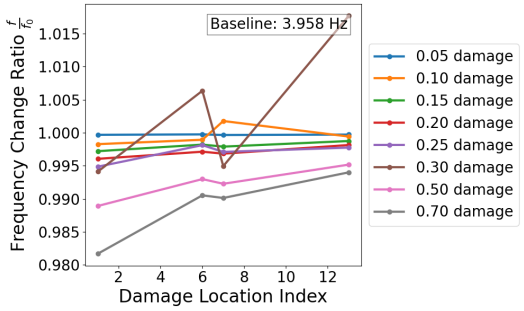
(b) 1st Lateral Asymmetric Modes - DSym



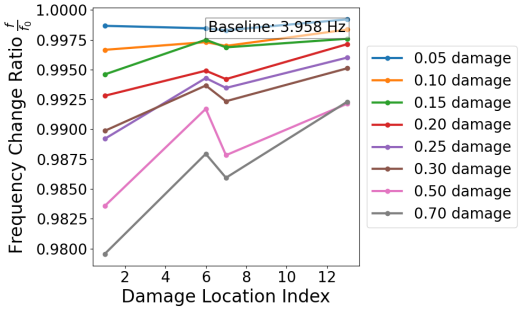
(c) 2nd Lateral Asymmetric Modes - XSym



(d) 2nd Lateral Asymmetric Modes - DSym

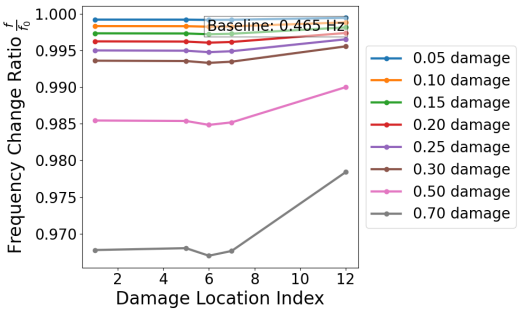


(e) 3rd Lateral Asymmetric Modes - XSym

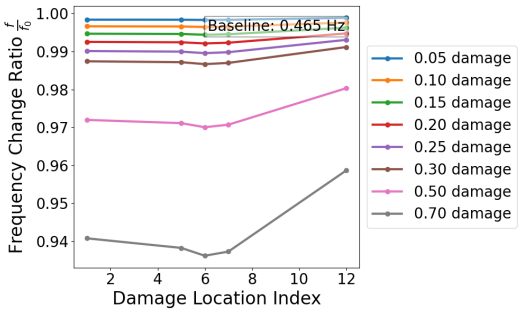


(f) 3rd Lateral Asymmetric Modes - DSym

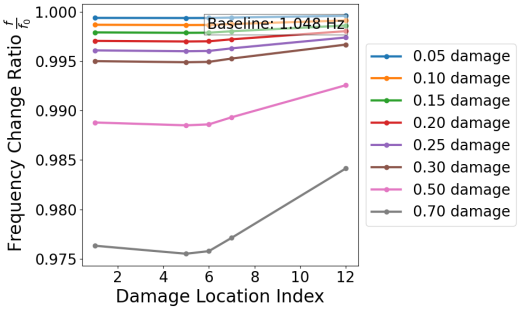
Figure I.8: FREQ34 Sing Asymmetrical Lateral Modes of Vibration



