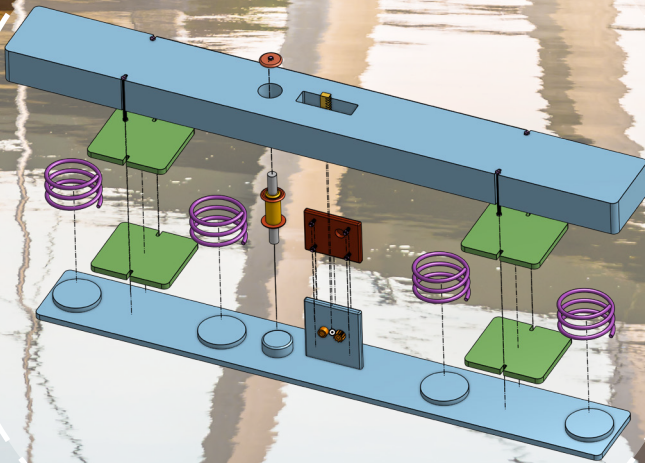


Geometric Optimization of Pedestrian Bridges Reducing Human-Induced Vibrations and External Damping Reliance

L. van Kouwen

Delft University of Technology



Geometric Optimization of Pedestrian Bridges

Reducing Human-Induced Vibrations and External Damping Reliance

by

L. van Kouwen

5425123

To obtain the degree of

Master of Science in Civil Engineering

at the Delft University of Technology,

to be defended publicly on February 11th, 2025 at 13:00.

Committee:	Dr ir. T. Tankova	TU Delft - Steel & Composite Structures	Chair
	Dr. ir. K. van Dalen	TU Delft - Dynamics of Structures	Supervisor
	Ir. P. Atzampou	TU Delft - Dynamics of Structures	Supervisor
	Ir. J. Moen	ABT - Adviseurs in Bouwtechniek	Company supervisor

Faculty: Civil Engineering and Geosciences

Company: ABT - Adviseurs in Bouwtechniek

Duration: October 2nd, 2023 - February 11th, 2025

Cover credits | Fietsbrug Voldijk, Dura Vermeer / Jorrit Lousberg Fotografie
Style | TU Delft Report Style

Preface

This thesis is written as part of the Master of Science program in Civil Engineering at Delft University of Technology, within the Structural Engineering track, specializing in Steel, Timber & Composite Structures. The research was conducted in collaboration with ABT - Adviseurs in Bouwtechniek, an engineering firm specializing in structural, building, and geotechnical engineering.

The study is intended for engineers and researchers with an interest in human-induced vibrations, pedestrian bridge design, structural optimization, and external damping systems namely tuned mass dampers (TMDs). The thesis is structured into three parts: Part I provides an extensive literature review on human-induced vibrations and their implications for structural design. Part II focuses on the design and assessment of a case study involving a pedestrian bridge. Finally, Part III employs multi-objective particle swarm optimization (MOPSO) to geometrically optimize the case study, aiming to reduce human-induced vibrations and dependence on external damping.

This research contributes to advancing knowledge in assessing human-induced vibrations, as outlined in the literature. It presents a detailed case study of a self-anchored suspension bridge designed for pedestrians and cyclists. Emphasis is placed on the theoretical foundation of self-anchored suspension bridges, including the form-finding process to establish the initial stress state of the main cable system. An extensive description of the finite element (FE) model used for analysis is provided, followed by insights into the structural optimization process. The thesis highlights the principles of particle swarm optimization (PSO), a stochastic, population-based algorithm leveraging swarm intelligence, and its application in optimizing the bridge geometry to mitigate vibrations and reduce reliance on external damping.

I am deeply grateful to my family and friends for their support throughout this journey. I would like to express gratitude to my parents for their encouragement and for providing me with the opportunity to pursue my studies. To my friends, thank you for your constant motivation and support.

A special note of gratitude goes to my supervisor, Joris Moen, for his guidance and interest in this project from the beginning. I would also like to thank my supervisors from the TU Delft: Trayana Tankova, Karel van Dalen, and Panagiota Atzampou for their patience, valuable feedback, and continuous support throughout the thesis.

Working on this thesis has been an immensely rewarding experience. I hope this work inspires and sparks curiosity in you as a reader!

*L. van Kouwen
Delft, February 2025*

Abstract

Advancements in structural engineering increasingly lead to more slender and architecturally challenging footbridge designs, characterized by reduced height-to-span ratios and lower self-weight. These designs, often serving as landmarks with aesthetic and functional purposes, are becoming more susceptible to human-induced vibrations. The increased live-to-dead load ratio and reduced eigenfrequencies increase the risk of resonance when pedestrian step frequencies align with the structure's natural frequencies. This resonance amplifies deck accelerations, compromising pedestrian comfort and freedom of movement. Synchronization effects are at risk of further intensifying amplification and obstructing movement. No studies have demonstrated that human-induced vibrations cause structural failure in the ultimate limit state (ULS); rather, they primarily affect user comfort in the serviceability limit state (SLS).

Over the past three decades, substantial research has focused on understanding and mitigating human-induced vibrations in footbridges. The temporary closures of iconic structures such as the Passerelle Solférino in Paris (1999) and the London Millennium Bridge (2000) have potentially accelerated findings and highlighted the significant challenges posed by pedestrian-structure interaction. In particular, a focus is drawn to the lateral lock-in phenomenon, which resulted in excessive lateral vibrations and discomfort. Lateral lock-in and other human-induced incidents resulted in extensive testing, leading to the development of new guidelines and advancing the study of lightweight footbridges across Europe through both in-situ testing and numerical simulations. However, the literature reveals a considerable variation in assessment methods and verification techniques, which complicates the accurate evaluation of a footbridge's dynamic behaviour by structural engineers. Nevertheless, a consensus is reached in the literature that design situations must be carefully considered in every footbridge design. Pedestrian comfort and dynamic response are crucial design factors, requiring a thorough understanding of expected traffic patterns and the structure's dynamic behaviour. Design situations encompass a range of conditions, such as daily pedestrian use or special events, to establish realistic performance limits under ranging circumstances. Studies show that higher pedestrian density leads to reduced walking speeds and restricted movement, which in turn influences the dynamic loads on the structure. Comfort is assessed through acceleration measurements during loading, with predefined ranges to categorize acceptable performance levels. These evaluations stress the importance of a comprehensive analysis of dynamic effects, rather than relying on a single limit criterion.

To mitigate the issues observed in the aforementioned bridge designs, external control devices are applied to offer additional damping and reduce vibrations to acceptable levels. These control devices are applied after footbridge construction, enabling thorough testing of the bridge to determine the structure's dynamic properties and ensuring the damping system is optimized and properly tuned. External damping has various forms applied in civil engineering structures. Most notably there are three categories to be distinguished, namely: tuned mass/liquid, viscoelastic and viscous fluid dampers. Tuned mass dampers (TMDs) are most commonly used due to their ease of application, allowing for effective control of vibrations post-installation. By tuning the TMD's eigenfrequency to match the primary structure's critical natural frequency, energy dissipation is achieved through the mass of the damper and its motion relative to the structure to which it is attached. TMD design is highly effective in controlling the target frequency. However, it should be noted that this localized damping primarily addresses the response of a single frequency, rather than the total response of the structure. If eigenfrequencies are closely spaced, a shift in frequency could lead to a new resonant response in the overall structure, as damping the initially critical mode may consequently amplify nearby modes.

These imposed challenges raise the question if reducing excessive human-induced vibrations within footbridge design can be achieved through other means. A promising method regards the geometric modification of the structural design, providing a change of dynamic characteristics and reducing resonance effects. Modern-day advancements enable engineers to perform more complex problem solving, namely through computational power by utilising optimisation techniques which require many iterations. An optimisation is characterised by its objective function, constraints, design variables and requirements it should satisfy. Evolutionary algorithms, such as genetic behaviour from groups observed by animals in nature provide effective results for optimisation.

To provide context to such an optimisation algorithm, a case study is presented, showcasing how the concept can be utilised. The optimization primarily targets reducing the structure's mass as an objective, improving cost and sustainability while influencing dynamic performance. Minimizing accelerations is likewise pursued to evaluate the extent of reduction possible and identify the most influential geometric parameters. The structure must meet ultimate limit state requirements in the optimized design to ensure feasibility. Data from the original design, including FEM models, analysis reports, and TMD specifications, informs the optimization process. A parametric model is developed to support geometric optimization. Key design variables, objectives and constraints are carefully selected to maximize the effectiveness of the optimization and achieve a design that either mitigates or eliminates the need for external damping devices.

When footbridge design deviates significantly from conventional girder bridge design will the effectiveness of assessment methods drop, requiring more extensive analysis to address dynamic behaviour. The presented case study shows the closest adherence to measured results via direct time integration, being most costly in time whilst requiring a substantial level of engineering judgment. Furthermore, does the correct assessment of damping in footbridge design play a major role, showing agreement with the proposed mean damping values addressed in the literature.

Optimisation to exclude the need for external damping devices through evolutionary algorithms by conducting a geometric parameter study is a feasible approach. However, it requires a deep understanding of the structural behaviour of footbridges and a robust parametric model capable of performing both static and dynamic analyses to account for geometric changes. In the case study, significant improvements were achieved through this optimization process, resulting in a new design that eliminates the need for TMDs.

Contents

Preface	i
Abstract	ii
Nomenclature	xi
1 Introduction	1
1.1 Research context	1
1.2 Problem statement	1
1.3 Research methodology	2
1.4 Research questions	2
1.5 Thesis outline	2
I Assessment of pedestrian-structure interaction	3
2 Literature review	4
2.1 Maximum acceleration	4
2.2 Lateral lock-in effect	7
2.3 Design situations	9
2.3.1 Traffic classes	10
2.3.2 Comfort classes	11
2.4 Design steps	12
2.4.1 Evaluation of natural frequencies	12
2.4.2 Critical range of natural frequencies	13
2.4.3 Assessment of design situations	14
2.4.4 Assessment of structural damping	15
2.5 Dynamic load models	20
2.5.1 Step force model	22
2.5.2 Equivalent number of pedestrians	23
2.5.3 Force component	26
2.5.4 Reduction coefficient	27
2.5.5 Moving harmonic point load model	28
2.5.6 Harmonic load model for pedestrian streams	28
2.5.7 Harmonic load model for intentional excitation	29
2.6 Assessment methods	31
2.6.1 SDOF method	31
2.6.2 Response spectra method	32
2.6.3 Four Footfall Harmonics	33
2.6.4 Direct time integration method	36
2.6.4.1 Direct Integration Method	37
2.6.4.2 Numerical damping	39
2.6.4.3 Newmark direct integration method	39
2.6.4.4 Non-linear dynamics and direct integration	40
2.7 External damping	42
2.7.1 Den Hartog	42
2.7.2 Damping of the primary structure	44
2.7.3 Application to footbridges	46
II Voldijk bridge - design & assessment	47
3 Case study design	48
3.1 Introduction	48
3.2 Suspension bridges	49
3.3 Parallel structural system	50
3.3.1 Earth-anchored theory	50
3.3.2 Self-anchored theory	53
3.3.3 Procedure for self-anchored suspension bridges	56

3.4	Structural design	58
3.5	Finite element model	61
3.5.1	Substructure	61
3.5.2	Superstructure	62
3.6	Eigenfrequencies	64
3.6.1	Eigenvalue analysis in FEM	65
3.6.2	Sledgehammer tests	68
3.6.3	Comparison of eigenfrequencies	69
3.7	Structural damping	69
3.8	Design situations	70
3.9	External damping	70
3.9.1	Evaluation of response	70
3.9.2	TMD design	74
4	Case study assessment	78
4.1	Four Footfall Harmonics	79
4.2	Pedestrian crowds	81
4.2.1	SDOF	81
4.2.2	Direct time	82
4.2.3	Sledgehammer tests	84
4.2.4	Comparison	85
4.3	Moving loads	86
4.3.1	SDOF	86
4.3.2	Direct time	87
4.3.3	Comparison	87
III	Geometric optimisation to reduce human-induced vibrations and external damping reliance	88
5	Optimisation	89
5.1	Methodology	89
5.2	Design variables	90
5.2.1	Pylon height	90
5.2.2	Number of hangers	91
5.2.3	Main girder	92
5.3	Objectives and constraints	92
5.4	Particle Swarm Optimisation (PSO)	96
5.5	Optimisation procedure case study	99
6	Results	101
6.1	Effect design variables	101
6.1.1	Pylon height	101
6.1.2	Cross-section dimensions	102
6.1.3	Number of hangers	102
6.2	Multi-Objective Particle Swarm Optimisation (MOPSO)	103
6.3	Comparison of designs	104
7	Discussion	107
7.1	Interpretation	107
7.2	Implications	108
7.3	Limitations	108
8	Conclusion and recommendations	112
8.1	Conclusion	112
8.2	Recommendations	114
	Bibliography	115
A	Appendix A - Python source code	118
A.1	Parametric script of the case study - Step 2	118
A.2	ULS verification - Step 5	158
A.3	Mode shape clustering - Step 7	160
A.4	MOPSO algorithm - Steps 1, 9 & 10	164
B	Appendix B - Load cases & combinations	170
B.1	Self-weight	170

B.2	Additional dead loads	170
B.3	Pedestrian loads	171
B.4	Wind loads	172
B.5	Load combinations	173
C	Appendix C - SOFiSTiK output	175
C.1	Original design - ULS & direct time integration	175
C.2	Optimised design - ULS & direct time integration	195

List of Figures

2.1	ISO baseline curve for human perception and vertical peak acceleration limits for various civil structures	5
2.2	Vertical RMS acceleration limits found in literature	5
2.3	Limit values for vertical acceleration enlisted in standards and guidelines	5
2.4	Comfort values for vertical acceleration enlisted in literature [6]	5
2.5	Limit values for horizontal acceleration enlisted in standards and guidelines[6]	6
2.6	Comfort values for horizontal acceleration enlisted in literature [6]	6
2.7	Design procedure enlisted in EUR23984	7
2.8	Design procedure enlisted in Sétra guideline	7
2.9	Schematic description of synchronous walking	8
2.10	Acceleration (m/s^2) [pink] and applied force (N) [blue] with ten random pedestrians on the test rig [4]	8
2.11	Design situation approach	9
2.12	Traffic classes in JRC-document	10
2.13	Design steps according to EUR23984 [5]	12
2.14	Design steps according to "European design guide for footbridge vibration" publication [22]	12
2.15	Results of pedestrian interviews concerning perception of vibration for the Kochenhofsteg in Stuttgart	15
2.16	Results of pedestrian interviews concerning perception of vibration for the Wachtelsteg in Pforzheim	15
2.17	Variation of modal damping ratios with natural frequency: (a) mass-proportional damping and stiffness-proportional damping; (b) Rayleigh damping [28]	17
2.18	Three cases of Rayleigh damping to be defined by the use of two distinct damping values ξ_1 and ξ_2 [29]	18
2.19	[28]	19
2.20	Example of subsystem assembly to account for soil-structure-interaction of different modal damping ratios [28]	20
2.21	Typical vertical force patterns for different types of human activities from a single footstep [30]	21
2.22	Dependence of stride length, contact time and DLF for different step frequencies [30] . .	21
2.23	Periodic walking time histories in vertical, lateral and longitudinal directions. [31] . . .	22
2.24	Equivalent number of pedestrians derived from Monte Carlo simulations for stochastic crowd loading	25
2.25	Dynamic load factor of footfall forces observed in studies [11]	26
2.26	Vertical and longitudinal reduction coefficient ψ for walking according to guidelines and codes	27
2.27	Lateral reduction coefficient ψ for walking according to guidelines and codes	27
2.28	Vertical reduction coefficient ψ for jogging according to guidelines and codes	28
2.29	Stationary loading specifying its direction according to the given torsional mode shape [4] .	29
2.30	Constants to estimate generalised (modal) mass [5]	33
2.31	Peak velocities of the first four harmonics of footfall forces for ranging frequencies as per found in literature [11]	33
2.32	Building vibration z-axis base curve for acceleration (foot-to-head vibration direction) [12] .	35
2.33	Response factors for various mode shapes and eigenvalues according to Four Footfall Harmonics [11]	36
2.34	Simplified FEM modelling of a cable stayed footbridge [6]	37
2.35	Percentage of period decay for various $\frac{\Delta t}{T}$ ratios [34]	38
2.36	Percentage of amplitude decay for various $\frac{\Delta t}{T}$ ratios [34]	38
2.37	Effective filtering of higher frequencies for a displacement response after one-hundred steps, Wilson $\theta = 1.4$ [34]	39
2.38	Newton-Raphson iteration scheme for Newmark Beta method [35]	42
2.39	Two-mass system with undamped primary mass and TMD [41]	43
2.40	Fixed points P and Q for various damping ratio's ξ_d whilst looking at transfer function H_u of the primary mass at different forcing frequencies Ω for $\mu = 0.01$ and $\rho_{tune} = 1$ [41]	43
2.41	Amplification factors H_u for optimised tuning ρ_{opt} and damping ratio $\xi_{d,opt}$ [41]	44

2.42	SDOF representation of primary mass and TMD for a fully damped system as proposed by Ghosh and Basu [41]	45
2.43	Schematic overview TMD with its main components	46
3.1	Voldijk bridge Tilburg - Graphical representation of the case study [43]	48
3.2	Types of suspension bridges and their structural schematics	49
3.3	Components of a self-anchored suspension bridge	50
3.4	Structural system of a one span earth-anchored suspension bridge	50
3.5	Bending moments for a main girder segment dx subjected to live loads w_L only	52
3.6	Structural system of a three span self-anchored suspension bridge including fabrication camber [44]	53
3.7	Bending moments for a main girder segment dx subjected to live loads w_L only	55
3.8	Graphic representation of steps 2 and 3 of the methodology, in which the horizontal forces of the main cables are parallel to those of the main girder, allowing for translation of the cable roller supports in longitudinal direction	57
3.9	Case study representing the cross section of the deck, troughs elements, coupler and deck element connection [43]	58
3.10	Prefabricated reinforced concrete pylons of the case study [43]	59
3.11	Support conditions for the abutments located on the South and North side as well and bevelled corbel connection at the pylons of the case study [43]	59
3.12	Self-anchored suspension bridge concept and force transfer in reference project [43]	60
3.13	Bending moment line of original and form-finding design for permanent loads (self-weight and additional dead load) of the case study Voldijk bridge	60
3.14	Wireframe model of the pylon foundation modelled in FEM	62
3.15	Wireframe model of the abutment foundation modelled in FEM	62
3.16	Connection detail of the superstructure at the abutment in FEM	63
3.17	Connection detail of the superstructure at the pylons in FEM	63
3.18	Meshing of the FE model for a mesh size of 600 mm Mode shapes and eigenfrequencies obtained from eigenvalue	64
3.19	analysis in Oasys GSA regarding the original design for permanent loads Mode shapes and eigenfrequencies obtained from eigenvalue	66
3.20	analysis in SOFiSTiK regarding the original design for per- manent loads	67
3.21	Excitation point of sledgehammer and position accelerometers for performed sledgehammer tests	68
3.22	Mode shapes obtained from sledgehammer tests	68
3.23	Measurement points - Transient analysis Oasys GSA Obtained vertical accelerations in point A1 and A2 via transient	71
3.24	analysis via Oasys GSA, depicting the reduction coefficient still to be applied for dynamic loading based on walking frequency Obtained torsional accelerations in point B1 and B2 from transient	72
3.25	analysis via Oasys GSA, depicting the reduction coefficient still to be applied for dynamic loading based on walking frequency	73
3.26	Positioning of four TMDs according to final design	74
3.27	Modelling of TMDs in FEM	75
3.28	Eigenfrequencies obtained after instalment TMDs	75
3.29	Transient response of the TMD design for ranging traffic classes for a target frequency of $f_{vert.1=1.610} Hz$	76
4.1	Representation of 4FFH analysis of the case study	79
4.2	Full spectrum of the 4FFH analysis observed at point A1 of the bridge deck, assuming $\xi_{s,min} = 0.2\%$	80
4.3	Combined response of the 4FFH analysis observed at point A1 of the bridge deck, assuming $\xi_{s,min} = 0.2\%$	80
4.4	Time step sensitivity analysis of the first vertical eigenmode, assuming $\xi_{s,min} = 0.2\%$	82
4.5	Transient response of the first vertical mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$	83
4.6	Transient response of the first torsional* mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$	83
4.7	Transient response of the second vertical mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$	83
4.8	Transient response of the first torsional mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$	84
4.9	Transient response of pedestrians and joggers for the first vertical mode according to TC3, assuming $\xi_{s,min} = 0.2\%$	87
5.1	Optimisation methodology [52]	89
5.2	Schematic representation of pylon height influence of the case study	91

5.3	Schematic representation of the influence of hangers of the case study.	91
5.4	Hanger configurations considered for the case study.	92
5.5	Main girder design of the original case study.	92
5.6	Three designs, A, B, and C, are plotted against two objectives, f_1 and f_2 . The region in the shaded rectangle highlights points that are dominated by design A [52]	93
5.7	Pareto front of a MO optimisation, Pareto optimal solutions are denoted in red [52]	93
5.8	Weighted sum method as a function for objectives f_1 and f_2 , with results for the convex parts of the function, showing inconsistent spacing along the Pareto front [52]	94
5.9	Inertia, memory and social components of the velocity vector in PSO[52].	96
5.10	Single objective PSO of the case study presenting optimisation for two design variables at iteration 0, 50, 75 and 100 respectively	98
5.11	Optimisation procedure of the case study	99
6.1	Maximum acceleration for the effect of varying pylon height plotted for dominant mode shapes within the critical range	101
6.2	Maximum acceleration for the effect of varying cross-sections of the main girder plotted for dominant mode shapes within the critical range	102
6.3	Influence hanger configuration regarding the maximum deck accelerations of the case study for each critical mode shape	102
6.4	Comparison of different iterations for the MOPSO of the case study showing the progression of solutions. A black point represents a particle containing a design solution whereas a red point represents a Pareto dominant solution.	103
6.5	Transient response of the first vertical mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$	105
6.6	Transient response of the first torsional mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$	105
6.7	Transient response of the second vertical mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$	106
6.8	Transient response of the first torsional* mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$	106
7.1	DBSCAN algorithm identifying clustering of the mode shapes, where the second torsional* mode shape shows incorrect clustering	110
7.2	Stress response of the main girder at the support and midspan for intentional excitation according to prEN1991-2-2021 subjected to five vandals located at midspan with DLF=1.6111	
B.1	Load cases considered for pedestrian loads	171
B.2	Load cases considered for wind loads	172
B.3	ψ factors and load combinations according to NEN-EN-1990-2019 [32]	174