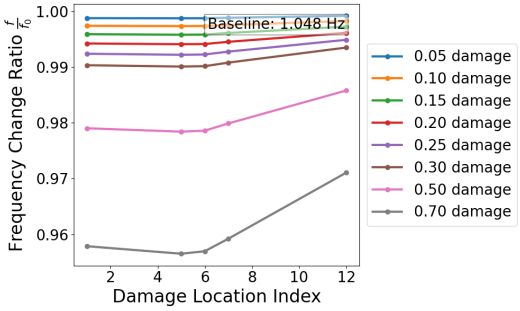
(a) 1st Vertical Symmetric Modes - XSym



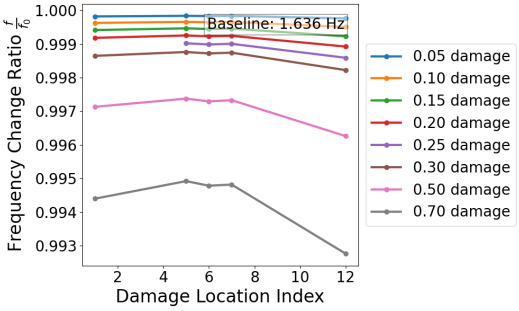
(b) 1st Vertical Symmetric Modes - DSym



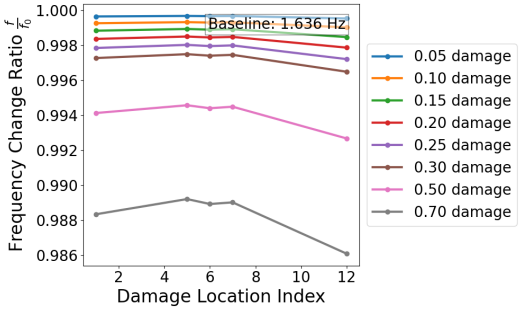
(c) 2nd Vertical Symmetric Modes - XSym



(d) 2nd Vertical Symmetric Modes - DSym

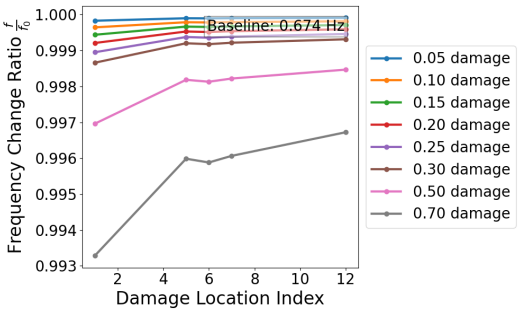


(e) 3rd Vertical Symmetric Modes - XSym

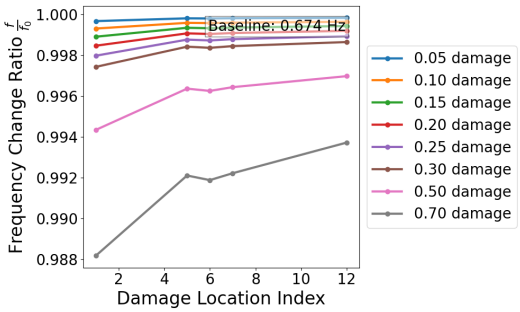


(f) 3rd Vertical Symmetric Modes - DSym

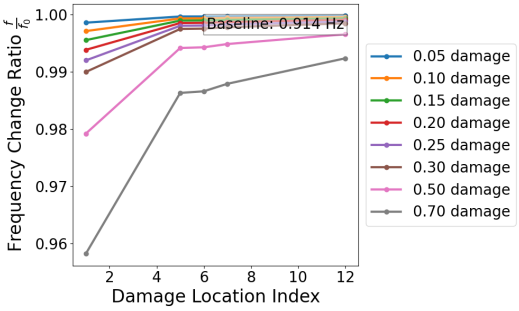
Figure I.9: FREQ34 Doub Symmetrical Vertical Modes of Vibration



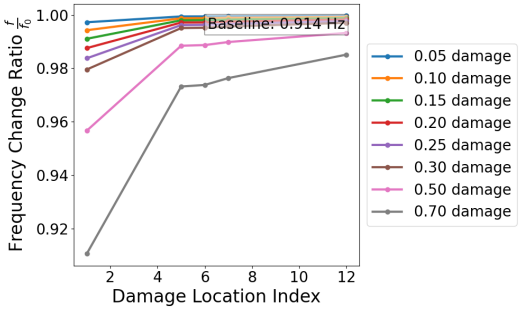
(a) 1st Vertical Asymmetric Modes - XSym



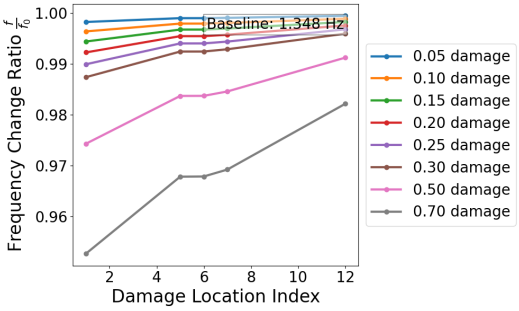
(b) 1st Vertical Asymmetric Modes - DSym



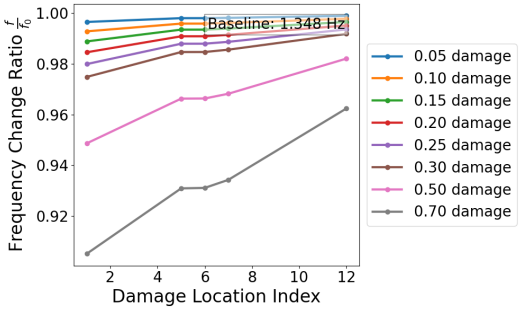
(c) 2nd Vertical Asymmetric Modes - XSym



(d) 2nd Vertical Asymmetric Modes - DSym

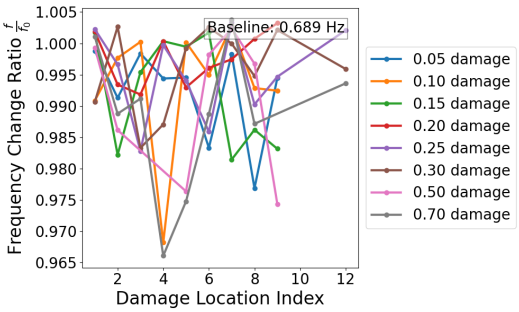


(e) 3rd Vertical Asymmetric Modes - XSym

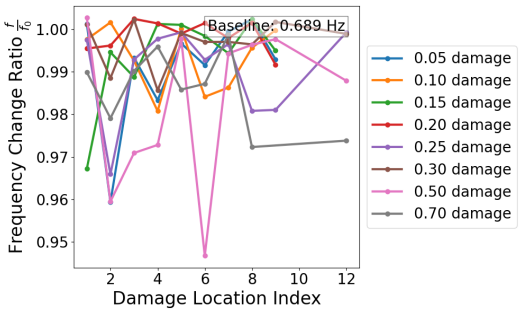


(f) 3rd Vertical Asymmetric Modes - DSym

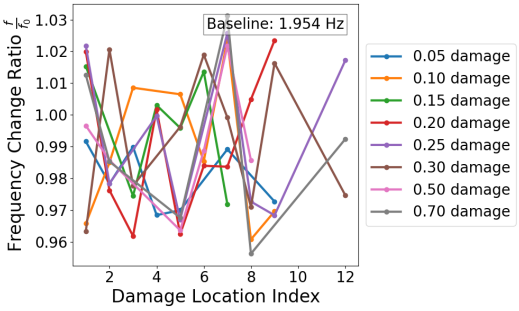
Figure I.10: FREQ34 Doub Asymmetrical Vertical Modes of Vibration