List of Tables

2.1	Design situations - EUR23984	10
2.2	Design situations - Sétra guideline	10
2.3	Traffic classes in Sétra guideline	11
2.4	Comfort levels in EUR23984 and Sétra guideline	11
2.5	Pedestrian activity according to Bachmann [26]	14
2.6	Critical frequency ranges as per specified in guidelines and codes	14
2.7	Minimum and average damping ratio's per construction type	15
2.8	Increased damping ratio's per construction type	15
2.9	Fourier coefficients, phase angles as proposed by studies found in literature [5]	23
2.10	Traffic classes and group sizes for harmonic load models as per prEN1991-2.	23
2.11	Characteristics of the P_j component specified in codes and guidelines	26
2.12	Analysis methods described in literature to determine maximum bridge deck acceleration due to pedestrian-structure interaction of footbridges	31
2.13	Constants for vertical accelerations [5]	32
2.14	Constants for lateral accelerations [5]	32
2.15	Proposed dynamic load factors for the first four harmonics	34
2.16	Performance targets for bridges, ramps and walkways regarding the response factor [11]	35
2.17	Comparison of direct integration methods and their stability criterion [35]	38
3.1	Eigenvalue analysis in Oasys GSA of the original design for permanent loads and permanent loads including the static load of pedestrians for traffic class 3 (TC3).	65
3.2	Eigenvalue analysis in SOFiSTiK of the original design for permanent loads and permanent loads including the static load of pedestrians for traffic class 3 (TC3).	66
3.3	Comparison of eigenvalue analysis for permanent loads and permanent loads including traffic class 3 (TC3) obtained in Oasys GSA and SOFiSTiK	67
3.4	Eigenfrequencies derived from sledgehammer tests	68
3.5	Comparison of eigenfrequencies obtained from eigenvalue analysis and hammer impact tests for permanent loads	69
3.6	Difference between measured and FEM-derived eigenfrequencies	69
3.7	Design situations to be considered for the case study Eigenfrequencies obtained in Oasys GSA, subjected	70
3.8	to transient analysis for tuned mass damper design of the case study	70
3.9	Verification TMD design case study excluding reduction factor ψ	77
4.1	Critical eigenfrequencies obtained in SOFiSTiK used for assessment	78
4.2	Assessment methods utilised for the case study	78
4.3	Maximum deck accelerations for the critical vertical modes, assuming $\xi_{s,min} = 0.2\%$	81
4.4	Results of the transient analysis of both FE models due to dynamic loads of pedestrian streams for all specified traffic classes, assuming $\xi_{s,min} = 0.2\%$	84
4.5	Acceleration response derived from sledgehammer tests	84
4.6	Comparison SDOF and direct time assessment methods, assuming $\xi_{s,min} = 0.2\%$	85
4.7	Comparison direct time method and measured results sledgehammer testing	85
4.8	Steady-state response of moving loads for the first vertical mode, assuming $\xi_{s,min} = 0.2\%$	86
4.9	Comparison moving load analysis SDOF and direct time method plus equivalent crowd-loading analysis for TC3, assuming $\xi_{s,min} = 0.2\%$	87
5.1	Design situation three considered for assessment of the case study, see 3.8 - Design situations.	90
6.1	Pareto front obtained from MOPSO of the case study	104
6.2	Comparison of optimal solution to original design case study	104
6.3	Critical eigenfrequencies original design and optimised solution for design scenario 3	105
7.1	Design situation three considered for assessment of the case study, see 3.8 - Design situations.	109

Nomenclature

Abbreviations

Abbreviation	Definition
4FFH	Four FootFall Harmonics
ALS	Accidental Limit State
API	Application Programming Interface
BML	Bending Moment Line
BS	British Standard
CC	Comfort Class
DLF	Dynamic Load Factor
DIAMOND	Graphical interface toolbox for comparative modal analysis
EN	European Norm
EOM	Equation Of Motion
EUR23984	European Union Regulation 23984
FE	Finite Element
FEM	Finite Element Method
FFT	Fast Fourier Transform
ISO	International Organization for Standardization
JRC	Joint Research Centre
MOPSO	Multi-Objective Particle Swarm Optimisation
MO	Multi-Objective
Oasys GSA	Structural Analysis and Design Software
PSO	Particle Swarm Optimisation
Python	Programming language
Rhino Grasshopper	Visual programming language and plugin for Rhinoceros
RMS	Root Mean Square
SDOF	Single Degree Of Freedom
S��tra	Service d'��tudes Techniques des Routes et Autoroutes
SLS	Serviceability Limit State
SOFiSTiK	Structural Analysis and Design Software
SOPSO	Single-Objective Particle Swarm Optimisation
SRSS	Square Root of the Sum of Squares
SYNPEX	SYNchronous Pedestrian EXcitation
TC	Traffic Class
TCUD	Target Configuration Under Deadload
TMD	Tuned Mass Damper
TM	Torsional Mode
ULM	Unstrained Length Method
ULS	Ultimate Limit State
VM	Vertical Mode
P - Δ effect	Additional internal forces that develop due to the displacement of a structure under a normal load

Latin symbols

Symbol	Definition	Unit
a	Acceleration value	$[m/s^2]$
$a_{h,lock-in}$	Acceleration value for which lateral lock-in is observed in the literature	$[m/s^2]$
$a_{i,lim}$	Acceleration limit for the i th direction	$[m/s^2]$
$a_{i,max}$	Maximum deck acceleration for any part of the bridge in vertical or horizontal direction	$[m/s^2]$
d	Pedestrian density	$[P/m^2]$
E_c, A_c	Axial stiffness of the cable	$[N/mm^2]$

Symbol	Definition	Unit
E_g, I_g	Bending stiffness of the girder	[N/mm ²]
$F_{m,i}(t, v)$	Moving harmonic point load due to walking or jogging in the ith direction	[N]
$F_{ped,i}(t, v)$	Dynamic loading of a single moving pedestrian in the ith direction	[kN]
$F_{s,i}(t)$	Harmonic area force for pedestrian streams in the ith direction	[N/m]
$F_{vandal}(t)$	Harmonic point load for vandals at critical location x of the bridge	[N]
$f_{b,n}$	Natural frequency of a simply supported beam for bending at n modes	[Hz]
$f_{d,opt}$	Optimal tuning frequency of the damper	[Hz]
f_{max}	Maximum obtained frequency in the critical range	[Hz]
f_n	Eigenfrequency of mode n	[Hz]
$f_{n,lat}$	Natural frequency for lateral vibrations	[Hz]
f_o	Fundamental frequency	[Hz]
$f_{p,m,n}$	Natural frequency of a simply supported plate for bending at m and n modes	[Hz]
f_s	Step frequency of pedestrians	[Hz]
f_{syn}	Synchronized step frequency of pedestrians	[Hz]
H_w	Horizontal force in the main cable system subjected to permanent loads	[kN]
h_p	Height of the pylons	[m]
h_{gird}	Height of the web of the main girders	[mm]
k_{tot}	Total vertical translation stiffness of the foundation pile	[N/mm]
L	Length of the bridge deck	[m]
m_d	Mass of the damper	[kg]
m_n	Modal mass	[kg]
m^*	Modal mass	[kg]
m''	Modified modal mass accounting for the additional mass of pedestrians	[kg]
N_L	Number of people triggering the lateral lock-in effect	[-]
n_{hang}	Number of hangers	[-]
n'	Equivalent number of pedestrians	[-]
p^*	Modal load	[N]
q_c	Cable loads of the cable system	[kN/m]
q_g	Girder and deck loads of the bridge deck	[kN/m]
R	Response factor	[-]
S	Area of the loaded surface (bridge deck)	[m ²]
T_h	Distributed tension in the hangers of the suspension bridge	[N]
t_{pass}	Passing time for pedestrians to cross the bridge	[s]
v	Velocity	[m/s]
$v_{cr,n}$	Critical velocity of the bridge for mode n	[m/s]
v_{jog}	Walking velocity of joggers	[m/s]
v_{ped}	Walking velocity of pedestrians	[m/s]
w_D	Permanent loads in the parallel system of a suspension bridge	[kN/m]
w_L	Additional loads in the parallel system of a suspension bridge	[kN/m]
w_{gird}	Width of the flange of the main girders	[mm]

Greek symbols

Symbol	Definition	Unit
α	Inertia term of the velocity vector	[-]
α_k	Proportionality factor for the correct scaling to the modal load for mode k	[-]
$\alpha_{i,j}$	Fourier coefficient of the ith harmonic for the jth type of loading	[-]
β	Memory term of the velocity vector	[-]
χ_n	Imaginary part of the complex-valued eigenvector of mode n	[m/rad]
δ	Logarithmic decrement	[-]
Δt	Time step used for numerical integration	[s]
γ	Social term of the velocity vector	[-]
λ_n	Conjugate pair of the complex-valued eigenvalue at mode n	[rad/s]
μ	Mass ratio of the tuned mass damper mass in relation to the total mass	[-]
μ_d	Bridge deck mass per unit span length	[kg/m]
μ_p	Pedestrian mass per unit span length	[kg/m]
ϕ_i	Phase shift for the ith harmonic	[rad]

Symbol	Definition	Unit
ϕ_n	Real part of the complex-valued eigenvector of mode n	[m/rad]
$\psi_n, \bar{\psi}_n$	Conjugate pair of the complex-valued eigenvector of mode n	[m/rad]
ψ_i	Reduction coefficient for the magnitude of loading based on critical frequency in the ith direction	[-]
ρ_{opt}	Optimum tuning ratio for TMD design	[-]
ξ	Structural damping ratio	[-]
$\xi_{d,opt}$	Optimum damping ratio for TMD design	[-]
$\xi_{s,avg}$	Average structural damping ratio for steel pedestrian bridges	[-]
$\xi_{s,min}$	Minimum structural damping ratio for steel pedestrian bridges	[-]
ω_n	Angular eigenfrequency of mode n	[rad/s]

1.1 Research context

Footbridges have been a fundamental part of human infrastructure, dating back to early civilizations, with examples like the Tarr Steps (1000 BC) and Pons Sublicius (600 BC) showcasing their historical significance [1, 2]. Modern-day advancements in materials and engineering techniques have enabled the construction of lightweight and architecturally intricate footbridges reaching spans of over one hundred metres. These structures are not only designed for utility but are often intended as an eye-catcher for the public as well. These advancements come with design challenges. The growing demand for slender and lightweight structures has heightened the vulnerability of footbridges to human-induced vibrations, resulting in challenges such as resonance and reduced comfort. Pedestrian movement, characterized by walking velocity, stride length, step frequency, and loading patterns, interacts with the dynamic properties of the structure. When step frequencies coincide with the eigenfrequencies of the footbridge, resonance effects can amplify vibrations, leading to discomfort or, in extreme cases, structural failure.

External damping devices, such as TMDs, are effective in controlling specific eigenfrequencies and improving dynamic performance. However, their design, implementation, and maintenance require detailed knowledge of the structure's dynamic behaviour, often relying on in-situ testing or empirical data from similar structures. This process can be intensive and costly. Moreover, suboptimal tuning or improper installation of these devices often leads to reduced effectiveness.

An alternative approach to address dynamic issues, through geometric optimization of the footbridge design. By altering dynamic characteristics such as eigenfrequencies and stiffness, it is possible to mitigate resonance effects without relying on external damping systems. This approach offers a robust, cost-effective solution with reduced maintenance requirements. The complexity of this problem primarily lies in the stochastic nature of pedestrian-induced loads, explained by variations in walking behaviour, group clustering, step frequencies, and phase lag contributions.

This research attempts to address the challenges of pedestrian-induced vibrations through a comprehensive methodology. By integrating assessment techniques, case study analysis, and geometric optimization, the study aims to develop a framework for designing footbridges that minimize or eliminate the need for external damping devices.

1.2 Problem statement

Due to rising demands, footbridge design has become more susceptible to human-induced vibrations. An increase in the pedestrian-structure mass ratio combined with slender, lightweight design makes for profound dynamic effects. Pedestrian movement is primarily defined by walking velocity, stride length, step frequency, and the nature of dynamic loading. When the step frequency of pedestrians, whether individuals or crowds, coincides with the eigenfrequency of a structure for a specific load, resonance effects are triggered. These induced vibrations can lead to loss of comfort resultant from accelerations rising to an unbearable level. Additionally, structural failure can occur due to collapse, instability, or unacceptable deformation.

The design and implementation of external damping devices demand an extensive understanding of a structure's dynamic behaviour, particularly when dealing with complex systems like footbridges. Accurately determining the dynamic characteristics of such structures, specifically, their eigenfrequencies and levels of structural damping, poses a significant challenge. Determining these characteristics is often regarded as a complex task, relying heavily on empirical data obtained from similar structures or, more commonly applied, through comprehensive in-situ testing. These methods, although reliable can be intensive and time-consuming.

When designed and tuned correctly, external damping devices can be highly effective in improving the dynamic response of a footbridge. By targeting specific locations on the structure, damping devices can mitigate undesirable vibrations without requiring significant material addition. This makes for an attractive option enhancing structural performance with marginal impact on the overall design. However, it is important to note that these damping systems are often misunderstood, leading to suboptimal tuning or improper implementation, and frequently resulting in significant costs and delays outweighing their implementation.

Contrary to external damping devices, if the dynamic characteristics of a footbridge could be altered or optimized in such manner that the need for external damping is either reduced or eliminated, a more robust and redundant design could be achieved. Such an approach would not only simplify the structural design but likewise reduce the long-term operational and maintenance costs associated with damping devices.

1.3 Research methodology

The research methodology is structured into three key phases: assessing human-induced vibrations, analyzing a case study, and optimizing the bridge geometry to minimize vibrations.

The first part consists of a systematic literature review on pedestrian-induced vibrations in footbridges. Various assessment methods are examined, evaluating their theoretical foundations, practical applicability, and limitations. This review serves as the basis for selecting the most suitable evaluation techniques for the case study.

In the second part, a case study is conducted using a finite element (FE) model of a pedestrian bridge developed in SOFiSTiK. The model is validated to ensure compliance with Ultimate Limit State (ULS) criteria, and its dynamic properties, as the eigenfrequencies, mode shapes, and maximum acceleration response—are analysed under pedestrian-induced loading scenarios. Additionally, a Tuned Mass Damper (TMD) is incorporated into the original model to assess its effectiveness in vibration mitigation.

The third part focuses on optimizing the bridge geometry to reduce human-induced vibrations and minimize reliance on external damping devices. A Multi-Objective Particle Swarm Optimization (MOPSO) algorithm is employed, with the objective of improving structural efficiency while reducing peak accelerations. The optimization process considers key design variables, such as deck geometry and stiffness distribution, while ensuring compliance with structural strength, serviceability limits, and pedestrian comfort criteria. The optimized solutions are then compared with the original design to evaluate performance improvements.

1.4 Research questions

The main research question is formulated as follows:

”What is the impact of parametric optimization of footbridge geometry on mitigating pedestrian-induced vibrations, and how does it affect the necessity of external damping systems?”

To answer the main research question, the following sub-questions are proposed:

1. ”What methods are most effective for assessing pedestrian-induced vibrations in footbridges?” (Part I & II)”
2. ”How can geometric parameter studies be performed to optimise footbridge performance, tailored to reducing the need for external damping devices? (Part III)”

1.5 Thesis outline

The report is structured into three main parts. Part I presents a literature study on assessment methods, highlighting their differences and outlining procedures for effective evaluation. Part II introduces a case study of a pedestrian bridge design, applying the selected assessment methods while exploring TMD design and establishing the parametric model for optimization. Part III focuses on the optimization process, employing a stochastic, meta-heuristic population-based algorithm to eliminate the need for external damping in footbridge design. The optimized results are then compared to the initial design, followed by a discussion of findings and their limitations. At last, conclusions are drawn and recommendations for further research are presented.

Part I

Assessment of pedestrian-structure interaction

2

Literature review

Over the past thirty years, various studies have been conducted on the assessment of human-induced vibrations of footbridges. One could argue that findings were accelerated by the temporary closure of the now famous Paris' Passerelle Solférino (1999)¹ and London Millenium footbridge (2000)². Two bridges which during inauguration experienced pedestrian-structure interaction, a phenomenon described as lateral lock-in, see 2.2 Lateral lock-in effect, causing excessive lateral vibrations leading to serious loss of comfort. Following this event, the footbridges were submitted to thorough in-situ testing, resulting in new guidelines [4, 5] and sparking the need for testing and numerical simulation of other lightweight footbridges in Europe [6].

As stated by European guideline SYNPEX: "Vibrations of footbridges may lead to serviceability problems, as effects on the comfort and emotional reactions of pedestrians might occur. Whilst collapse or even damage due to human-induced dynamic forces have occurred very rarely [6]". Additionally, the dynamic loading of cyclists has close to no impact in comparison to pedestrians, leaving only pedestrian-induced load models up for analysis. Whenever the structures' eigenfrequency and the pedestrians' pacing frequency coincide, resonance occurs, resulting in unwanted vibrations.

This literature review primarily focuses on the evaluation of human-induced vibrations that affect footbridge serviceability leading to discomfort. By means of various methods available in literature, namely analytical formulas and (finite element method) FEM analysis, vibrations are accessed and verified. Lastly, the concept of external damping through use of a tuned mass damper (TMD) is presented. This form of damping is highlighted since it is the most commonly applied method to suppress human-induced vibrations.

2.1 Maximum acceleration

For the vibration serviceability of footbridges, numerous assessment methods exist in literature. Ranging from international codes [7–10] to guidelines [4, 5, 11] and standards [12, 13]. All sources in essence state, that maximum accelerations albeit vertical and horizontal, for any part of the bridge deck should be lower than a certain limit value:

$$a_{i,max} < a_{i,lim} \quad (2.1)$$

where:

$a_{i,max}$ = maximum deck acceleration for any part of the bridge deck in the i th direction,
 $a_{i,lim}$ = limit value for acceleration in the i th direction.

¹On December 15, 1999, the Passerelle Solférino footbridge, a 140 m long steel arch footbridge across the Seine in Paris, was opened to the public for crossing. On the opening day, unexpected lateral oscillations were observed and the bridge was subsequently closed. Installation of fourteen tuned mass dampers (TMDs) followed by vibration testing and monitoring of the bridge. In November 2000, the bridge was reopened after almost a year of closure [3].

²The London Millennium Bridge, which connects St. Paul's Cathedral with the Tate Modern Gallery is a shallow suspension bridge in three spans; a south span of 108 m, a central span of 144 m and a north span of 81 m. On June 12, 2000, it was decided to close the bridge while a retrofit solution could be developed and implemented. During the next eighteen months, an extensive test program, similar to that in Paris was undertaken. After one and a half years, the bridge was reopened [3].

How to obtain these maximum accelerations and what these limit values are, differs substantially. The available methods hold regard to the acceptability of vibrations within footbridges based on human perception levels. The ISO baseline curve [12] presents the threshold of human perception, whereas for footbridges this curve is multiplied by a certain factor. Figure 2.1 presents the ISO baseline curve of the peak acceleration for different loading frequencies, whereas figure 2.2 presents the root mean square (RMS) acceleration for footbridges found in literature [14], quantifying the overall response to human-induced vibrations.

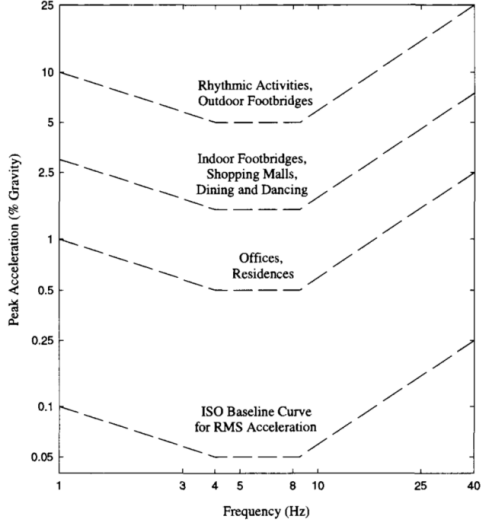


Figure 2.1: ISO baseline curve for human perception and vertical peak acceleration limits for various civil structures

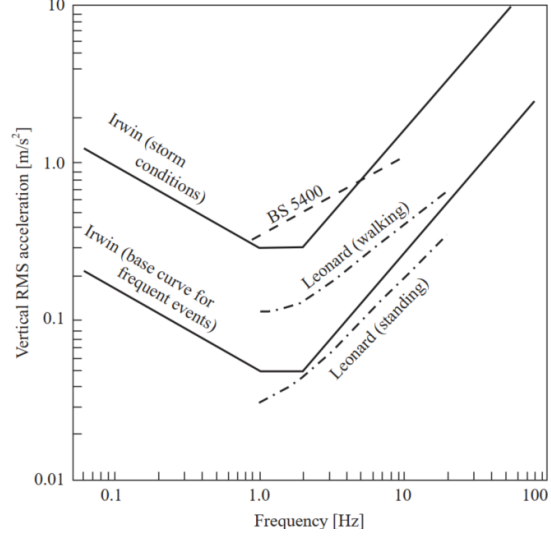


Figure 2.2: Vertical RMS acceleration limits found in literature

Figure 2.3 presents vertical acceleration limits as specified in standards and guidelines. Observed can be that both an increased and decreased maximum acceleration limit is prescribed for ranging natural frequencies f_n , resulting in contradictory results. Whereas figure 2.4 presents the vertical acceleration limits according to comfort studies, in which the studies by Matsumoto [15], Wheeler and Korenev [16] are not necessarily tailored to footbridge structures but accelerations on the human body in general.

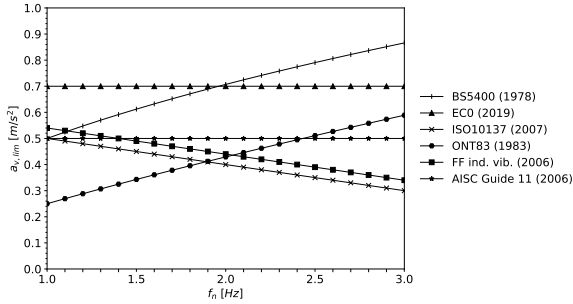


Figure 2.3: Limit values for vertical acceleration enlisted in standards and guidelines

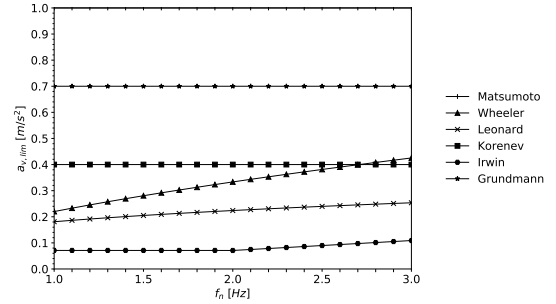


Figure 2.4: Comfort values for vertical acceleration enlisted in literature [6]

For horizontal limit accelerations, Eurocode EN1990 (2019) [9] specifies thresholds for normal and exceptional use. Exceptional use refers to scenarios such as a marching band or a group of hooligans crossing a bridge. Figure 2.5 shows these limit values, including the beginning of the lateral lock-in effect, $a_{h,lock-in} = 0.10 \sim 0.15 \text{ m/s}^2$, as explained in 2.2 Lateral lock-in effect. As for comfort values, figure 2.6 shows horizontal accelerations found in literature as outlined in the SYNPEX publication [6].

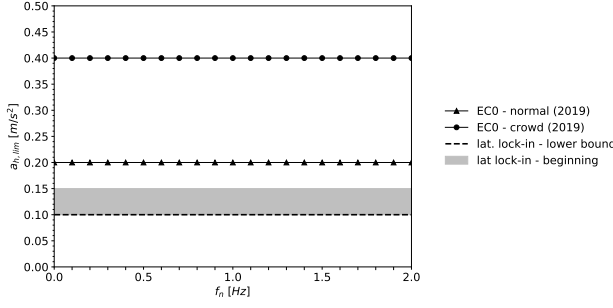


Figure 2.5: Limit values for horizontal acceleration enlisted in standards and guidelines[6]

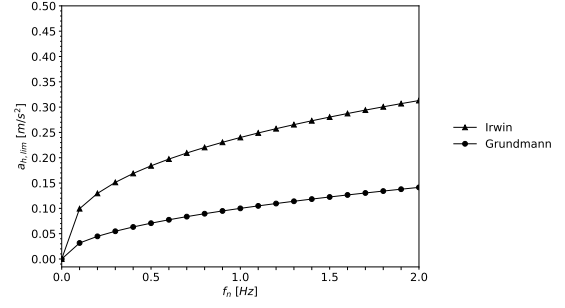


Figure 2.6: Comfort values for horizontal acceleration enlisted in literature [6]

Observing figures 2.3 to 2.6, one could state that the available literature lacks consensus on the specified accelerations limits. The aforementioned multiplication factor of the ISO baseline curve is not fixed, but is influenced by various contributing factors. Furthermore, research on horizontal accelerations is comparatively limited, for which often the lateral lock-in effect is overlooked.

In addition to defining acceleration limit values, it is essential to determine the accelerations present in the structure. One of the earliest methods for evaluating vertical vibration serviceability was introduced in the British Standard BS5400 (1978) [7]. Therein a simplified method is mentioned for one, two or three-span continuous symmetric superstructures, using a configuration factor based on span layout:

$$a = 4\pi^2 f_o^2 y_s K \psi \quad (2.2)$$

where:

f_o = fundamental natural frequency [Hz]

y_s = static deflection [m]

K = configuration factor (based on span layout)

ψ = dynamic response factor (based on design graph for various damping ratio's)

Additionally, a general method for quantifying maximum vertical acceleration is mentioned. By assuming dynamic loading applied by a single pedestrian represented by a pulsating point load F_{ped} , moving across the main span of the superstructure at a constant speed $v = 0.9 f_o$, see equation 2.2. The code states that the maximum obtained acceleration should be limited to $a_{max} = 0.5\sqrt{f_o}$ m/s², however, the means of obtaining this maximum acceleration is unspecified.

$$F_{ped}(t, v) = 180 \sin(2\pi f_o t) \delta(x - vt) \quad (2.3)$$

where:

$F_{ped}(t, v)$ = dynamic load of a single moving pedestrian [kN]

f_o = fundamental frequency [Hz]

$\delta(x - vt)$ = delta dirac function

As previously mentioned, do vertical and horizontal acceleration limits vary within a substantial bandwidth, for which no account of the lateral lock-in effect is made. The method of determining the maximum acceleration present is either simplified as for girder bridges or unspecified, leaving the assessor limited in quantifying the response. This is why JRC-document EUR23984 [5] and French guideline S etra [4] enlist a more extensive and comparable design methodology, aiming for improved assessment, as presented in figures 2.7 and 2.8 respectively.

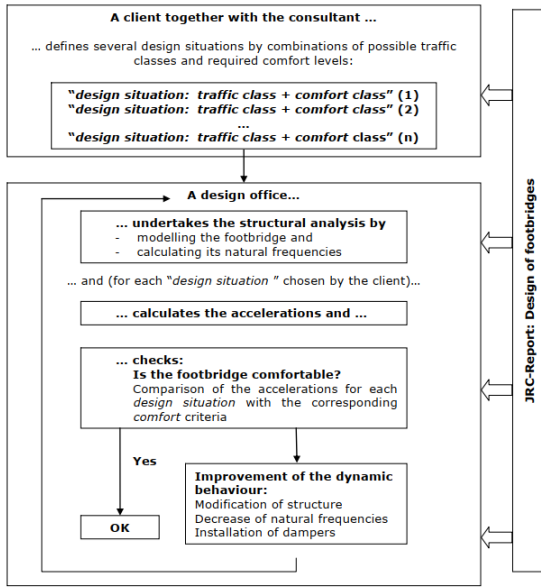


Figure 2.7: Design procedure enlisted in EUR23984

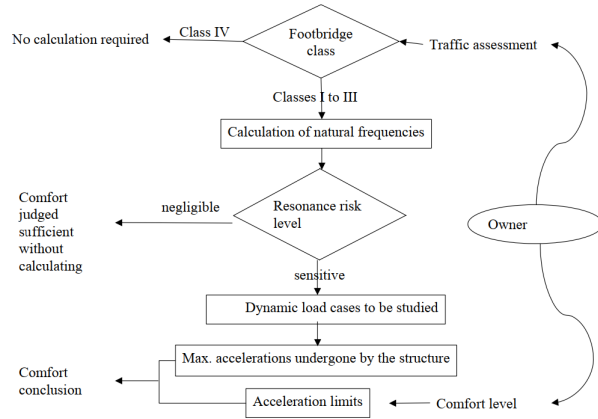


Figure 2.8: Design procedure enlisted in S tra guideline

The procedure is characterized by the client specifying all possible design situations based on expected traffic and required comfort. Natural frequencies are determined and verified whilst within critical range. Assessment of structural damping follows, and by use of dynamic load models, the maximum acceleration is evaluated. Furthermore, the occurrence of lateral lock-in is checked and the maximum acceleration of the bridge deck is verified against the limit value. If needed, the structure is modified in terms of mass, stiffness and/or by applying additional damping devices.

The newly proposed Eurocode prEN1990 (2021) [10] matches the assessment outlined by both guidelines, stating that: "the assessment of human-induced vibrations should be considered to ensure that: vibrations due to pedestrian traffic are acceptable for users; the lock-in phenomenon does not arise and the footbridge does not collapse when subjected to intentional excitation (accidental limit state)". The refined assessment methodology is outlined from section 2.3 onwards, providing a better understanding of human-induced vibrations.

2.2 Lateral lock-in effect

The lateral lock-in effect is concerned with horizontal accelerations and is to be subjected to verification in almost all methodologies after the events of the Paris' Passerelle Solf rino and London Millenium footbridge. When a bridge deck is subjected to lateral vibrations and a pedestrian walks across, additional movement that the pedestrian experiences is compensated for by aligning the swaying motion of the bridge deck with its centre of gravity. The pedestrian's behaviour is intuitive and even small vibrations lead to an adjustment of movement. This change of movement is achieved by adapting the walking frequency and widening of the gait, resulting in a greater dynamic load factor (DLF), see 2.6. Looking at a group of pedestrians crossing the bridge, certain individuals will start synchronizing their movement which will increase over time affecting more pedestrians.

A study by  ivanovi  et al. [17] showed that the factors contributing to this synchronisation effect are: the natural frequency of the footbridge, amplitude of the response, number of people involved, density and velocity of the people. Fortunately, the S tra guideline [4] states that: "Actual synchronisation is much weaker and when footbridge movement is such that pedestrians can no longer put their best foot forward, they have to stop walking and the phenomenon can no longer evolve". Figure 2.9 gives a schematic description of synchronous walking. The increase of synchronization over time is also described as lateral lock-in effect since people tend to get "locked in" to a certain motion.

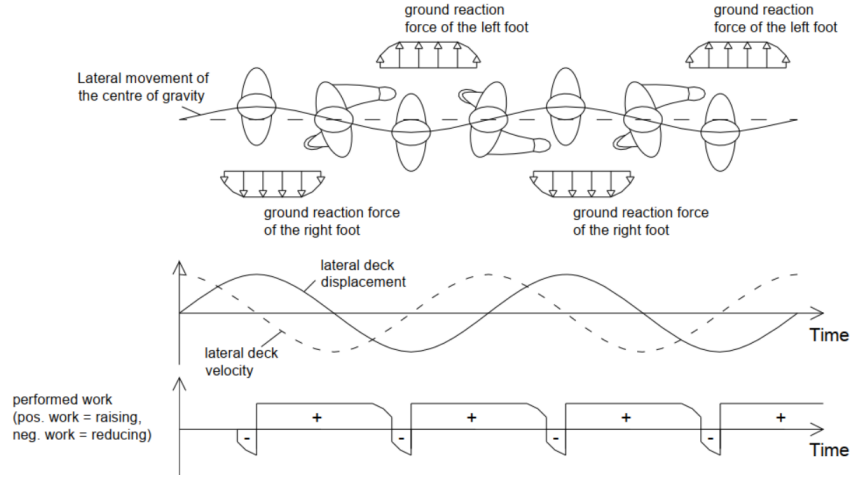


Figure 2.9: Schematic description of synchronous walking

Lateral lock-in can be explained as a person's movement being synchronised with the vibration of the bridge, resulting in ground reaction forces being in line with ground displacement resulting in positive work. Whenever the deck's natural frequency matches the lateral vibration, resonance takes place which gives rise to large vibrations over time. A publication by Ellis [18] explains that: "The force generated laterally by the right foot is normally in the opposite direction to the force induced by the left foot. Therefore the time required for reproducing the same pattern of force can be counted from the left (or right) foot to the next step of the left (or right) foot. In other words, the period of lateral forces is just twice the period of the vertical force induced by people walking". This means that synchronous walking leads to double the natural frequency for lateral vibrations, see equation 2.4.

$$\lim_{t \rightarrow \infty} f_{s.sync}(t) = 2 \cdot f_{n.lat} \quad (2.4)$$

where:

$f_{s.sync}(t)$ = Synchronised step frequency of pedestrians over time
 $f_{n.lat}$ = Natural frequency for lateral vibrations

Important to note is that the lock-in effect could potentially give rise to resonance via pedestrian synchronisation. Tests on a rig in France and the Passerelle Solferino footbridge [4] indicate that lock-in happens when the lateral natural frequency lies within a certain range of half the possible step frequencies and the deck reaches an amplitude of $0.10 \sim 0.15 \text{ m/s}^2$ or higher, see condition 2.5. Results from the test rig are also depicted in figure 2.10, showing the synchronisation starting from 0.10 m/s^2 onwards.

$$a_{deck.lat} > a_{st.lock-in} \quad \& \quad 0.8 < f_{n.lat} < 1.2 \text{ Hz} \quad (2.5)$$

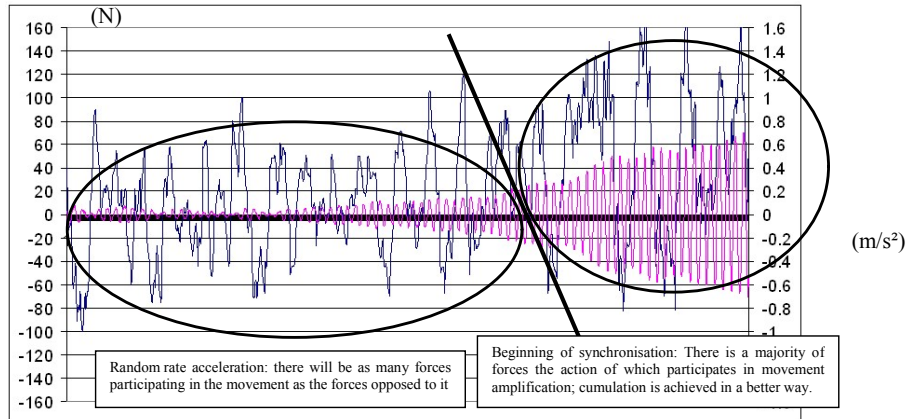


Figure 2.10: Acceleration (m/s^2) [pink] and applied force (N) [blue] with ten random pedestrians on the test rig [4]

Bachmann specifies the largest risk of excitation by pedestrians (95 % confidence interval) in vertical direction for a range of $f_s = 1.6 \sim 2.4$ Hz and for horizontal direction half this frequency, giving $f_{s,lat} = 0.8 \sim 1.2$ Hz. Other codes and guidelines differ from this interval (see table 2.6), specifying a broader range of at maximum $f_{n,lat} = 0.3 \sim 1.3$ Hz.

An additional approach to verify lateral lock-in is to consider an equivalent amount of pedestrians N_L which triggers amplification. Equation 2.6 denotes the number of pedestrians that could lead to a vanishing of the overall damping, producing a (sudden) amplified response. When the total number of pedestrians on the bridge exceeds this threshold, lateral lock-in is likely to occur. This approach has been validated through tests conducted on the Millennium Footbridge [19], as well as experiments on footbridges in Portugal [20], which reported acceleration amplitudes in the range of $0.15 \sim 0.20$ m/s², suggesting these approaches might be related.

$$N_L = \frac{8\pi\xi m_n f_{n,lat}}{k} \quad (2.6)$$

where:

- N_L = Number of people triggering lateral lock-in
- ξ = structural damping ratio
- m_n = modal mass
- $f_{n,lat}$ = natural frequency within critical range
- k = constant (300 Ns/m approximately over the range of $0.5 \sim 1.0$ Hz, by Dallard et al. [19])

2.3 Design situations

Design situations are specified by the client in collaboration with the consultant, to account for all anticipated traffic and desired comfort levels for each potential scenario throughout the bridge's lifetime (see Figure 2.11). Both the EUR23984 and S etra guideline mention design scenarios, ensuring the comfort level is more effectively aligned with actual events whilst addressing the bandwidth of accelerations presented earlier.

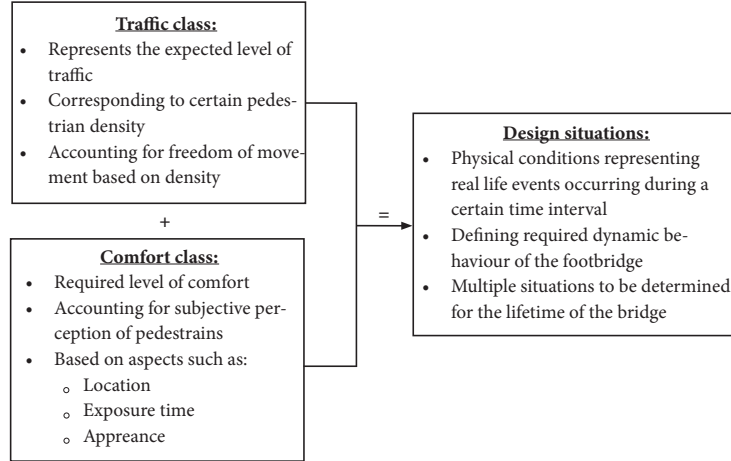


Figure 2.11: Design situation approach

EUR23984 states that: "It is strongly recommended to discuss comfort requirements and expected pedestrian traffic to the obtained dynamic response with the owner to develop realistic limits and boundary conditions for the design of the particular structure", meaning that different design situations account for different pedestrian comfort. An example would be a once-in-a-lifetime event, like the inauguration of a footbridge. A persistent design situation, including the everyday density of pedestrians, would be valued at a higher level of comfort. As for the S etra guideline: "Depending on footbridge class and on the ranges within which its natural frequencies are situated, it is necessary to carry out dynamic structure calculations for all or part of a set of three load cases." Both design situation approaches are shown in tables 2.1 and 2.2.

Design situation	Description	Traffic class	Expected occurrence	Comfort class
1	inauguration of the bridge	TC4	once in the lifetime	CL3
2	commuter traffic	TC2	daily	CL1
3	rambler at the weekend	TC1	weekly	CL2
⋮	⋮	⋮	⋮	⋮

Table 2.1: Design situations - EUR23984

Traffic	Class	Natural frequency range		
		1	2	3
sparse	III		nil	nil
dense	II	case 1	case 1	case 3
very dense	I	case 2	case 2	case 3

Table 2.2: Design situations - Sétra guideline

By taking into account the various design situations across the lifespan of the bridge, a more realistic assessment is achieved rather than verifying for one limit value as required for the limits given in figures 2.3 to 2.6. EN1990 (2019) [9] specifies that multiple design situations are to be adopted if required by the client, but does not list a range of limit accelerations as per set in both EUR23984 and Sétra guideline. The newly proposed Eurocode prEN1990 (2021) [10] fully adopts the acceleration range specified in the JRC-document. It should be noted that these design situations do not account for pedestrian formations or marching soldiers, which will need additional consideration if specified by the client [5].

2.3.1 Traffic classes

Traffic classes represent the effects of pedestrians in a simplified and applicable manner to quantify realistic behaviour for the structure in question. As for EUR23984, see figure 2.12, a pedestrian density range of $d = 0.1 \sim 1.5$ p/m² is specified regarding the respective traffic classes.




Traffic Class	Density d (P = pedestrian)	Description	Characteristics
TC 1*)	group of 15 P ; $d = 15 P / (B L)$	Very weak traffic	(B =width of deck; L =length of deck)
TC 2	$d = 0,2 P/m^2$	Weak traffic 	Comfortable and free walking Overtaking is possible Single pedestrians can freely choose pace
TC 3	$d = 0,5 P/m^2$	Dense traffic 	Still unrestricted walking Overtaking can intermittently be inhibited
TC 4	$d = 1,0 P/m^2$	Very dense traffic 	Freedom of movement is restricted Obstructed walking Overtaking is no longer possible
TC 5	$d = 1,5 P/m^2$	Exceptionally dense traffic	Unpleasant walking Crowding begins One can no longer freely choose pace

Figure 2.12: Traffic classes in JRC-document

A study by Oeding [21] shows a correlation between pedestrian densities and step frequencies by taking the mean value between walking speed and traffic density. The study shows that when pedestrian density is increased, walking velocity and subsequently step frequency is reduced. Low pedestrian densities $d < 0.5 \text{ p/m}^2$ do not obstruct the flow of traffic, meaning that walking velocity is not reduced and there is freedom of movement. For dense traffic $d = 0.5 \sim 1.0 \text{ p/m}^2$, passing becomes more difficult and a single pedestrian has to adjust walking velocity to the moving crowd. Above a density of $p \geq 1.0 \text{ p/m}^2$, very dense traffic makes that freedom of movement is restricted and overtaking is no longer possible. Exceptionally dense crowds $p \geq 1.5 \text{ p/m}^2$, completely obstruct freedom of movement and the walking velocity is greatly reduced.

French guideline Sétra specifies a pedestrian density range of $d = 0.5 \sim 1.0 \text{ p/m}^2$, see table 2.3. There is no traffic class enlisted for either weak or exceptionally dense traffic as for the JRC-document.

Class	Density of crowd
III	0.5 pedestrians / m^2
II	0.8 pedestrians / m^2
I	1.0 pedestrians / m^2

Table 2.3: Traffic classes in Sétra guideline

For the pedestrian densities given in both guidelines, assumed is a crowd being uniformly distributed over the total area of the footbridge, for which the frequency and phase of the pedestrians coincide with one or more natural frequencies of the structure in question. Important to note is that these (general) traffic classes do not account for pedestrian formations. Formations can be characterised by a marching lock-step, in some cases enforced by music such as parades and processions. The difference between general traffic and pedestrians formation is that when formation takes place: "The single pedestrian is unable to move freely because of a given beat or linked arms. Hence the step phase is widely synchronized [4]". This synchronised lock-step behaviour can lead to strong dynamic loading and loss of comfort when frequencies align, requiring additional consideration if specified by the client.