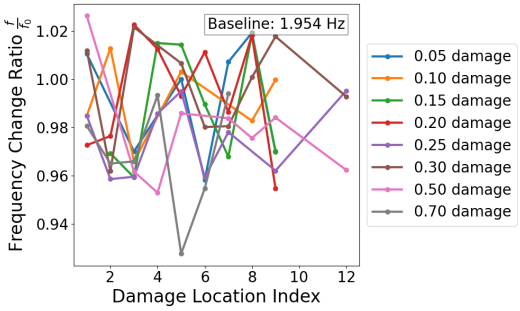
(a) 1st Lateral Symmetric Modes - XSym



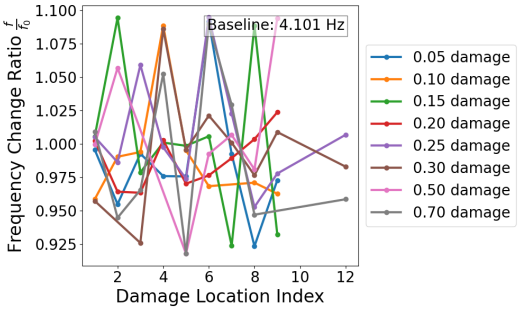
(b) 1st Lateral Symmetric Modes - DSym



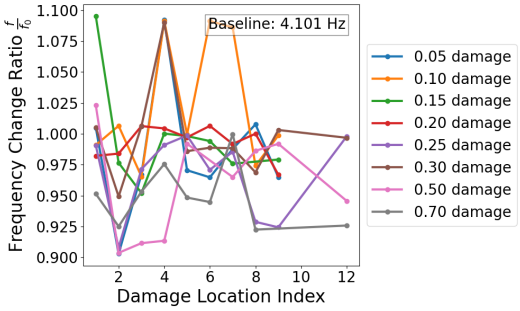
(c) 2nd Lateral Symmetric Modes - XSym



(d) 2nd Lateral Symmetric Modes - DSym

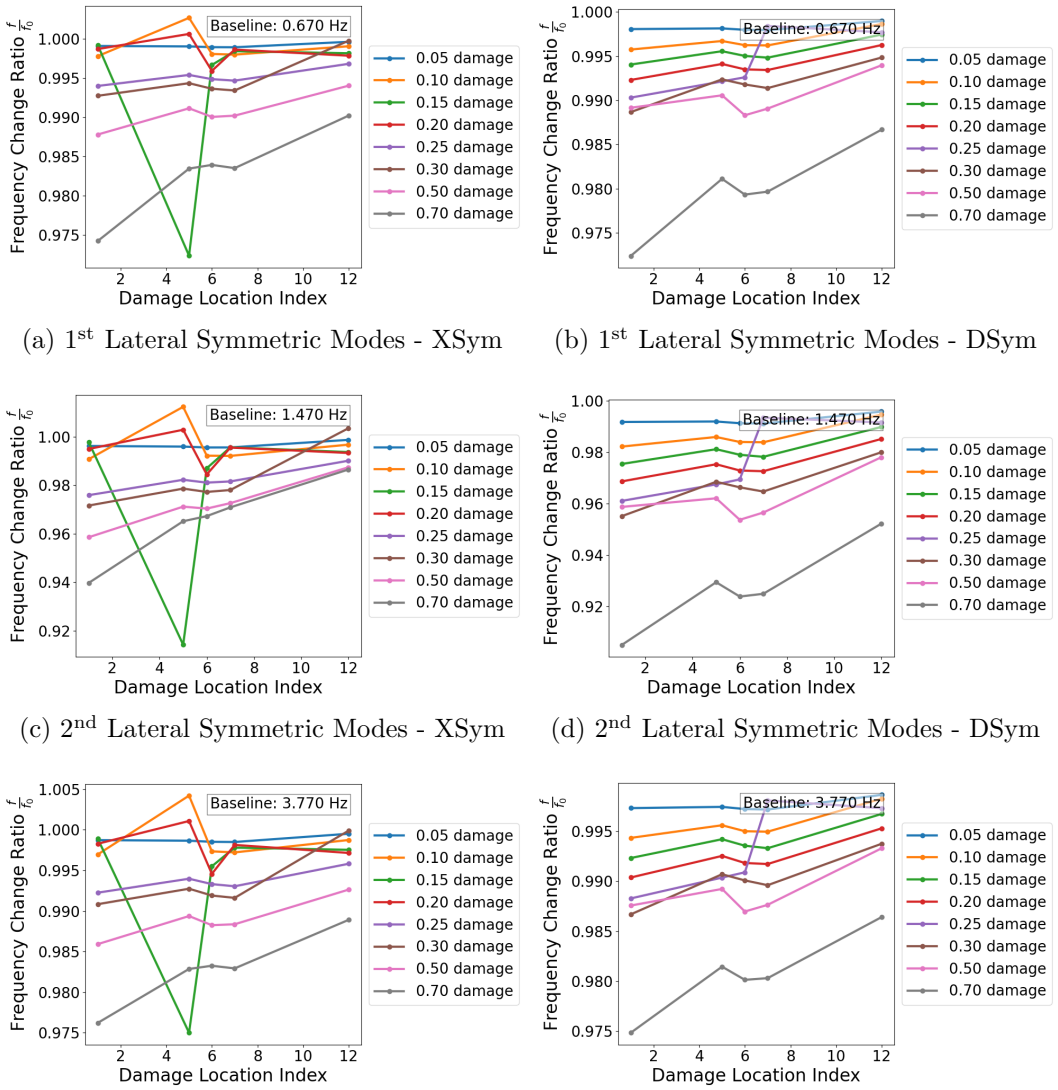


(e) 3rd Lateral Symmetric Modes - XSym



(f) 3rd Lateral Symmetric Modes - DSym