2.3.2 Comfort classes

Comfort is assessed by acceleration undergone by any part of the structure during loading. Given the subjective nature of comfort, classes are defined by an acceleration range. Table 2.4 shows these ranges for both the EUR23984 and Sétra guideline. Contrary to these ranges are the acceleration limits as per specified in figures 2.3 / 2.6, for which no account is made regarding multiple design situations and limit acceleration, both vertically and horizontally, is limited to a single value.

Comfort class	Degree of comfort	Vertical limit	Horizontal limit
I	Maximum	$< 0.50 \text{ m/s}^2$	$< 0.10 \text{ m/s}^2$ ³
II	Medium	$0.50 \sim 1.00 \text{ m/s}^2$	$0.10 \sim 0.30 \text{ m/s}^2$
III	Minimum	$1.00 \sim 2.50 \text{ m/s}^2$	$0.30 \sim 0.80 \text{ m/s}^2$
IV	Unacceptable	$> 2.50 \text{ m/s}^2$	$> 0.80 \text{ m/s}^2$

Table 2.4: Comfort levels in EUR23984 and Sétra guideline

In general, a comfort class is determined by the client based on its users and importance. Elderly or disabled people frequently using a footbridge could result in a more demanding design. Additionally, there is many other 'soft' aspects which contribute to comfort such as: frequency of use; exposure time; expectancy of vibration and number of users.

³Sétra guideline gives for class I a horizontal acceleration limit of $a_h < 0.15 \text{ m/s}^2$

2.4 Design steps

In addition to the general procedures outlined in EUR23984 and the Sètra guidelines, as shown in Figures 2.7 and 2.8 respectively, specific design steps are provided. These steps offer a systematic breakdown of the assessment process, utilizing methods such as spectral analysis, SDOF, or FEM evaluation. For each design scenario, the limits are verified, and recommendations for improving the structure and its damping characteristics are provided. This section presents the key aspects of these design steps. Due to their significant impact on the overall design procedure, dynamic load models and the determination of accelerations are discussed in separate sections.

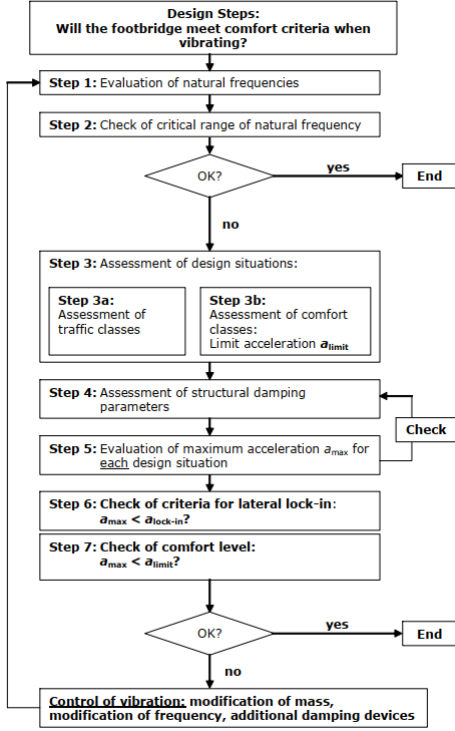


Figure 2.13: Design steps according to EUR23984 [5]

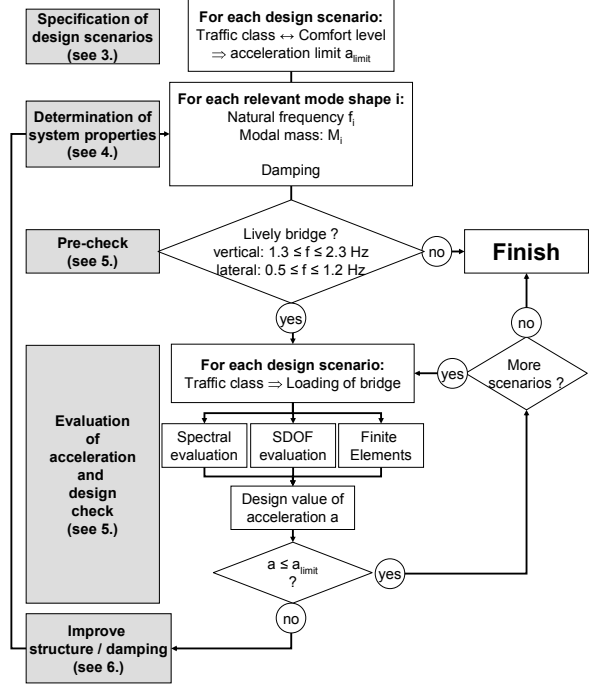


Figure 2.14: Design steps according to "European design guide for footbridge vibration" publication [22]

2.4.1 Evaluation of natural frequencies

As prEN1991-2 states [23], natural frequencies should be computed using either analytical solutions from closed-form expressions and iterative methods, or by use of numerical solutions with discretized models such as Finite Element Models (FEM). When a bridge structure can be idealised by a 1D beam element or rectangular 2D plate element, analytical calculation is possible as given by equation 2.7.

$$f_{b,n} = \frac{n^2 \pi}{2L^2} \cdot \sqrt{\frac{EI}{\rho A}} \quad f_{p,m,n} = \sqrt{\frac{D}{\rho h}} \cdot \left[\left(\frac{m\pi}{a} \right)^2 + \left(\frac{n\pi}{b} \right)^2 \right] \quad (2.7)$$

where:

$f_{b,n}$ = Natural frequencies of a simply supported beam of constant characteristics for bending at n sags

$f_{p,m,n}$ = Natural frequencies of a simply supported rectangular plate with length a and width b for bending at m and n sags

Next to closed-form expressions, numerical solutions can be obtained through means of discretized models. Generally speaking, most FEM software apply modal analysis in which the method of Lanczos [24] is widely used to obtain eigenvalues. Important to note is that the EUR23984 states that: "It is recommended that the mass of pedestrians should be considered when calculating the natural frequencies only when the modal mass of the pedestrians is more than 5% of the modal deck mass." Whereas the

Sétra guideline states that eigenfrequency analysis is determined for two mass assumptions, namely an empty footbridge and a footbridge loaded throughout its bearing area at one 700 N pedestrian per square meter (70 kg/m²), giving a range of frequencies.

The equation of motion (EOM) for a (discretized) multiple degrees-of-freedom system is represented by:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) \quad (2.8)$$

To perform an eigenvalue analysis, the assumption is made that the response $\mathbf{u}(t)$ is modelled by a mode shape vectors Φ that control the shape, and a vector $\mathbf{q}(t)$ that controls the amplitude:

$$\mathbf{u}(t) = \Phi \cdot \mathbf{q}(t) \quad (2.9)$$

If undefined, the eigenvalue analysis is most often performed for the undamped system. "Structures such as buildings, bridges, dams, and offshore structures, have a damping ratio of $\xi < 0.15$ and thus can be categorized as underdamped structures. The basic dynamic properties estimated using damped or undamped assumptions are approximately the same" [25]. For which the EOM is given by:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{0} \quad (2.10)$$

Substituting a harmonic function of circular frequency ω_m for the modal displacement gives:

$$q_m = A_m \cos(\omega_m t) + B_m \sin(\omega_m t) \quad (2.11)$$

where:

A_m , B_m = Unknown constants, to be determined by initial conditions: $u(0) = u_0$, $\dot{u}(0) = \dot{u}_0$.

The nontrivial solution of this eigenvalue problem reads:

$$\det [\mathbf{K} - \omega_m^2 \mathbf{M}] = 0 \quad (2.12)$$

Which results in N real and positive frequencies arranged $\omega_1 < \omega_2 < \dots < \omega_n$. Once the eigenfrequencies are obtained, the corresponding mode shapes are determined by:

$$[\mathbf{K} - \omega_m^2 \mathbf{M}] \Phi = 0 \quad (2.13)$$

In addition to undamped systems, classical damping such as modal or Rayleigh damping could be introduced for a more accurate eigenvalue analysis, see section 2.4.4.

2.4.2 Critical range of natural frequencies

A footbridge experiences resonance whenever the structures' eigenfrequency and the pedestrians' pacing frequency coincide. It is therefore possible to define a critical range where natural frequencies pose problems due to this resonance phenomenon. The bounds of this critical range are set by the walking behaviour of pedestrians using the footbridge.

Bachmann [26] performed a study on the stepping rate of pedestrians, stating that: "The average walking rate is 2 Hz with a standard deviation of 0.175 Hz. meaning that 50% of pedestrians walk at rates between 1.9 Hz and 2.1 Hz, or alternatively, 95% of pedestrians walk at rates between 1.65 and 2.35 Hz". Table 2.5 shows data of this study with the belonging design rate for each activity.

Representative types of activity			Range of applicability		
Designation	Definition	Design activity rate [Hz]	Actual activities	Activity rate [Hz]	Structure type
"walking"	walking with continuous ground contact	1.6 to 2.4	<ul style="list-style-type: none"> slow walking (ambling) normal walking fast, brisk walking 	~ 1.7 ~ 2.0 ~ 2.3	<ul style="list-style-type: none"> pedestrian structures (pedestrian bridges, stairs, piers, etc.) office buildings
"running"	running with discontinuous ground contact	2.0 to 3.5	<ul style="list-style-type: none"> slow running (jog) normal running fast running (sprint) 	~ 2.1 ~ 2.5 ≥ 3.0	<ul style="list-style-type: none"> pedestrian bridges on jogging tracks, etc.
"jumping"	normal to high rhythmical jumping on the spot with simultaneous ground contact of both feet	1.8 to 3.4	<ul style="list-style-type: none"> fitness training with jumping, skipping and running to rhythmical music jazz dance training 	~ 1.5 to 3.4 ~ 1.8 to 3.5	<ul style="list-style-type: none"> gymnasias, sport halls gymnastic training rooms
"dancing"	approximately equivalent to "brisk walking"	1.5 to 3.0	<ul style="list-style-type: none"> social events with classical and modern dance (e.g. English Waltz, Rumba, etc.) 	~ 1.5 to 3.0	<ul style="list-style-type: none"> dance halls concert halls and other community halls without fixed seating
"hand clapping with body bouncing while standing"	rhythmical hand clapping in front of one's chest or above the head while bouncing vertically by forward and backward knee movement of about 50 mm	1.5 to 3.0	<ul style="list-style-type: none"> pop concerts with enthusiastic audience 	~ 1.5 to 3.0	<ul style="list-style-type: none"> concert halls and spectator galleries with and without fixed seating and "hard" pop concerts
"hand clapping"	rhythmical hand clapping in front of one's chest	1.5 to 3.0	<ul style="list-style-type: none"> classical concerts, "soft" pop concerts 	~ 1.5 to 3.0	<ul style="list-style-type: none"> concert halls with fixed seating (no "hard" pop concerts)
"lateral body swaying"	rhythmical lateral body swaying while being seated or standing	0.4 to 0.7	<ul style="list-style-type: none"> concerts, social events 		<ul style="list-style-type: none"> spectator galleries

Table 2.5: Pedestrian activity according to Bachmann [26]

In accordance with this study, several guidelines and codes have specified the critical natural frequency ranges which should either be avoided entirely, or have the structure be subjected to an extensive dynamic vibration analysis. Table 2.6 shows the critical ranges for vertical, longitudinal, lateral and torsional vibrations. Within this research, the limits as proposed by Eurocode prEN1990-2021 are seen as critical ranges.

Guideline / Code	Vertical / longitudinal	Type of vibration [Hz]	
		Vertical / longitudinal including 2nd harmonic	Lateral / torsional
EUR23984	1.25 \sim 2.3	1.25 \sim 4.6	0.5 \sim 1.2
S��tra	1 \sim 2.6	1 \sim 5	0.3 \sim 1.3 ⁴
EN1990-2019	< 5	-	< 2.5
prEN1990-2021	1.25 \sim 2.3	1.25 \sim 4.6	0.5 \sim 1.2
	1.9 \sim 3.5 ⁵		
	1.7 \sim 3.0 ⁶		

Table 2.6: Critical frequency ranges as per specified in guidelines and codes

2.4.3 Assessment of design situations

The perception of motion is mostly subjective, meaning that people tolerate vibrations based on their assessment of the situation. When a footbridge is located near a hospital, pedestrians might be more sensitive to vibrations than supporters on their way to a football match. Another factor is the expectation due to the bridge's appearance, meaning that a bridge which looks sturdy but undergoes (adequate) vibrations will have a large impact on pedestrian assessment.

⁴Critical frequency range of 0.3 \sim 2.5 Hz for the 2nd harmonic

⁵Range for jogger excitation

⁶Range for intentional excitation

A good example of this perception bandwidth is outlined in a study by RWTH Aachen [6], which consisted of a comfort questioning of pedestrians. Two bridges, namely the Wachtelsteg in Pforzheim and the Kochenhofsteg in Stuttgart, with the same natural frequency of 2 Hz and similar dynamic characteristics were referenced for this studies. Around forty pedestrians for both bridges were questioned on four topics of vibration perception. The data suggests that the Wachtelsteg in Pforzheim is perceived as 30% more disturbing (see figure 2.15 and 2.16) than the Kochenhofsteg in Stuttgart, despite both bridges having the same eigenfrequency and characteristics.

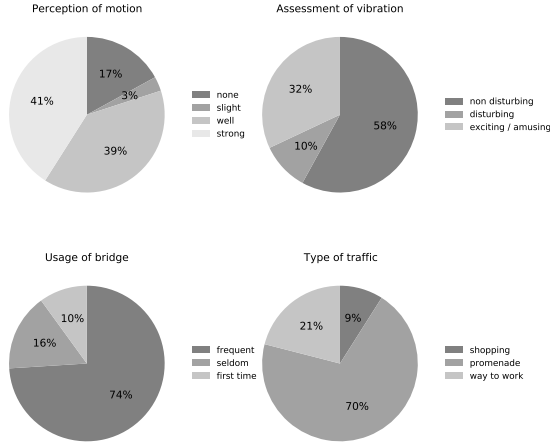


Figure 2.15: Results of pedestrian interviews concerning perception of vibration for the Kochenhofsteg in Stuttgart

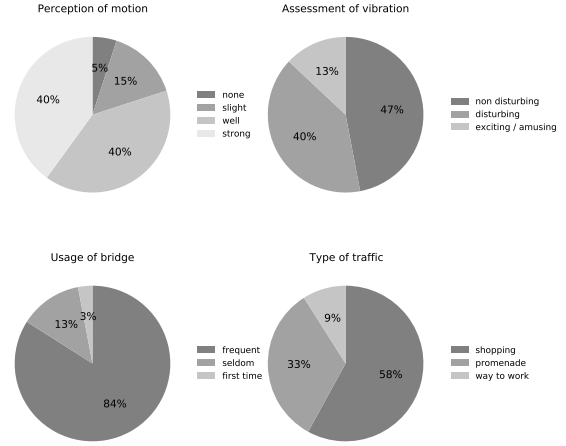


Figure 2.16: Results of pedestrian interviews concerning perception of vibration for the Wachtelsteg in Pforzheim

2.4.4 Assessment of structural damping

Structural damping consists of intrinsic damping of construction materials, local effects of bearings, the presence of non-structural elements (like handrails), and if present, damping by control devices. In order to account for the natural damping in a structure, various guidelines and codes specify the use of linear viscous dampers, implying damping proportional to the velocity of the bridge deck. Reason for this application is that linear dynamic equations arise, approximating only real damping for low levels of oscillation. This enables the use of modal analysis, greatly reducing complexity and computational time. However, the use of control devices could lead to a non-diagonal damping matrix, requiring iterative calculation like direct numerical integration methods to perform analysis. Table 2.7 shows the damping ratio's for various construction materials as per specified in codes and guidelines.

Construction material	Minimum ξ	Average ξ
Reinforced concrete	0.8%	1.3%
Prestressed concrete	0.5%	1.0%
Composite steel-concrete	0.3%	0.6%
Steel	0.2%	0.4%
Timber	1.0%	1.5%

Table 2.7: Minimum and average damping ratio's per construction type

In case of large vibrations, caused by intentional jumping or other exceptional design situations, larger damping ratios can be observed. prEN1991-2 states that: "For such cases, alternative values of damping may be used as agreed for the specific project by the relevant parties." Table 2.8 shows these increased damping ratios as specified in guidelines.

Construction type	Damping ratio ξ
Reinforced concrete	5%
Prestressed concrete	2%
Steel, welded joints	2%
Steel, bolted joints	4%

Table 2.8: Increased damping ratio's per construction type

Classical damping

Classical damping is deemed a valid approximation when similar damping mechanisms are distributed throughout the structure under the assumption that it is lightly damped ($\xi < 10\%$) [27]. This method generates a diagonal stiffness matrix allowing for modal analysis. The most commonly applied forms of classical damping are presented in this paragraph. Mathematically speaking, a damping matrix is defined classical if it is diagonalisable by a set of mode shapes Φ , such that:

$$C = \Phi^T c \Phi = \begin{bmatrix} c_1 & & \\ & \ddots & \\ & & c_n \end{bmatrix} \quad (2.14)$$

Superposition of modal damping

Modal damping is a combination of elastic vibration modes, in which damping ratio's resulting from experiments and empirical results of comparable structures are applied. It comprises of a (classical) damping matrix defined as:

$$C = \begin{bmatrix} 2\xi_1 M_1 \omega_1 & & \\ & \ddots & \\ & & 2\xi_n M_n \omega_n \end{bmatrix} \quad (2.15)$$

where:

- M_n = Generalised modal masses ($\Phi_n^T M \Phi_n$) at mode n
- ξ_n = Modal damping ratio
- ω_n = Angular frequency at mode n

The superimposed damping matrix c for N modes is given by:

$$C = M \left(\sum_{n=1}^N \frac{2\xi_n \omega_n}{M_n} \phi_n \phi_n^T \right) M \quad (2.16)$$

where:

- M = Mass matrix of the structure
- ϕ_n = Mode shape at mode n
- N = Number of modes considered for damping

Rayleigh damping

Rayleigh damping, also known as proportional damping, consists of a weighted sum of the mass and stiffness matrices:

$$C = a_0 M + a_1 K \quad (2.17)$$

Considering mass and stiffness proportional damping individually first, the following coefficients can be distinguished:

$$\xi_m = \frac{a_0}{2} \frac{1}{\omega_m} \qquad \xi_m = \frac{a_1}{2} \omega_m \quad (2.18)$$

where:

a_0 = Coefficient for mass-proportional damping at mode m
 a_1 = Coefficient for stiffness-proportional damping at mode m

What can be observed is that for mass-proportional damping a_0 , the damping ratio ξ_m is inversely proportional to the eigenfrequency ω_m . Whilst for stiffness-proportional damping a_1 , the damping ratio increases linearly with the eigenfrequency ω_m . Rayleigh damping combines these methods, giving a (classical) damping matrix as defined in equation 2.17. Figure 2.17 explains the concepts of mass- and stiffness-proportional damping and Rayleigh damping graphically.

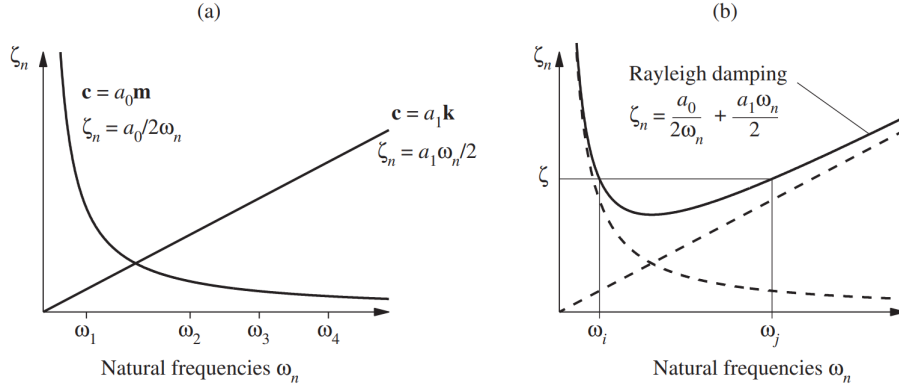


Figure 2.17: Variation of modal damping ratios with natural frequency: (a) mass-proportional damping and stiffness-proportional damping; (b) Rayleigh damping [28]

If the assumption is made that two modes contain the same damping ratio ξ , the coefficients a_0 and a_1 are determined by:

$$a_0 = \xi \frac{2\omega_i\omega_j}{\omega_i + \omega_j} \quad a_1 = \xi \frac{2}{\omega_i + \omega_j} \quad (2.19)$$

where:

ω_i = Lower frequency bound for the defined damping ratio
 ω_j = Upper frequency bound for the defined damping ratio

If the damping ratios of two principal modes are known (or can be estimated), the theory can be extended and the values of a_0 and a_1 can be determined by equation 2.20. Once these constants are obtained, a relationship can be established for the damping ratios of all orthogonal modes, allowing the damping ratio ξ_i to be determined for every distinct eigenmode.

$$a_0 = \frac{2\omega_1\omega_2(\xi_1\omega_2 - \xi_2\omega_1)}{\omega_2^2 - \omega_1^2} \quad a_1 = \frac{2(\xi_2\omega_2 - \xi_1\omega_1)}{\omega_2^2 - \omega_1^2} \quad (2.20)$$

Three cases can be distinguished if construction of the structure's damping occurs using Rayleigh damping via the determination of two distinct principal modes ξ_1 and ξ_2 [29]. Figure 2.18 presents these cases.

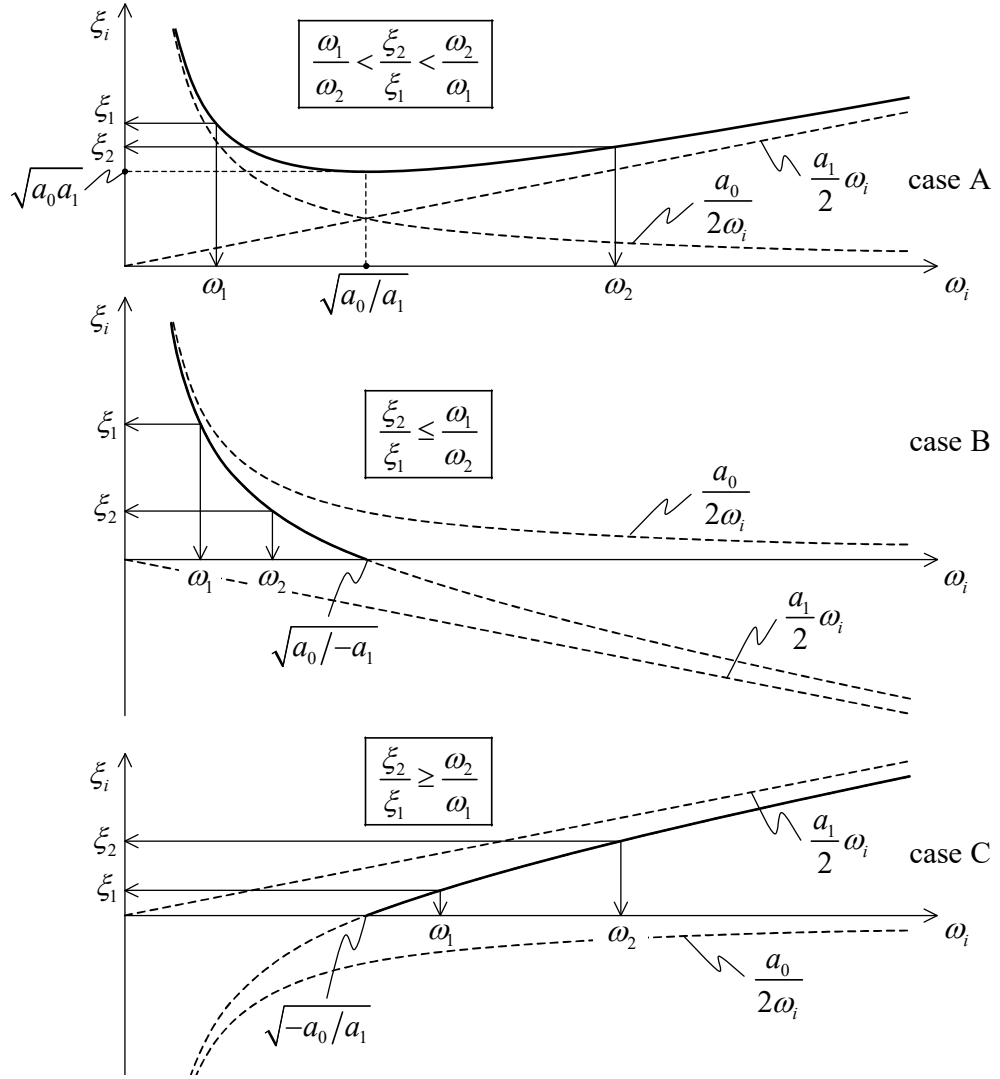


Figure 2.18: Three cases of Rayleigh damping to be defined by the use of two distinct damping values ξ_1 and ξ_2 [29]

Case A : $\left[\frac{\omega_1}{\omega_2} < \frac{\xi_2}{\xi_1} < \frac{\omega_2}{\omega_1} \right]$

All values of ξ_i are positive, meaning no problems will arise due to negative damping (amplifications). Of course the condition holds that $\xi_i \leq 1$

Case B : $\left[\frac{\xi_2}{\xi_1} < \frac{\omega_2}{\omega_1} \right]$

This may cause problems for higher frequencies since the damping ratio becomes negative for natural frequencies $\omega_i > \sqrt{a_0/-a_1}$, this requires careful consideration, as it leads to physically unacceptable results.

Case C : $\left[\frac{\xi_2}{\xi_1} > \frac{\omega_2}{\omega_1} \right]$

The last case mainly causes problems for lower natural frequencies, since the damping ratio is negative for $\omega_i < \sqrt{-a_0/a_1}$. In footbridge design, low-frequency content is of the essence since step frequencies are identified within a range of $1.0 < f_s < 2.5$ Hz. This makes case C an unviable case and should be avoided in damping assessment completely.

Caughey damping

When there is more than two modes within the critical range with the same modal damping properties, Caughey damping as an extension of Rayleigh damping can be applied for more accurate damping values at each mode. Caughey damping c is determined for N modes by:

$$\mathbf{C} = \mathbf{M} \sum_{l=0}^{N-1} a_l [\mathbf{M}^{-1} \mathbf{K}]^l \quad (2.21)$$

where:

$a_l = \text{Constants}$

Figure 2.19 explains the concept of Caughey damping for multiple modes graphically. It can be observed that the damping matrix for a specified modal damping ratio ξ for three and four modes is shown. Important to note, is that the contribution of higher modes than specified lead to either an underestimation (substantially increased damping), or unrealistic behaviour in which (potential) negative damping values lead to a growth of free vibrations in time which is far from realistic behaviour in conventional situations.

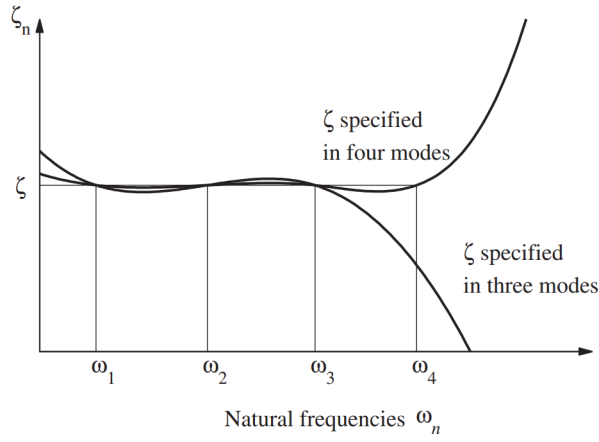


Figure 2.19: [28]

Important to note is that Rayleigh damping is obtained when $N = 2$ modes are considered, as is shown below:

$$\text{for } N=2: \quad a_0 \mathbf{M} (\mathbf{M}^{-1} \mathbf{K})^0 = a_0 \mathbf{M}, \quad a_1 \mathbf{M} (\mathbf{M}^{-1} \mathbf{K})^1 = a_1 \mathbf{K}$$

$$\mathbf{C} = a_0 \mathbf{M} + a_1 \mathbf{K}$$

Non-classical damping

Whenever a system is defined by subsystems with ranging damping properties, modal analysis with a classical damping matrix cannot be performed since the principle of orthogonality no longer holds and modal coupling arises. The solution for the complex eigenvalue problem is given by [28]:

$$\underline{u}(t) = \underline{\Psi} e^{\underline{\lambda} t} \quad (2.22)$$

Which leads to the quadratic eigenvalue problem:

$$(\underline{\lambda}^2 \mathbf{M} + \underline{\lambda} \mathbf{C} + \mathbf{K}) \underline{\Psi} = \underline{0} \quad (2.23)$$

A complex-conjugate pair of eigenvalues denoted by λ_n and $\bar{\lambda}_n$, may be expressed in the same form as for classically damped systems:

$$\lambda_n, \bar{\lambda}_n = \xi_n \omega_n \pm i \omega_n D \quad (2.24)$$

Note that ω_n and ξ_n are related to the eigenvalues as follows:

$$\omega_n = |\lambda_n| \quad \xi_n = -\frac{\text{Re}(\lambda_n)}{|\lambda_n|} \quad (2.25)$$

The associated pair of complex-valued eigenvectors is separated into its real and imaginary parts:

$$\psi_n, \bar{\psi}_n = \phi_n \pm i\chi_n \quad (2.26)$$

It is important to note that for classical damping matrices, the eigenvectors are real-valued and equal to those of the associated undamped system i.e. $\chi_n = 0$, $\psi_n = \bar{\psi}_n = \phi_n$. Whereas non-classical damping contains an imaginary component for the eigenvectors. In most cases, modal analysis is deviated from for non-classical damping since it requires complex expression evaluation due the systems remaining coupled. Numerical time integration is often utilised as an alternative with methods such as Newmark-Beta or Wilson- θ . The process to obtain the damping matrix is by use of subsystem assembly, in which the matrix is constructed by directly assembling damping matrices of the classically defined subsystems. Figure 2.20 gives a graphical representation of this assembly methodology.

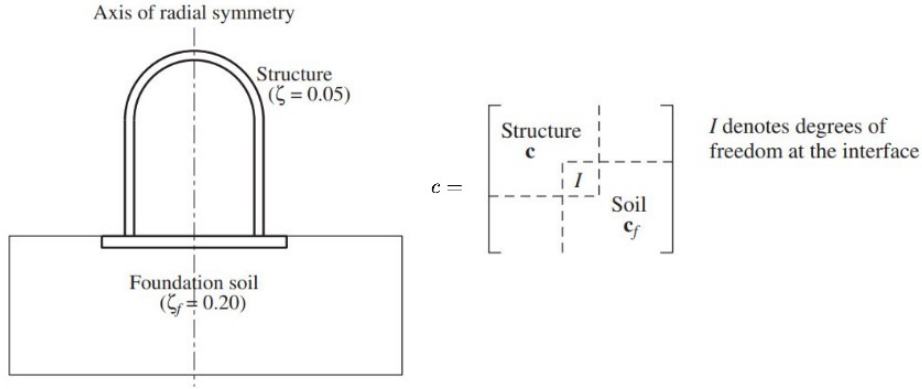


Figure 2.20: Example of subsystem assembly to account for soil-structure-interaction of different modal damping ratios [28]

2.5 Dynamic load models

According to codes and guidelines, dynamic load models are required for:

- Individual or group of hikers/pedestrians
- Individual or group of joggers
- Pedestrian streams (crowds)
- Vandals causing intentional excitation

The load models listed here should be applied as part of the procedure outlined in paragraph 2.4 Design Steps. Given their significant impact on the assessment, additional context is provided in this section. Pedestrian-induced forces relevant to footbridge dynamic excitation were comprehensively studied by Wheeler [30]. The study presented the dependence of many walking parameters, such as step length, moving velocity, peak force and contact time as a function of the pacing frequency [14]. Figure 2.21 shows the typical vertical force patterns for different types of human activities from a single footstep. Where the dynamic load factor (DLF), i.e. force divided by static weight, is depicted for various pacing frequencies (activities). For lower step frequencies, two force speaks and a trough are observed, whereas the transition to running changes the shape function, converging to a single peak.

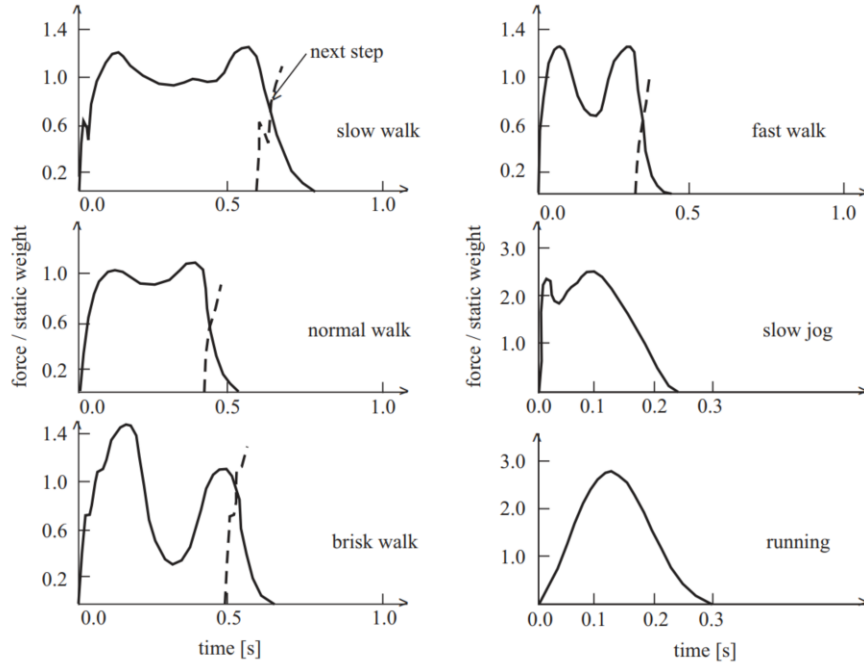


Figure 2.21: Typical vertical force patterns for different types of human activities from a single footstep [30]

All parameters differ for each pedestrian, but the study draws general conclusions, stating that with increasing step frequency the velocity, stride length and DLF increase whereas contact time decreases. These conclusions are also shown in figure 2.22.

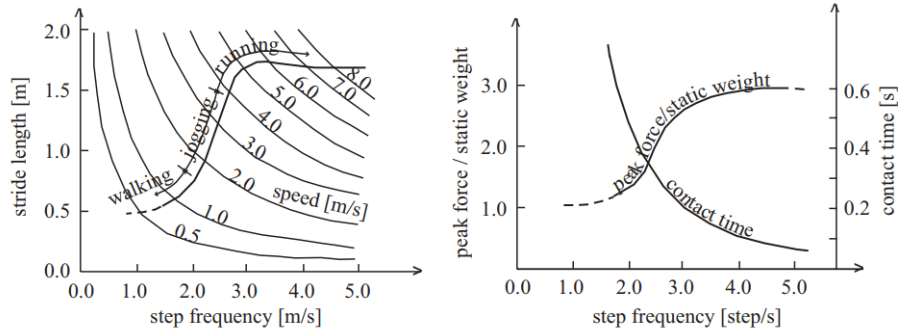


Figure 2.22: Dependence of stride length, contact time and DLF for different step frequencies [30]

Additional to vertical pedestrian-induced forces, lateral and longitudinal forces are present. As stated by Andriacchi et al. [31]: "general shapes for continuous forces in all three directions can be constructed if their perfect periodicity is assumed." Figure 2.23 shows these pedestrian-induced force shapes in vertical, longitudinal and lateral directions.

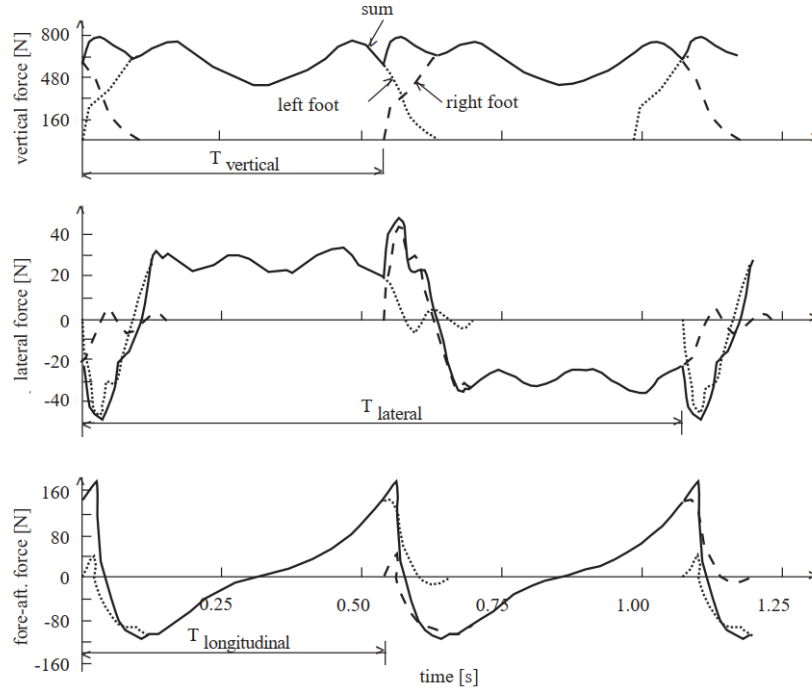


Figure 2.23: Periodic walking time histories in vertical, lateral and longitudinal directions. [31]

2.5.1 Step force model

In order to effectively model these step forces, a periodic function is constructed by means of a Fourier series. A combination of harmonic functions is used to simulate the step force induced by pedestrians, see equation 2.27. The Fourier coefficients α_i can be interpreted as a DLF for every i^{th} harmonic. In literature, many studies have been conducted to determine the Fourier coefficients for walking and running. Table 2.9 shows the coefficients found in studies.

$$\begin{aligned}
 F_{p.\text{vert}}(t, v) &= P \left[1 + \sum_{i=1}^n \alpha_{i.\text{vert}} \sin(2 \pi i l f_s t - \phi_i) \right] \delta(x - vt) \\
 F_{p.\text{lat}}(t, v) &= P \sum_{i=1}^n \alpha_{i.\text{lat}} \sin(\pi i f_s t - \phi_i) \delta(x - vt) \\
 F_{p.\text{long}}(t, v) &= P \sum_{i=1}^n \alpha_{i.\text{long}} \sin(2 \pi i f_s t - \phi_i) \delta(x - vt)
 \end{aligned} \tag{2.27}$$

where:

- $F_{p.j}$ = periodic force due to walking or running in j^{th} direction (vertical, lateral or longitudinal)
- P = pedestrian's weight (assumed to be between 700 to 800 N)
- $\alpha_{i,j}$ = fourier coefficient of the i^{th} harmonic for vertical, lateral and longitudinal forces
- f_s = step frequency
- ϕ_i = phase shift of the i^{th} harmonic
- n = total number of contributing harmonics
- v = stepping velocity
- $\delta(x)$ = dirac delta function

Author(s)	Fourier coefficients / Phase angles	Comments	Type of activity and load direction
Blanchard et al.	$\alpha_1 = 0.257$		Walking – vertical
Bachmann & Ammann	$\alpha_1 = 0.4 - 0.5; \alpha_2 = \alpha_3 = 0.1$	$f_p = 2.0 - 2.4$ Hz	Walking – vertical
Schulze	$\alpha_1 = 0.37; \alpha_2 = 0.10; \alpha_3 = 0.12; \alpha_4 = 0.04; \alpha_5 = 0.015$	$f_p = 2.0$ Hz	Walking – vertical
Bachmann et al.	$\alpha_1 = 0.4/0.5; \alpha_2 = \alpha_3 = 0.1$ $\alpha_1 = \alpha_2 = \alpha_3 = 0.1$ $\alpha_1/2 = 0.1; \alpha_1 = 0.2; \alpha_2 = 0.1$ $\alpha_1 = 1.6; \alpha_2 = 0.7; \alpha_3 = 0.3$ $\phi_2 = \phi_3 = \pi/2$	$f_p = 2.0/2.4$ Hz $f_p = 2.0$ Hz $f_p = 2.0$ Hz $f_p = 2.0 - 3.0$ Hz	Walking – vertical Walking – lateral Walking – longitudinal Running – vertical Walking – vertical & lateral
Kerr	$\alpha_1, \alpha_2 = 0.07; \alpha_3 = 0.2$	α_1 is frequency dependent	Walking – vertical
Young	$\alpha_1 = 0.37(f_p - 0.95) \leq 0.5$ $\alpha_2 = 0.054 + 0.0088f_p$ $\alpha_3 = 0.026 + 0.015f_p$ $\alpha_4 = 0.01 + 0.0204f_p$	Mean values for Fourier coefficients	Walking – vertical
Charles & Hoorpah	$\alpha_1 = 0.4$ $\alpha_1 = 0.05$ $\alpha_1 = 0.2$		Walking – vertical Walking – lateral Walking – longitudinal
EC5, DIN1074	$\alpha_1 = 0.4; \alpha_2 = 0.2$ $\alpha_1 = \alpha_2 = 0.1$ $\alpha_1 = 1.2$		Walking – vertical Walking – lateral Jogging – vertical

Table 2.9: Fourier coefficients, phase angles as proposed by studies found in literature [5]

The step force model accurately represents the loading caused by pedestrian-structure interaction, for which account is to be made for the stochastic behaviour of pedestrians. The randomness of pedestrians can be incorporated by assuming a normally distributed step frequency f_s and a uniformly distributed phase φ_i . However, its complexity poses challenges for integration in FEM software. Implementing single-moving loads to represent pedestrian streams often results in an overly elaborate representation. To address this, harmonic loads, both moving point loads and pedestrian streams, are introduced as a simplified alternative. These harmonic loads assume an equivalent number of perfectly synchronized pedestrians n' , as explained in paragraph 2.5.2.

2.5.2 Equivalent number of pedestrians

prEN1991-2 provides a specification for each traffic class regarding the equivalent number of pedestrians n' to be considered moving across the footbridge and crowd densities for streams. Table 2.10 shows these values for each class, in which the equivalent amount of pedestrians moving across the bridge, hikers/pedestrians and joggers, can be determined via $n' = \sqrt{n}$, assuming a Poisson distribution regarding the arrival probability of pedestrians [26]. The equivalent number of pedestrians is used to represent fully synchronous behaviour, meaning the same step frequency f_s and no phase φ_i .

Traffic Class	Description	Hik. group [P]	Jog. group [P]	Ped. stream [P/m ²]
TC1	Very weak traffic	1	0	0.1
TC2	Weak traffic	2	0	0.2
TC3	Dense traffic	4	1	0.5
TC4	Very dense traffic	8	2	1.0
TC5	Excep. dense traffic	16	4	1.5

Table 2.10: Traffic classes and group sizes for harmonic load models as per prEN1991-2.