Figure I.11: FREQ11 Doub Symmetrical Lateral Modes of Vibration



(a) 1st Lateral Symmetric Modes - XSym

(b) 1st Lateral Symmetric Modes - DSym

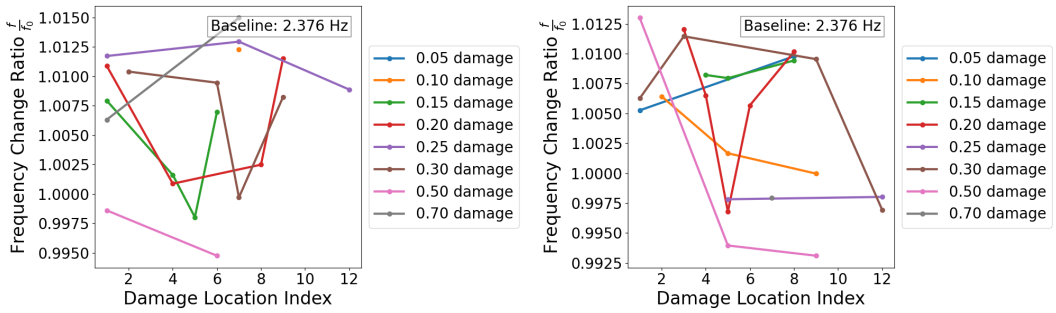
(c) 2nd Lateral Symmetric Modes - XSym

(d) 2nd Lateral Symmetric Modes - DSym

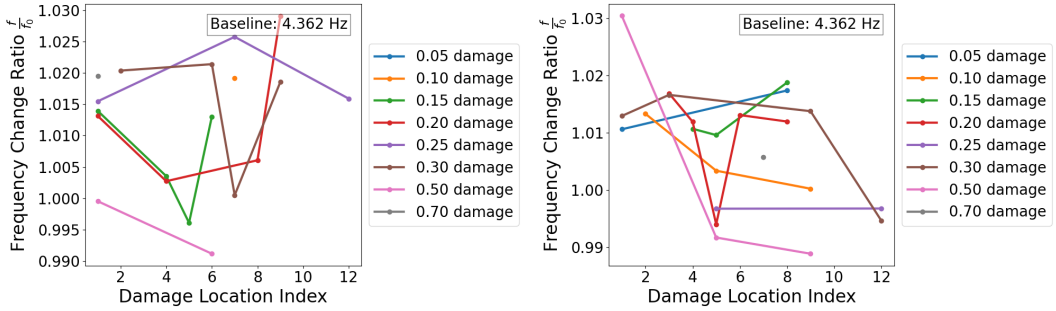
(e) 3rd Lateral Symmetric Modes - XSym

(f) 3rd Lateral Symmetric Modes - DSym

Figure I.12: FREQ34 Doub Symmetrical Lateral Modes of Vibration

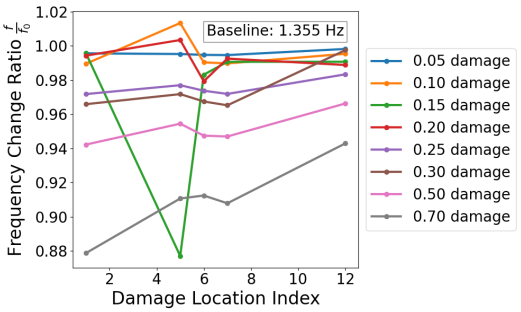


(a) 1st Lateral Asymmetric Modes - XSym (b) 1st Lateral Asymmetric Modes - DSym

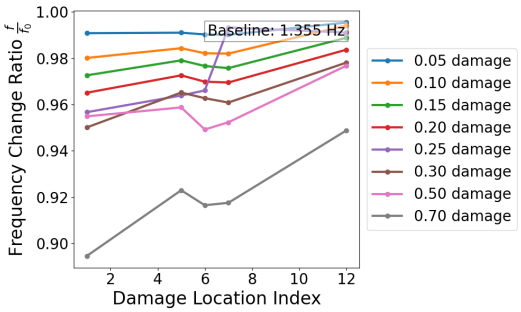


(c) 2nd Lateral Asymmetric Modes - XSym (d) 2nd Lateral Asymmetric Modes - DSym

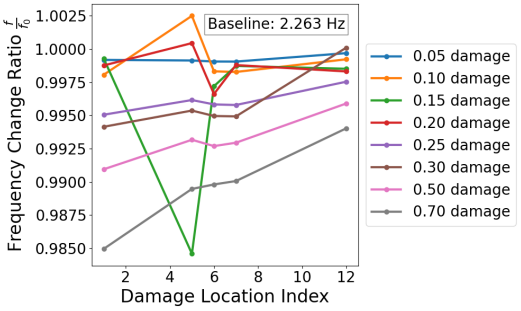
Figure I.13: FREQ11 Doub Asymmetrical Lateral Modes of Vibration