For pedestrian streams, the equivalent number of pedestrians is not equal to the square root of the total number of pedestrians, since a crowd is characterised by more random behaviour due to the interaction of pedestrians with both the bridge as well as one-another. For crowd loading, the equivalent number of pedestrians n' represents a number of perfectly synchronised people for an idealised stream. In other words, it considers a crowd with individuals having the same step frequency f_s whilst having a phase distribution $\varphi_{+/-}$ synchronous with that of the mode shape Φ_n in question.

All guidelines provide the same empirical formulas for the equivalent number of pedestrians, which have been derived from Monte Carlo simulations to incorporate the stochastic loading and randomness of crowds. The reason for using an equivalent number of pedestrians in a harmonic load model for crowds is due to its simplicity. It eliminates the use of stochastic loading whilst still incorporating crowd interactions, greatly reducing computational time. The equivalent number of pedestrians in streams is provided below, for which a distinction is made between "sparse to dense" and "very dense" crowds.

- Sparse to dense crowds ($d < 1.0 \text{ P/m}^2$): $n' = \frac{10.8\sqrt{n}\cdot\xi}{S}$
- Very dense crowds ($d \geq 1.0 \text{ P/m}^2$): $n' = \frac{1.85\sqrt{n}}{S}$

Where:

n' = equivalent number of pedestrians

d = pedestrian density P/m^2 , based on the chosen traffic class as per 2.3.1

P = Persons present on the bridge

S = area of the loaded surface (bridge deck) m^2

n = number of pedestrians ($d \cdot S$)

ξ = structural damping ratio

To clarify how these formulas have been derived, the procedure for the Monte Carlo simulations is explained. First, the total number of pedestrians n loading the bridge is determined via the pedestrian density according to the traffic class in question, as per table 2.10. The loads are modelled stationary, for which each individual load $F_i(t)$ represents a pedestrian and is assigned a normally distributed step frequency f_i and random phase φ_i . The total loading $F(t)$ is obtained by summing the individual components, without considering any specific mode shape excitation.

The maximum acceleration response which occurs for this crowd $a_{max,crowd}$ within a significant period of time, taken as the time to pass the bridge twice whilst walking at a speed of $v = 1.5 \text{ m/s}$, is obtained from the numerical model. Furthermore, an equivalent number of perfectly synchronised pedestrians n' having the same natural frequency f_{syn} being in phase $\varphi_{+/-}$ with the mode shape of the natural frequency Φ_n , is simulated for a broad range of pedestrians. The total loading $F(t)$ is obtained by summing the individual synchronised components and applying the load in the direction of the mode shape. Whenever the maximum acceleration of the random crowd response $a_{max,crowd}$ matches that of the equivalent synchronised (reduced) number of pedestrians $a_{max,syn}$, determined is how many equivalent pedestrians are synchronised n' . To get a reliable response, the characteristic value represented by 95% of the samples is selected. Figure 2.24 shows how the procedure was performed.

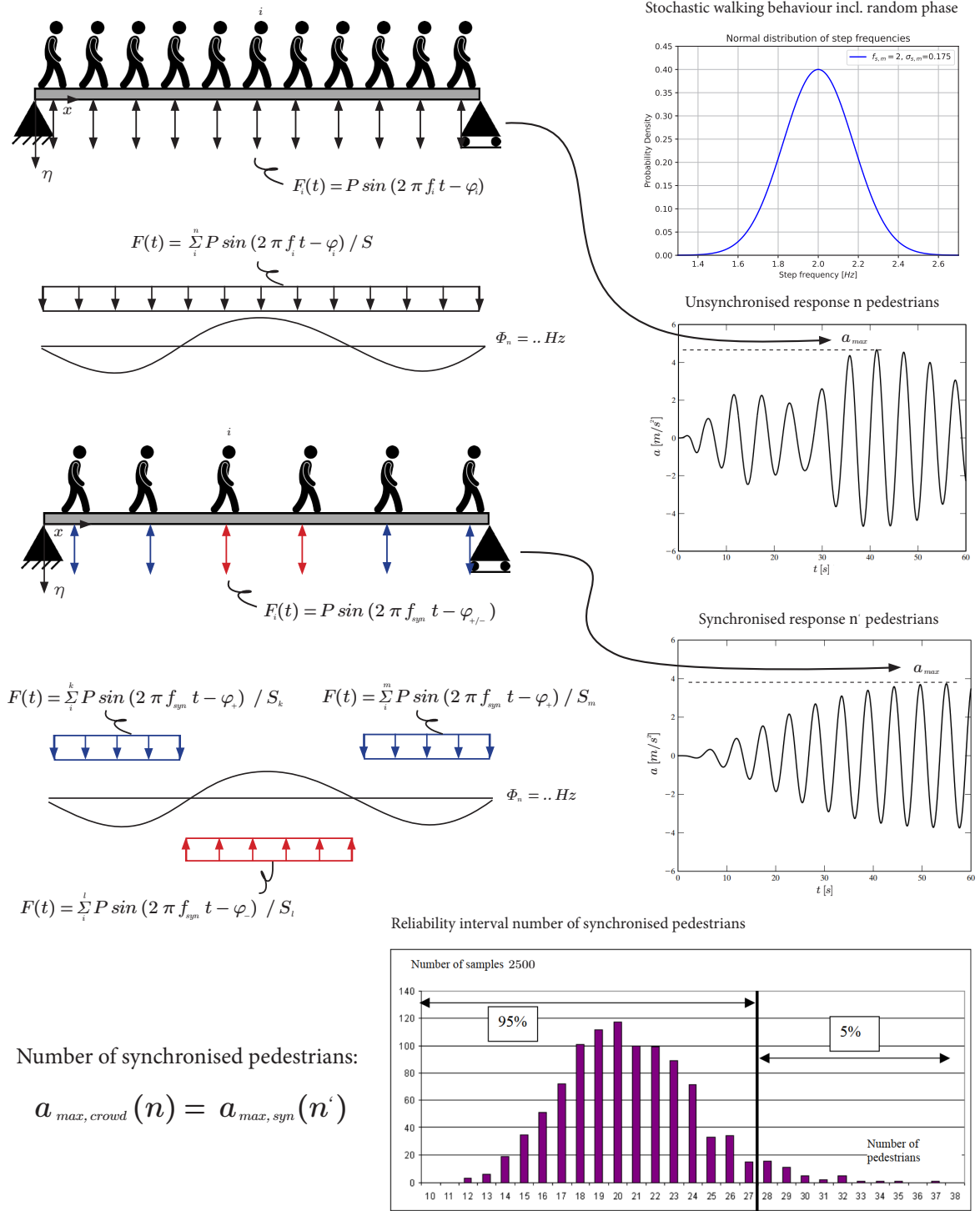


Figure 2.24: Equivalent number of pedestrians derived from Monte Carlo simulations for stochastic crowd loading

Where:

- $F_i(t)$ = Individual stationary force i for a change in time t
- $F(t)$ = Total stationary force for a change in time t
- P = Static load component of a pedestrian (assumed 800 N)
- f_i = Normally distributed step frequency $f_{s,m} = 2$ Hz, $\sigma_{s,m} = 0.175$
- φ_i = Random phase $\in [0, 2\pi]$
- S = Surface of the bridge deck
- Φ_n = Mode shape of the natural frequency of the bridge deck studied
- f_{syn} = Synchronised step frequency
- $\varphi_{+/-}$ = Phase synchronised with the mode shape Φ_n

2.5.3 Force component

The force component P_i which is present in all harmonic load models, has been studied extensively. A publication regarding footfall-induced vibrations presents a graph in which studies measured the DLF for walking of the first four footfall harmonics for varying frequencies [11]. Footfall harmonics are introduced whenever the structure's natural frequency and the pedestrian's walking pace align, for which higher harmonics are reached when this is a multiple of one-another ($r_i = f_n/f_s$, r being a whole number for every i^{th} harmonic). Observed can be a substantial scatter and greater DLF for increased step frequency, see figure 2.25.

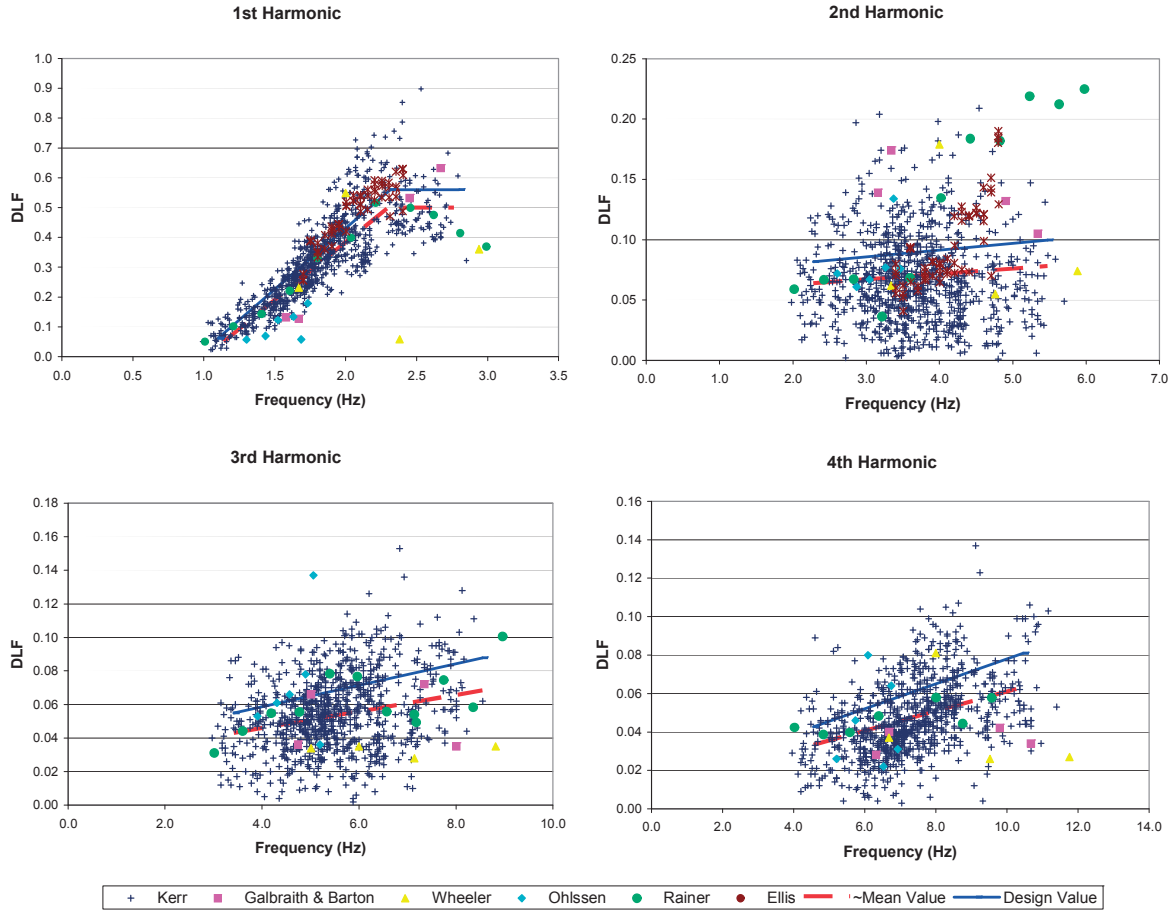


Figure 2.25: Dynamic load factor of footfall forces observed in studies [11]

Table 2.11 shows the characteristics of the force component P_i presented in codes and guidelines. Observed can be that for jogging, no longitudinal or lateral components are considered. Furthermore is only the first harmonic enlisted, leaving discussion as to why higher harmonics are being excluded. EUR23984 states that "A vertical vibration excitation by the second harmonic of pedestrian forces might take place. Until now there is no hint in the literature that significant vibration of footbridges due to the second harmonic of pedestrians has occurred. [5]"

Code / Guideline	Action	Harmonic	Type	Component P_j		
				Vertical	Longitudinal	Lateral
prEN1991-2; EUR23984; S�etra	Walking	1 st	Force [N]	280	140	35
			DLF	0.350	0.175	0.0438
	Jogging	1 st	Force [N]	1250	-	-
			DLF	1.56	-	-

Table 2.11: Characteristics of the P_j component specified in codes and guidelines

2.5.4 Reduction coefficient

The last component playing a substantial role in harmonic load models is the reduction factor ψ , which takes into account the probability that step frequency f_s and the natural frequencies f_n align. Codes and guidelines have obtained this coefficient by means of statistical analysis, simulating pedestrian streams of n pedestrians having a normally distributed step frequency f_s around the natural frequency f_n of the bridge with a random phase φ . Figures 2.26 and 2.27 show these reduction factors for walking as found in the literature. Observed can be that the S etra guideline encompasses a larger frequency range and gives no transition period before entering the second harmonic. Whereas EUR23984 / prEN1991-2021 do not imply a reduction coefficient for the second harmonic concerning lateral vibrations.

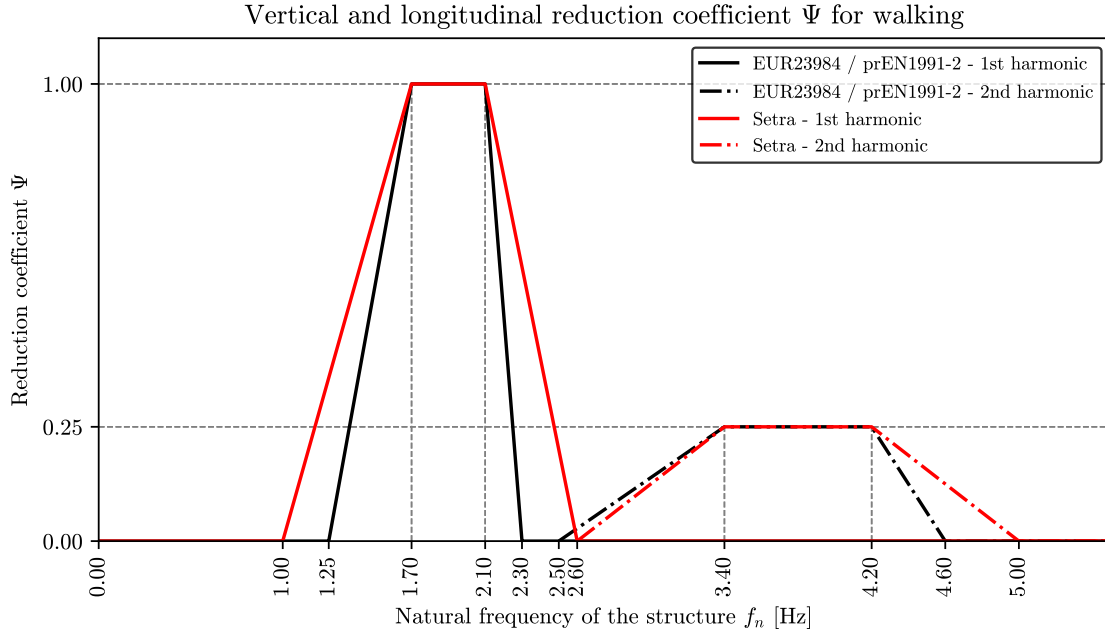


Figure 2.26: Vertical and longitudinal reduction coefficient ψ for walking according to guidelines and codes

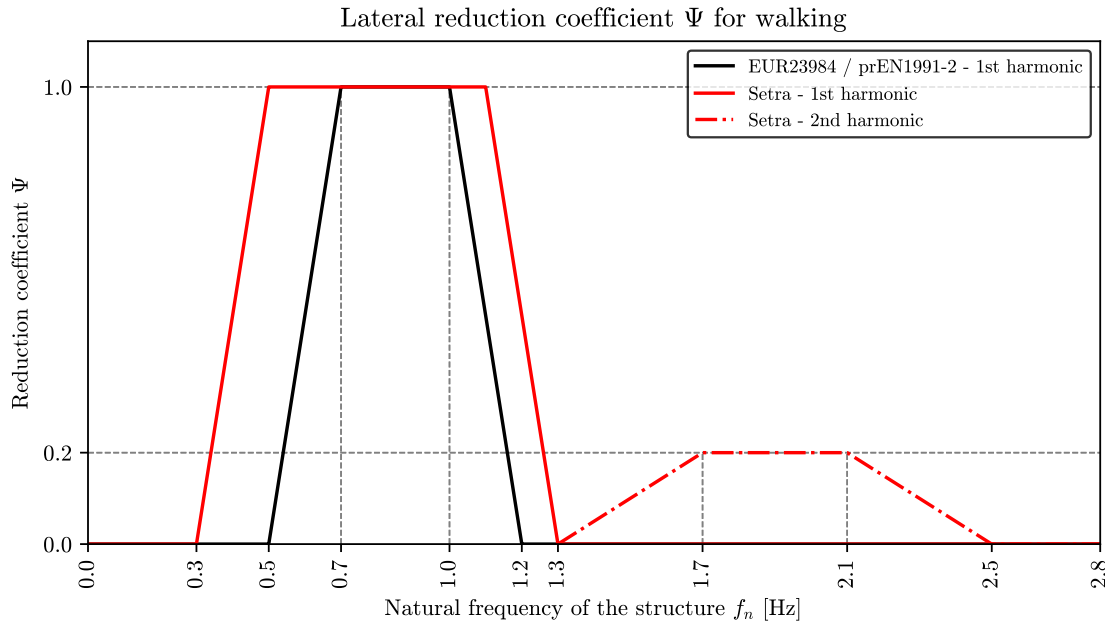


Figure 2.27: Lateral reduction coefficient ψ for walking according to guidelines and codes

Additionally, a vertical reduction coefficient for joggers is introduced, excluding the second harmonic as well as lateral and longitudinal vibrations. As the EUR23984 document states: "nevertheless, it is reasonable to suppose that the lateral component presents relatively small amplitude comparing to the vertical one". The S etra guideline labels loading by joggers 'non-relevant', giving no reduction factor altogether.

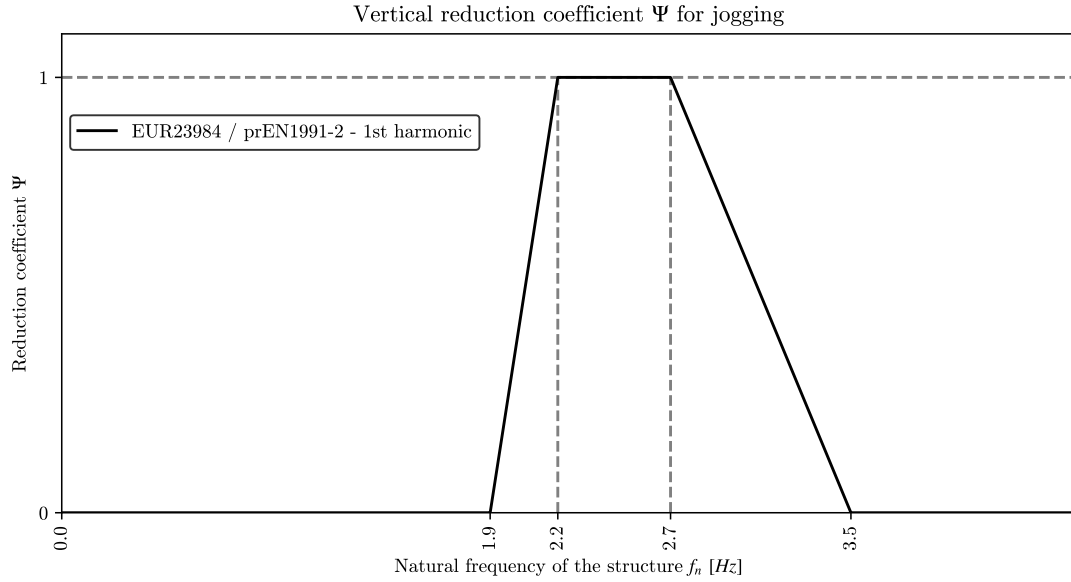


Figure 2.28: Vertical reduction coefficient ψ for jogging according to guidelines and codes

2.5.5 Moving harmonic point load model

In continuation of the elaborate periodic step-force model, a harmonic function can be applied to model pedestrian-induced forcing, utilising the concept of equivalent synchronised pedestrians. This load model concerns a single moving load which is used to simulate the behaviour of a hiker/pedestrian, jogger or small groups of both. Equation 2.28 presents the loading model, in which a walking pace v is to be assumed.

$$F_{m,i}(t, v) = P_i \cos(2\pi f_s t) n' \psi_j \delta(x - vt) \quad (2.28)$$

Where:

- $F_{m,i}$ = harmonic point load due to walking or jogging in the i^{th} direction (vertical, lateral or longitudinal)
- P_i = force component due to a single pedestrian walking in the i^{th} direction (vertical, lateral or longitudinal)
- f_s = step frequency, which is assumed equal to the footbridge natural frequency ($f_s = f_n$)
- n' = equivalent number of pedestrians ($n' = \sqrt{n}$)
- ψ_i = reduction coefficient taking into account the probability that footfall frequency approaches critical range of natural frequencies in the i^{th} direction (vertical, lateral or longitudinal)
- v = walking velocity of the activity in question (walking or jogging). $v = 1.7$ m/s walking, whereas $v = 3$ m/s for jogging.
- $\delta(x)$ = dirac delta function

2.5.6 Harmonic load model for pedestrian streams

As the S etra guideline states: "Until recently, dynamic dimensioning of footbridges was mainly based on the theoretical loading model with a single pedestrian completed by rather crude requirements concerning footbridge stiffness and natural frequency floor values. Such requirements are rather insufficient and do not cover the main problems raised by the use of footbridges in urban areas which are subject to the action of more or less dense pedestrian groups and crowds". In order to account for streams of pedestrians, a likewise harmonic model is implemented. The difference being that this load model incorporates a different number of synchronised pedestrians n' than for moving loads, see paragraph 2.5.2. The load model for pedestrian streams is given by equation 2.29.

$$F_{s,i}(t) = P_i \cos(2\pi f_s t) n' \psi_j \quad (2.29)$$

Where:

- $F_{s,i}$ = harmonic force due to pedestrian stream in the j^{th} direction (vertical, lateral or longitudinal)
- P_i = force component due to a single pedestrian walking in the j^{th} direction (vertical, lateral or longitudinal), see table 2.11 for walking
- f_s = step frequency, which is assumed equal to the footbridge natural frequency ($f_s = f_n$) under consideration
- n' = equivalent number of pedestrians, see 2.5.2
- ψ_i = reduction coefficient taking into account the probability that footfall frequency approaches critical range of natural frequencies in the j^{th} direction (vertical, lateral or longitudinal)

The harmonic load model for pedestrian streams is stationary since a constant stream of pedestrians is assumed, in which the number of people who enter and leave the bridge is equal to one-another. A distributed load is positioned such that the amplitude of the force is equal to the sign of the mode shape. Figure 2.29 shows how to apply the dynamic load model based on a given torsional mode shape.

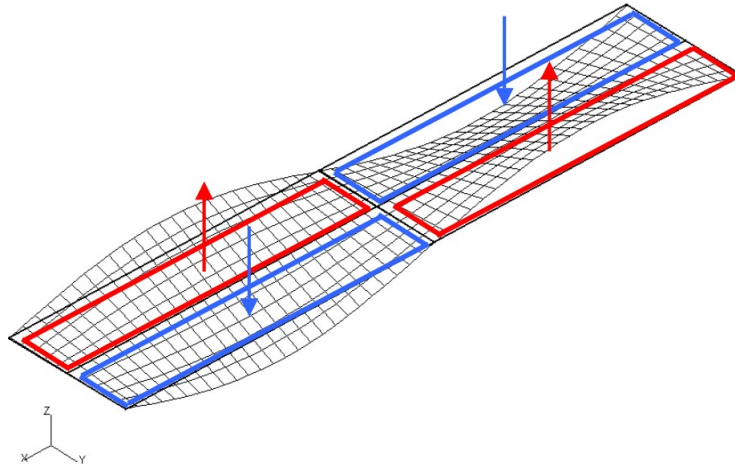


Figure 2.29: Stationary loading specifying its direction according to the given torsional mode shape [4]

Since a stationary load is applied, the loading time is not equal to the time needed to cross the bridge, such as for moving loads. The response is obtained by observing the steady-state response, meaning sufficient time is required to reach this state.

2.5.7 Harmonic load model for intentional excitation

In addition to single moving loads, groups and pedestrian streams, account should be made for intentional excitation, often referred to as "vandal loading". Eurocode prEN1991-2 [23] states that: "A footbridge should be designed such that forced vertical vibrations caused by coordinated jumping does not cause failure or damage of the bridge in ultimate limit state". Intentional excitation is to be seen as ULS rather than SLS. Studies by RWTH Aachen [6] have shown that at higher vibration amplitudes the structure's natural damping is increased, and that people often lose concentration and commitment when trying to excite a footbridge into resonance over time.

A distinction is to be made between jumping and bouncing behaviour. For which jumping is characterised by a large impact or excitation force due to a single person or synchronised group, in which their feet leave the ground and return in the next cycle. Bouncing on the other hand is explained by a persons body moving up and down via a bending motion of the knee. Additionally, bouncing can persist for a longer time, since synchronizing with the natural frequency of the bridge happens more easily when staying in contact with the structure opposed to jumping. However, all guidelines concerning footbridge design do not account for bouncing since the excitation force is less and people lose concentration and commitment over time.

There's no consistent approach in the literature for modeling intentional excitation, unlike the more standardized harmonic models for pedestrians, joggers, and streams. Some methods suggest increasing damping values, while others propose reducing them, with different dynamic load factors and varying assumptions about how many vandals to include. The methods are outlined in the sections below.

NEN-EN1991-2 (2019)

The Dutch national annex of the current Eurocode for bridge loads NEN-EN1991-2 (2019) [32] states that unless otherwise specified, verified must be that vibrations caused by this deliberate dynamic loading, do not result in exceeding the ultimate limit state. The vertical loading caused by an amount of vandals equivalent to a pedestrian stream of traffic class 5 (TC5) is applied, using the dynamic load factor of joggers being subjected to 50 % of the nominal structural damping. The placement of these vandals is at the most critical position of the bridge, based on mode shape characteristics $\Phi_n(x)$. Combining the aforementioned aspects leads to the vandal load model described below.

$$F_{vandal}(x, t) = P_{vandal} \cdot \cos(2\pi f_s t) \cdot n_{vandal} \quad (2.30)$$

Where:

$F_{vandal}(x, t)$ = harmonic point load for vandals at critical location x for a change of time t
 P_{vandal} = force component of a vandal, assumed that of joggers ($P_{vandal} = 1250$ N)
 f_s = step frequency, assumed equal to the natural frequency ($f_s = f_n$)
 n_{vandal} = number of vandals, based on TC5 ($n_{vandal} = \frac{280 \cdot n' \cdot \psi \cdot \cos(2\pi f_s t) \cdot S}{P_{vandal}}$)

prEN1991-2 (2021)

Contrary to the previous load model, the newly proposed Eurocode prEN1991-2 (2021) [10] states that depending on the location and use of the bridge, a range of two to five persons is to be assumed for intentional loading. This leaves up question to the designer since it is unclear how many vandals to consider. Larger bridges require pedestrian load class TC4 or TC5 (see Table 2.10) for the ultimate limit state design. Since no classification of large bridges is given, the implementation is once again up for debate.

Furthermore, claim is made that if no unacceptable vibrations from coordinated jumping are generated within a period of twenty seconds, the load case is to be ignored. Clearly hinting at the transient response of the structure, whereas EN1991-2 (2019) does not state a certain time period, Lastly, does this load model not account for reduced nominal damping, but rather increased values, see table 2.8 due to large vibrations caused by coordinated jumping.

S etra guideline

The last load model is provided by the S etra guideline [4], mentioning that in case of vandalism or an exceptionally large public demonstration, an accidental ultimate limit state for increased structural damping values, see table 2.8 is to be verified. Traffic class 4 is to be assumed, not translating into an equivalent number of vandals, but remaining the area load as presented in equation 2.29.

2.6 Assessment methods

To evaluate the maximum acceleration of a footbridge, several methods are available in the literature. Table 2.12 summarizes these methods, indicating whether modal analysis is required, the associated codes or guidelines, the dynamic load models utilized, and the type of vibration. The methods are organized in ascending order of complexity and computational effort. Analytical expressions, which can be solved quickly using design graphs and tables, are listed first, whereas numerical simulations, capable of incorporating stochastic loading to address uncertainties in pedestrian walking behavior, appear last to reflect their greater computational demand.

Analysis method	Modal Analysis?	Code/Guideline	Dynamic load models	Vibration type
Analytical Expressions (up to 3 span grider bridges)		BS5400, OHBDC	Moving harmonic point load	Vertical
Single Degree Of Freedom (SDOF)	x	EUR23984, S�etra	Harmonic load, pedestrian streams	All
Response spectra	x	EUR23984	Harmonic load, pedestrian streams	All
Four Footfall Harmonics	x	Footfall ind. vibrations	Moving harmonic, point load	Vertical
Finite Element Analysis (numerical simulations)	x	EUR23984, S�etra, SYNPEX	Moving harmonic point load, harmonic load pedestrian streams	All

Table 2.12: Analysis methods described in literature to determine maximum bridge deck acceleration due to pedestrian-structure interaction of footbridges

2.6.1 SDOF method

The single degree of freedom (SDOF) method for determining maximum acceleration finds application when the dynamic behaviour can be described by modal analysis. The total response of the structure is obtained by the linear combination of these modes. As the S etra guideline states: "If the function $f(t)$ is harmonic $f \sin(\omega t)$, at the frequency of one of the modes (mode j for example), then there is resonance of that mode. The response q_j of mode j is much greater than the others and the global response after a transitory period", see equation 2.31.

$$u(t) = \sum_{i=1}^N \phi_i q_i(t) \approx \phi_j q_j(t) \quad (2.31)$$

Both the EUR23984 and S etra guideline provide a formula to calculate the maximum acceleration at resonance for the SDOF system:

$$a_{max} = \frac{p^*}{m^*} \frac{\pi}{\delta} = \frac{p^*}{m^*} \frac{1}{2\xi} \quad (2.32)$$

where:

- p^* = modal (harmonic) load ($p_i^* = \int_L P \cdot \Phi_i(x) dx$)
- m^* = generalised (modal) mass ($m_i^* = \int_L M \phi_i(x) dx$)
- ξ = structural damping ratio ($\xi \approx \frac{\delta}{2\pi}$, for small ξ)
- δ = logarithmic decrement of damping

For all modes containing an eigenfrequency within the critical ranges as per specified in 2.4.2, the maximum obtained acceleration is to be verified against the comfort class specified in 2.3.2. Important to note is that unconventional superstructures can have complex modal properties, making analysis only possible through means of sophisticated FEM programs to obtain natural frequencies and mode shapes. The generalised (modal) mass m^* and modal load p^* are to be obtained from the FEM program directly, avoiding complex function evaluation.

2.6.2 Response spectra method

To avoid time domain analysis, the response spectra method can be applied by use of modal analysis to obtain the generalized (modal) mass. This method is derived from Monte Carlo simulations, and used to produce a design spectrum which incorporates stochastic forces of random pedestrian streams for various bridge geometries. Five thousand different pedestrian streams have been numerically simulated where each pedestrian has random properties for: weight, step frequency, start position and moment of first step. Considered are girder bridges with a span between twenty and two hundred meters, a width between three and five metres and four different stream densities.

The maximum acceleration is determined by:

$$a_{max,d} = k_{a,d} \cdot \sigma_a \quad (2.33)$$

Where:

$k_{a,d}$ = Peak factor: $k_{a,95\%} \cdot \psi$ ($\psi = 0.7$ according to Eurocode design practice)
 σ_a = Standard deviation of acceleration response

For which the standard deviation of the acceleration response is determined by:

$$\sigma_a^2 = k_1 \xi^{k_2} \frac{C \cdot \sigma_F^2}{m_i^{*2}} \quad (2.34)$$

Where:

k_1 = $a_1 f_i^2 + a_2 f_i + a_3$
 k_2 = $b_1 f_i^2 + b_2 f_i + b_3$
 $a_1, a_2, a_3, b_1, b_2, b_3$ = constants for either vertical or lateral accelerations based on pedestrian density
 f_i = considered natural frequency that coincides with the mean step frequency of the pedestrian stream
 ξ = structural damping ratio
 C = constant describing the maximum of the load spectrum
 $\sigma_F^2 = k_F \cdot n$ = variance of the loading (pedestrian induced forces)
 k_F = constant
 $n = d \cdot L \cdot B$ = number of pedestrians of the bridge with pedestrian density d, bridge length L and bridge width B
 m_i^* = modal mass of the considered mode i

$d [P/m^2]$	k_F	C	a_1	a_2	a_3	b_1	b_2	b_3	$k_{a,95\%}$
≤ 0.5	$1.20 \cdot 10^{-2}$	2.95	-0.07	0.60	0.075	0.003	-0.040	-1.000	3.92
1.0	$7.00 \cdot 10^{-3}$	3.70	-0.07	0.56	0.084	0.004	-0.045	-1.000	3.80
1.5	$3.34 \cdot 10^{-3}$	5.10	-0.08	0.50	0.085	0.005	-0.060	-1.005	3.74

Table 2.13: Constants for vertical accelerations [5]

$d [P/m^2]$	k_F	C	a_1	a_2	a_3	b_1	b_2	b_3	$k_{a,95\%}$
≤ 0.5	$2.85 \cdot 10^{-4}$	6.8	-0.08	0.50	0.085	0.005	-0.06	-1.005	3.77
1.0		7.9	-0.08	0.44	0.096	0.007	-0.071	-1.000	3.73
1.5		12.6	-0.07	0.31	0.120	0.009	-0.094	-1.020	3.63

Table 2.14: Constants for lateral accelerations [5]

Additionally, the guideline gives an expression to estimate the required generalised (modal) mass m_i^* for a given pedestrian density, see equation 2.35. Making modal analysis redundant, further simplifying the method.

$$m_i^* \geq \frac{\sqrt{n} (k_1 \xi^{k_2} + 1.65 k_3 \xi^{k_4})}{a_{lim}} \quad (2.35)$$

Where:

- m_i^* = modal mass for considered mode i
- n = number of pedestrians on the bridge
- k_1, k_2, k_3, k_4 = constants for vertical bending and torsion modes (see tables 2.30a and 2.30b)
- ξ = structural damping coefficient

$d [P/m^2]$	k_1	k_2	k_3	k_4	$d [P/m^2]$	k_1	k_2	k_3	k_4
≤ 0.5	0.7603		0.050		≤ 0.5	0.1205		0.012	
1.0	0.5700	0.468	0.040	0.675	1.0	0.5700	0.45	0.040	0.6405
1.5	0.4000		0.035		1.5	0.4000		0.035	

(a) Vertical bending and torsional modal mass

(b) Lateral modal mass

Figure 2.30: Constants to estimate generalised (modal) mass [5]

2.6.3 Four Footfall Harmonics

The methodology outlined in publication [11] regards the first four footfall harmonics to verify maximum accelerations, whereas other codes and guidelines often only implement effects of the first harmonic or observe a superstructure's total response using numerical analysis. The reason for using the first four harmonics is because footfall rates typically vary between 1.5 and 2.5 Hz. So any structure with modal frequencies between 1.5 and 10.5 Hz is potentially susceptible to higher responses due to resonance.

Figure 2.31 shows the peak velocities of a structure for 882 measured time history analyses in which the footfall harmonics r_i up to the twelfth harmonic are shown. Looking at the figure, the effect of higher harmonics contributes to the peak velocity showing (significant) contribution, whereas effects after the fourth footfall harmonic are considered negligible. Important to note is that this methodology only looks at vertical vibrations. Torsional modes of vibration can be quantitatively included by performing the moving load analysis at the outer edge of the deck, whereas lateral vibrations cannot be assessed utilizing this method.

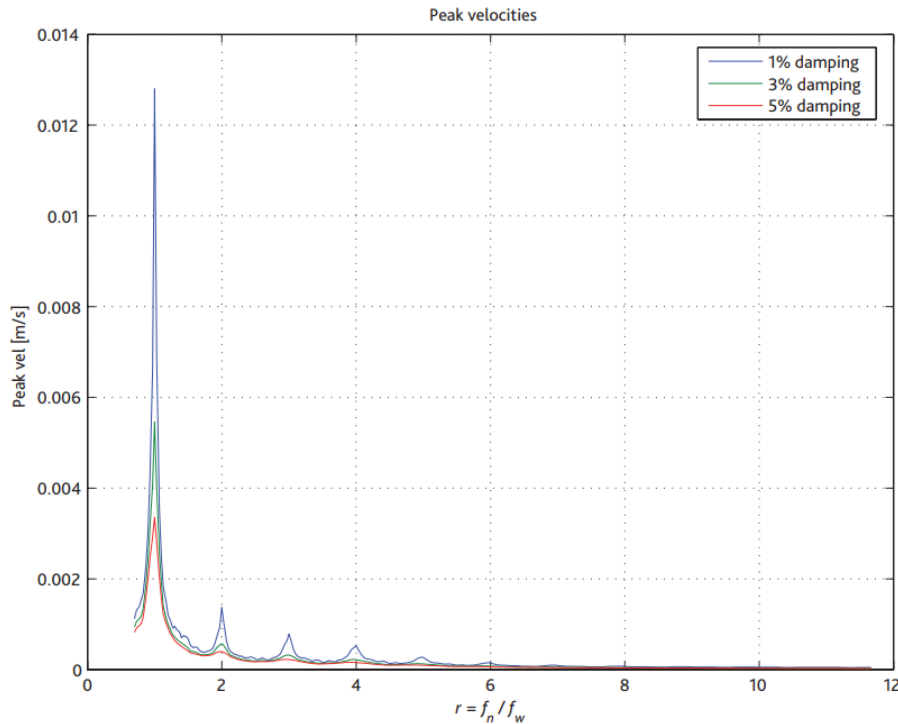


Figure 2.31: Peak velocities of the first four harmonics of footfall forces for ranging frequencies as per found in literature [11]

The publication states that structures with natural frequencies less than $f_n \leq 10 \text{ Hz}$ are potentially susceptible to resonance induced by the first four harmonics. The designer is tasked with selecting a range of walking frequencies which are suspected to occur during the lifetime of the footbridge. For this range of walking frequencies, up until the fourth harmonic is to be determined, including the belonging DLF, see table 2.15. Afterwards, the acceleration resulting from these harmonic loads is to be calculated for each vertical (or torsional) mode of vibration within range of $f_n \leq 10 \text{ Hz}$, as outlined in the procedure below via equations 2.36 / 2.40.

Harmonic number	Harmonic forcing frequency (Hz)	Design value (DLF)
1	1–2.8	$0.41(f - 0.95), > 0.56$
2	2–5.6	$0.069 + 0.0056f$
3	3–8.4	$0.033 + 0.0064f$
4	4–11.2	$0.013 + 0.0065f$

Table 2.15: Proposed dynamic load factors for the first four harmonics

Procedure Four Footfall Harmonics

1) Calculate the harmonic forcing frequency f_h for each harmonic from $h = 1$ to $h = 4$:

$$f_h = h \cdot f_w \quad (2.36)$$

Where:

f_w = Suspected walking frequency (for given range)

2) Calculate the resulting harmonic force F_h at h for each mode m within the vertical frequency range:

$$F_h = DLF \cdot P \quad (2.37)$$

Where:

DLF = Dynamic load factor, calculated by means of table 2.15

P = Static weight of a pedestrian, (assumed 700 N)

3) Determine the real $a_{real,h,m}$ and imaginary acceleration $a_{imag,h,m}$, in each mode m at harmonic h :

$$a_{real,h,m} = \left(\frac{f_h}{f_m} \right)^2 \frac{F_h \mu \rho_{h,m}}{\hat{m}_m} \frac{A_m}{A_m^2 + B_m^2} \quad (2.38)$$

$$a_{imag,h,m} = \left(\frac{f_h}{f_m} \right)^2 \frac{F_h \mu \rho_{h,m}}{\hat{m}_m} \frac{B_m}{A_m^2 + B_m^2}$$

Where:

f_m = Natural vibration of vertical mode in consideration

μ = Vertical mode shape of the mode in question (either use FEM or analytical expression)

$\rho_{h,m} = 1 - e^{-2\pi\xi_m N}$, with $N = 0.55 h \frac{L}{l}$ (L = length span, l = stride length ped.)

$A_m = 1 - \left(\frac{f_h}{f_m} \right)^2$, $B_m = 2\xi_m \frac{f_h}{f_m}$ (Constants given)

ξ_m = Suggested damping value for structure in question

4) Sum the real $a_{real,h}$ and imaginary responses $a_{imag,h}$ in all modes m and take the magnitude of this vector in order to obtain the total acceleration $|a_h|$:

$$a_{real,h} = \sum_m a_{real,h,m} \quad a_{imag,h} = \sum_m a_{imag,h,m} \quad (2.39)$$

$$|a_h| = \sqrt{a_{real,h}^2 + a_{imag,h}^2}$$

5) In order to verify the maximum acceleration $|a_h|$, a response factor R is determined. First the baseline peak acceleration $a_{r=1}$ is determined for each harmonic forcing frequency f_h by means of figure 2.32.

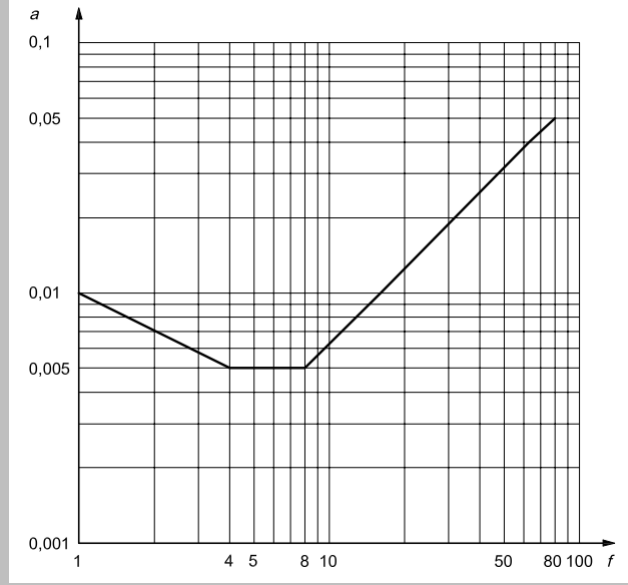


Figure 2.32: Building vibration z-axis base curve for acceleration (foot-to-head vibration direction) [12]

The response factor R is determined by:

$$R_{h=1..4} = \frac{|a_h|}{a_{R=1,h}} \quad (2.40)$$

$$R = \sqrt{R_1^2 + R_2^2 + R_3^2 + R_4^2}$$

Table 2.16, presents the response factor performance targets for footbridges, ramps and walkways. A target value of $R < 64$ is given for external bridges, excited by a single person at the most critical footfall rate. Next to this threshold for the response factor R , the analysis can be used quantitatively to see what vibration modes play a significant role in bridge deck accelerations. Additionally, the effect of higher harmonics can be observed. Figure 2.33 shows one of these analyses, used to gain insight into the eigenfrequencies and mode-shapes playing an important role in pedestrian-structure interaction.