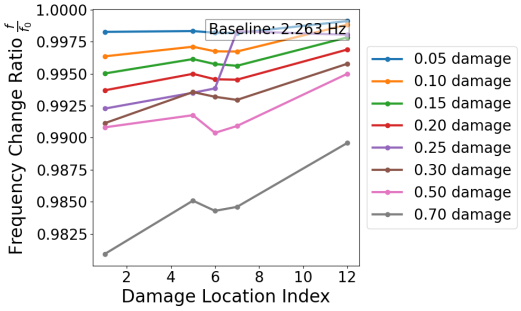
(a) 1st Lateral Asymmetric Modes - XSym



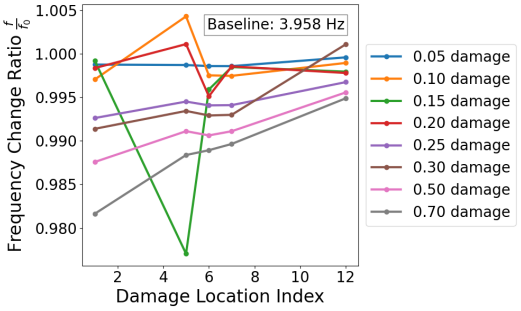
(b) 1st Lateral Asymmetric Modes - DSym



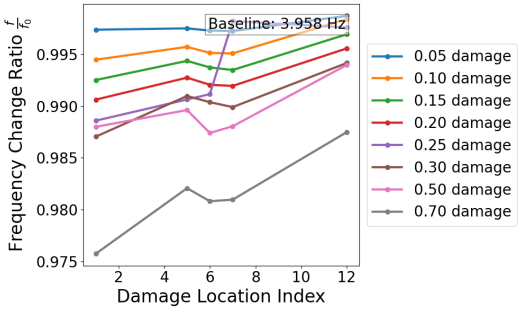
(c) 2nd Lateral Asymmetric Modes - XSym



(d) 2nd Lateral Asymmetric Modes - DSym

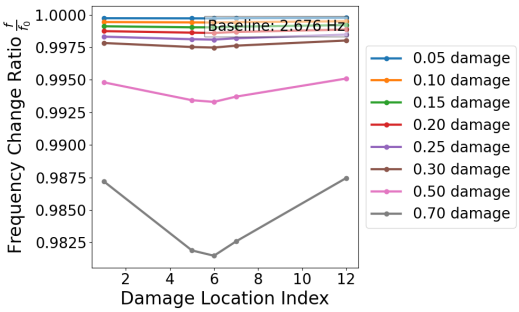


(e) 3rd Lateral Asymmetric Modes - XSym

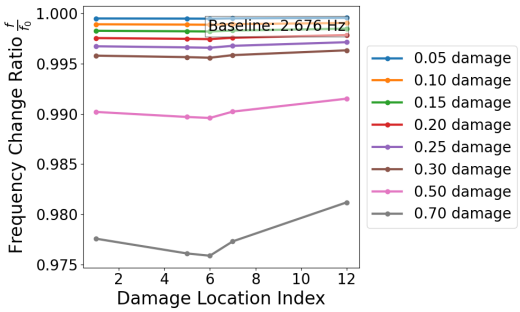


(f) 3rd Lateral Asymmetric Modes - DSym

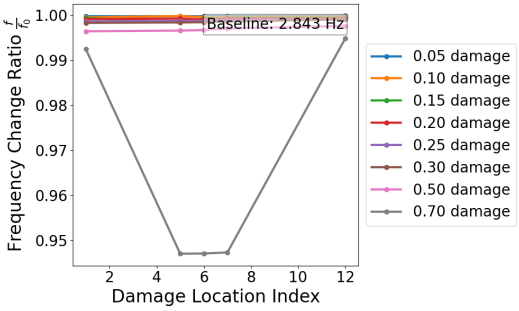
Figure I.14: FREQ34 Doub Asymmetrical Lateral Modes of Vibration



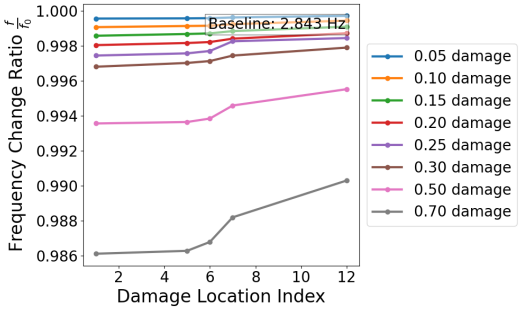
(a) 1st Torsional Symmetric Modes - XSym



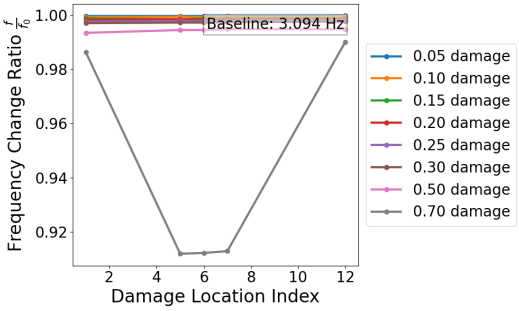
(b) 1st Torsional Symmetric Modes - DSym



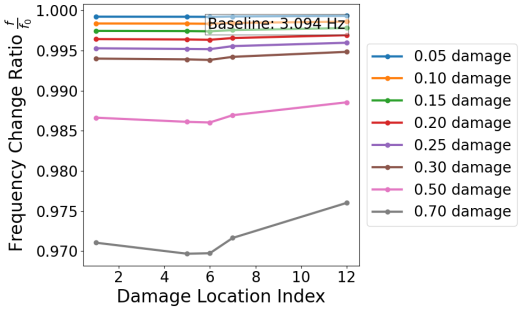
(c) 2nd Torsional Symmetric Modes - XSym



(d) 2nd Torsional Symmetric Modes - DSym

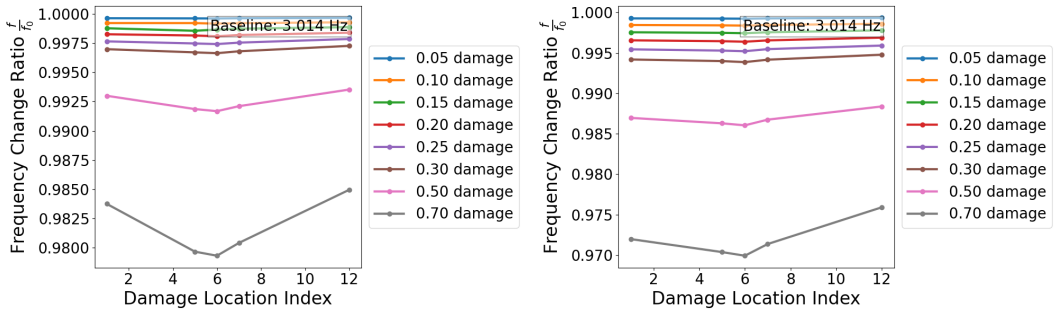


(e) 3rd Torsional Symmetric Modes - XSym

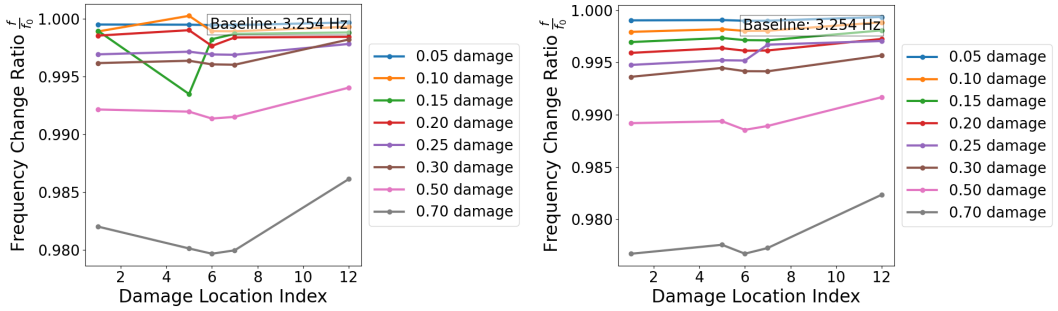


(f) 3rd Torsional Symmetric Modes - DSym

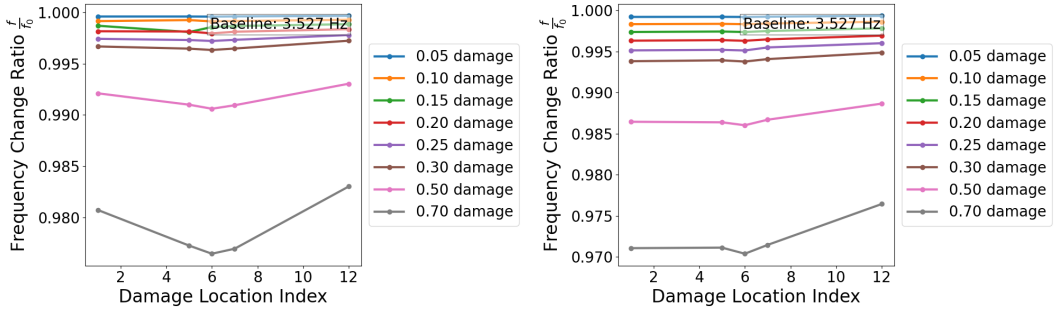
Figure I.15: FREQ34 Doub Symmetrical Torsional Modes of Vibration



(a) 1st Torsional Asymmetric Modes - XSym (b) 1st Torsional Asymmetric Modes - DSym



(c) 2nd Torsional Asymmetric Modes - XSym (d) 2nd Torsional Asymmetric Modes - DSym



(e) 3rd Torsional Asymmetric Modes - XSym (f) 3rd Torsional Asymmetric Modes - DSym

Figure I.16: FREQ34 Doub Asymmetrical Torsional Modes of Vibration