Type	Response factor
External bridges	$R \leq 64$
Indoor bridges	$R \leq 32$
Indoor bridges (not lightweight)	$R \leq 24$

Table 2.16: Performance targets for bridges, ramps and walkways regarding the response factor [11]

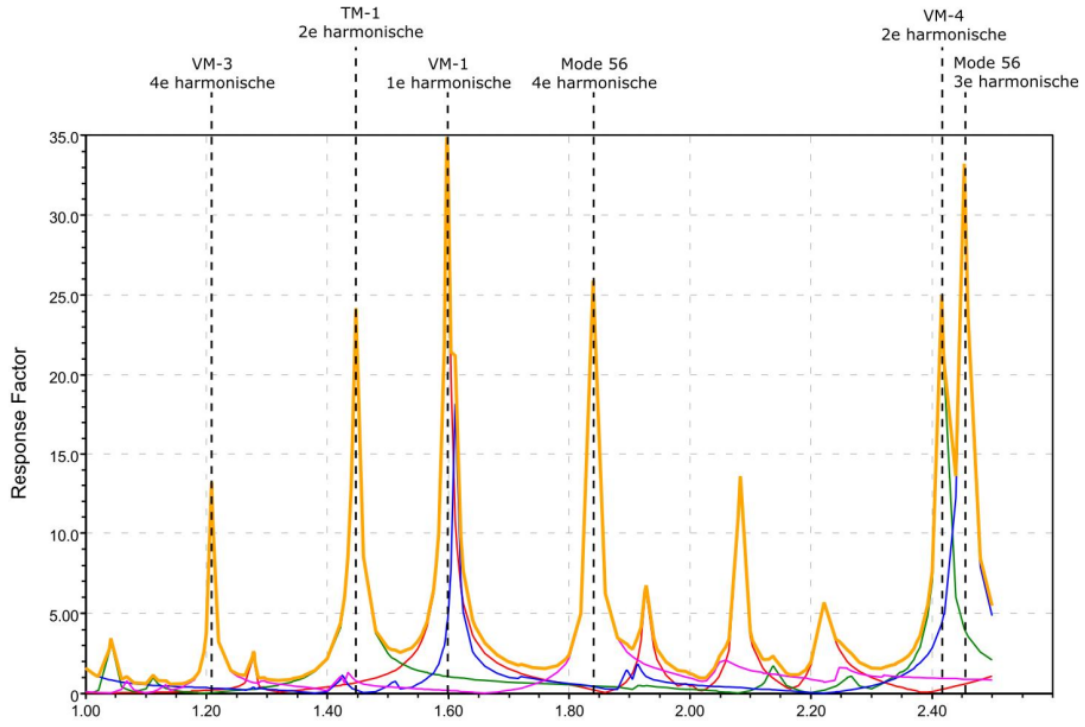


Figure 2.33: Response factors for various mode shapes and eigenvalues according to Four Footfall Harmonics [11]

2.6.4 Direct time integration method

The last and most refined method to determine maximum accelerations, is by means of numeric analysis in FEM. Numeric analysis refers to the time-dependant behaviour of footbridges subjected to dynamic loads. Whilst most programs offer modal analysis in order to acquire dynamic properties, some programs additionally offer numeric (dynamic) analysis, in which direct numerical integration is performed for each predefined time step. This can be useful for simulating highly stochastic pedestrian load models interacting with the structure in order to obtain a dynamic response.

In order to perform full transient analysis, the dynamic properties of the structure are to be known. This is firstly obtained via modal analysis, in which mode-shapes and eigenfrequencies within critical range are determined. Analysis for the undamped system is performed to obtain these values, since footbridges have low structural damping and the assumptions for lightly and undamped systems are approximately the same. Furthermore, damping is to be assigned to the model, either assigning modal damping ratio's to each eigenfrequency, or through use of Rayleigh or Caughey damping, see 2.4.4. In some cases, FEM software allow for (additional) implementation of hysteretic damping, in which a non-linear relationship between force and displacement is prescribed. The energy is lost due to internal friction within the material, whilst being independent of applied frequencies. At last, external damping can be defined, which represent devices, materials, or systems introduced specifically to reduce vibrations or oscillations.

Various load models can be constructed in ranging complexity to simulate pedestrians. Since these models are stochastic and have large uncertainty, complex modelling is mainly done by a step-by-step or step force model. The step-by-step model uses two force vectors which are jumping from spot to spot, taking into account vertical and horizontal forces induced by the feet separately. Whereas the step force model is composed of a periodic function containing multiple Fourier coefficients and phase angles, merging the ground reaction forces of both feet into one force vector sliding along a defined footpath. Both load models can be modified to consider imperfections in step frequency, oscillating around a predefined mean value. Other approaches would be to implement harmonic load models mentioned in chapter 2.5. By assuming the walking frequency to coincide with the natural frequency of the structure, resonance phenomena are being simulated and incorporated by means of a reduction coefficient ψ .

The representation of footbridge characteristics in FEM could be done with ranging complexity. Altering in element types and dimensions, mesh sizing, and the representation of certain structural systems and concepts. An example would be the simplification of a cable stayed footbridge as per figure 2.34, by

use of a single continuous equivalent beam representing the deck and springs representing the cables. Furthermore, geometric and material non-linearity can be introduced to better represent footbridge behaviour. Geometric non-linearity is mainly concerned with large deformations representing a non-linear relationship, such as the structural response of suspension bridges. Whereas material non-linearity is mainly considered in (post)-buckling behaviour for compressed areas in a the structural system. To conclude, the main goal is to accurately model the bridge's static and dynamic behaviour whilst keeping computational time low to perform analysis.

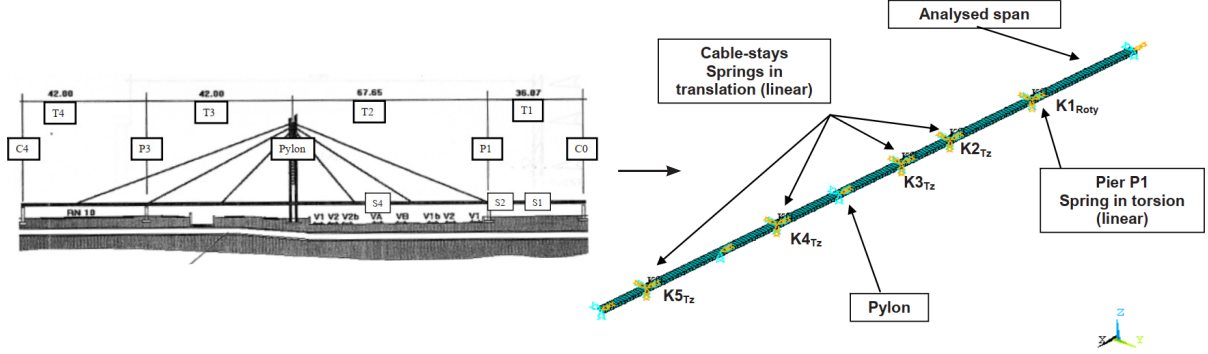


Figure 2.34: Simplified FEM modelling of a cable stayed footbridge [6]

Studying pedestrian-structure interaction can involve significant non-linear time histories, which is why most FEM software implement direct integration methods. In the section below, direct integration for both linear and non-linear systems is presented, showcasing what methods are most suited for footbridge analysis.

2.6.4.1 Direct Integration Method

Direct integration involves solving the equations of motion through a numerical, step-by-step procedure. At each predefined step, static equilibrium is evaluated. The term "direct" signifies that the equations remain in their original form and are not transformed prior numerical integration [33].

The equations of motion governing a dynamic system are given by:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) \quad (2.41)$$

It should be recalled that equation 2.41 is derived from considering static equilibrium at time t , such that:

$$\underline{F_I}(t) + \underline{F_D}(t) + \underline{F_E}(t) = \underline{R}(t) \quad (2.42)$$

Where:

$$\begin{aligned} \underline{F_I}(t) &= \text{inertia forces} \\ \underline{F_D}(t) &= \text{damping forces} \\ \underline{F_E}(t) &= \text{elastic forces} \\ \underline{R}(t) &= \text{externally applied loads} \end{aligned}$$

By solving equation 2.42 numerically for each predefined time step, the structural response is obtained. Distinction is to be made for linear and non-linear systems, and the use of implicit or explicit methods. Since the general idea and procedures of linear and nonlinear systems for direct time integration is the same, it is convenient to start with linear systems. Equation 2.42 represents a linear system since elastic forces are considered.

The difference between explicit and implicit systems is that explicit integration is directly computed from known information at the current time step. Whereas for implicit integration, the solution depends on the unknown future state. Implicit methods are more likely to be unconditionally stable, meaning that any error in displacements, velocities and accelerations at time t does not grow [34]. This stability

is often concerned with a critical step size Δt_{cr} , which is linked to the frequencies ω of interest. An explicit method requires less computational power but is often bounded by stability criteria. To give an overview of the most commonly applied direct integration methods, table 2.17 shows method, type and stability criterion.

Integration Method	Type of Method	Critical Step Size (Δt_{cr})
Newmark Method $\gamma = \frac{1}{2}, \beta = \frac{1}{6}$ (= Wilson- θ Method, for $\theta = 1$)	Implicit	$\frac{3.464}{\omega}$
Newmark Method $\gamma = \frac{1}{2}, \beta = \frac{1}{4}$	Implicit	Unconditionally stable
Newmark Method $\gamma = \frac{1}{2}, \beta = 0$ (= Central Difference Method)	Explicit	$\frac{2}{\omega}$
Wilson- θ	Implicit	Unconditionally stable when $\theta \geq 1.37$
Houbolt Method	Implicit	Unconditionally stable

Table 2.17: Comparison of direct integration methods and their stability criterion [35]

Figures 2.35 and 2.36 show the percentage of period elongation and amplitude decay for various direct integration methods respectively. What can be observed is that at higher frequencies (lower period T), period elongation and amplitude decay grows when step size Δt_{cr} is remained equal. Additionally, it can be noted that the Newmark Method with parameters $\gamma = \frac{1}{2}, \beta = \frac{1}{4}$ shows no amplitude decay, since its designed to maintain energy.

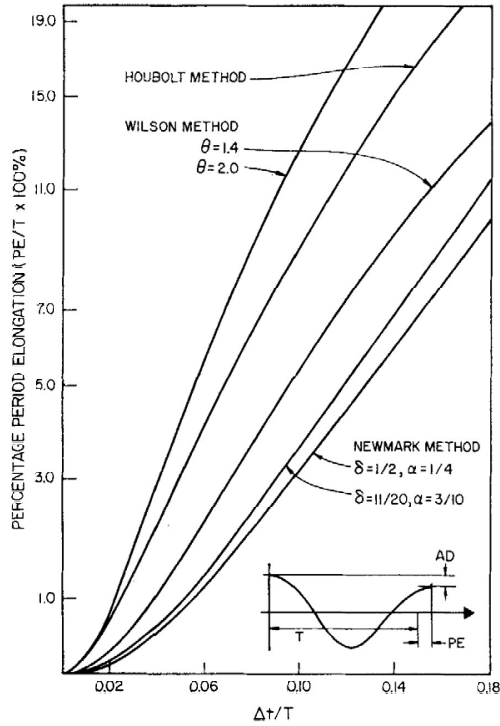


Figure 2.35: Percentage of period decay for various $\frac{\Delta t}{T}$ ratios [34]

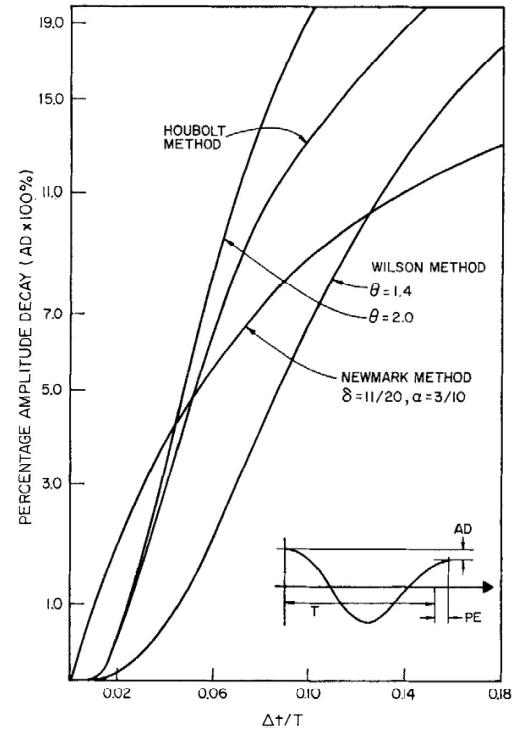


Figure 2.36: Percentage of amplitude decay for various $\frac{\Delta t}{T}$ ratios [34]

2.6.4.2 Numerical damping

The choice of direct integration method is thus heavily dependant on stability criteria, step size and frequencies of interest. In some cases, filtering out higher modes is wished for. Figure 2.37 shows the effective filtering of higher frequencies for a displacement response after one-hundred time steps. This decay in amplitude is often denoted as a form of numerical damping and can have a substantial influence on numerical results.

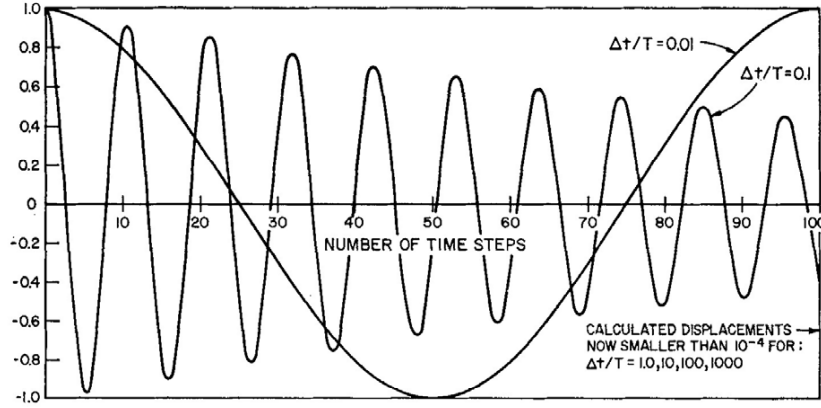


Figure 2.37: Effective filtering of higher frequencies for a displacement response after one-hundred steps, Wilson $\theta = 1.4$ [34]

In footbridge design, only the eigenfrequencies within critical range, see 2.4.2, are of interest. Therefore, it is not advisable to introduce numerical damping through amplitude decay when using direct integration methods. For this reason, the Newmark method with parameters $\gamma = \frac{1}{2}$, $\beta = \frac{1}{4}$ is considered the most appropriate choice. However, it is important to note that for large models, computational time can become excessive. In such cases, utilizing an explicit or conditionally stable implicit method with a controlled step size may, in some instances, reduce computational costs whilst still satisfying stability and numerical damping criteria. Ultimately, the engineer reviewing the numerical model must carefully evaluate the choice of method and time step.

2.6.4.3 Newmark direct integration method

The Newmark method, as mentioned previously, is unconditionally stable showing no amplitude decay for the parameters $\gamma = \frac{1}{2}$, $\beta = \frac{1}{4}$. The section below depicts how direct numerical integration is to be performed for a linear system using the Newmark Method for γ and β parameter selection [35].

Solve for each time step:

$$M\ddot{u}_{k+1} + C(\dot{\tilde{u}}_{k+1} + \ddot{u}_{k+1}\gamma\Delta t) + K(\tilde{u}_{k+1} + \dot{\tilde{u}}_{k+1}\beta\Delta t^2) = \underline{f}(t_{k+1}) \quad (2.43)$$

Predictors:

a)

$$\tilde{u}_{k+1} = u_k + \dot{u}_k\Delta t + \ddot{u}_k\left(\frac{1}{2} - \beta\right)\Delta t^2 \quad (2.44)$$

b)

$$\dot{\tilde{u}}_{k+1} = \dot{u}_k + \ddot{u}_k(1 - \gamma)\Delta t \quad (2.45)$$

Correctors:

a)

$$u_{k+1} = \tilde{u}_{k+1} + \ddot{u}_{k+1}\beta\Delta t^2 \quad (2.46)$$

b)

$$\dot{u}_{k+1} = \dot{\tilde{u}}_{k+1} + \ddot{u}_{k+1}\gamma\Delta t \quad (2.47)$$

Algorithm:

$$1. \quad \tilde{\underline{u}}_{k+1} \leftarrow \underline{u}_k + \underline{\dot{u}}_k \Delta t + \underline{\ddot{u}}_k \left(\frac{1}{2} - \beta \right) \Delta t^2 \quad (2.48)$$

$$2. \quad \tilde{\underline{\dot{u}}}_{k+1} \leftarrow \underline{\dot{u}}_k + \underline{\ddot{u}}_k (1 - \gamma) \Delta t \quad (2.49)$$

$$3. \quad \underline{\ddot{u}}_{k+1} \leftarrow (\underline{M} + \underline{C}\gamma\Delta t + \underline{K}\beta\Delta t^2)^{-1} (\underline{f}(t_{k+1}) - \underline{C}\tilde{\underline{\dot{u}}}_{k+1} - \underline{K}\tilde{\underline{u}}_{k+1}) \quad (2.50)$$

$$4. \quad \underline{u}_{k+1} \leftarrow \tilde{\underline{u}}_{k+1} + \underline{\ddot{u}}_{k+1} \beta \Delta t^2 \quad (2.51)$$

$$5. \quad \underline{\dot{u}}_{k+1} \leftarrow \tilde{\underline{\dot{u}}}_{k+1} + \underline{\ddot{u}}_{k+1} \gamma \Delta t \quad (2.52)$$

Stability:

$$a) \quad \text{Unstable if: } \gamma < \frac{1}{2} \quad (2.53)$$

$$b) \quad \text{Unconditionally stable if: } \beta = \frac{1}{4} \left(\gamma + \frac{1}{2} \right)^2 \quad (2.54)$$

$$c) \quad \text{Conditionally stable if: } \left(\gamma + \frac{1}{2} \right)^2 - 4\beta \leq \frac{4}{\omega^2 \Delta t^2} \quad (2.55)$$

Observed can be that for $\beta \neq 0$ an implicit algorithm is obtained, since the displacement \underline{u}_{k+1} becomes dependant of $\underline{\ddot{u}}_{k+1}$, see equation 2.51. Furthermore, does the selection of $\gamma = \frac{1}{2}$ allow for accuracy of order $(\Delta t)^2$. Lastly, it can be observed that for parameters $\gamma = \frac{1}{2}, \beta = \frac{1}{4}$, the algorithm is equivalent to the trapezoidal rule which makes it unconditionally stable, additionally eliminating numerical damping.

2.6.4.4 Non-linear dynamics and direct integration

Incorporating non-linear effects, such as plastic material behaviour and/or large displacements, in direct time integration methods is applicable via an iteration scheme. The obtained results are to be verified if they meet equilibrium equations and if not, are iterated until convergence is met and residual forces are below an acceptable level. The equilibrium equations for non linear dynamics can be written as:

$$\underline{M}\underline{\ddot{u}}_n + \underline{r}(\underline{u}_n, \underline{\dot{u}}_n) = \underline{f}(t_n) \quad (2.56)$$

Where:

\underline{r} = restoring force, dependant of displacement and velocity at time t
 \underline{f} = external force vector at time t

To solve for nonlinear direct integration by use of the Newmark method, the algorithm is slightly altered by incorporating a Newton-Raphson iteration scheme. Reason for outlining this iteration scheme is due to its wide applicability in FEM software. The predictor and corrector steps remain unchanged, likewise the stability conditions. Presented below is the altered process as addition to the linear methodology [35].

Solve for each time step:

$$\underline{M}\underline{\ddot{u}}_{k+1} + \underline{r}(\tilde{\underline{u}}_{k+1} + \underline{\ddot{u}}_{k+1} \beta \Delta t^2, \tilde{\underline{\dot{u}}}_{k+1} + \underline{\ddot{u}}_{k+1} \gamma \Delta t) = \underline{f}(t_{k+1}) \quad (2.57)$$

Algorithm:

$$1. \quad \underline{\ddot{u}}_{k+1} \leftarrow 0 \quad (2.58)$$

$$2. \quad \underline{u}_{k+1} \leftarrow u_k + \dot{u}_k \Delta t + \ddot{u}_k \left(\frac{1}{2} - \beta \right) \Delta t^2 + \ddot{u}_{k+1} \beta \Delta t^2 \quad (2.59)$$

$$3. \quad \dot{\underline{u}}_{k+1} \leftarrow \dot{u}_k + \ddot{u}_k (1 - \gamma) \Delta t + \ddot{u}_{k+1} \gamma \Delta t \quad (2.60)$$

$$4. \quad \varepsilon \leftarrow \underline{f}(t_{k+1}) - \underline{r}(\underline{u}_{k+1}, \dot{\underline{u}}_{k+1}) - \underline{M} \ddot{\underline{u}}_{k+1} \quad (2.61)$$

while: $\|\varepsilon\| \geq tol$ **do:**

$$5. \quad \Delta \ddot{\underline{u}}_{k+1} \leftarrow (\underline{M} + \underline{C} \gamma \Delta t + \underline{K} \beta \Delta t^2)^{-1} \varepsilon \quad (2.62)$$

$$6. \quad \ddot{\underline{u}}_{k+1} \leftarrow \ddot{u}_{k+1} + \Delta \ddot{\underline{u}}_{k+1} \quad (2.63)$$

$$7. \quad \dot{\underline{u}}_{k+1} \leftarrow \dot{u}_{k+1} + \Delta \ddot{\underline{u}}_{k+1} \gamma \Delta t \quad (2.64)$$

$$8. \quad \underline{u}_{k+1} \leftarrow u_{k+1} + \Delta \ddot{\underline{u}}_{k+1} \beta \Delta t^2 \quad (2.65)$$

$$9. \quad \varepsilon \leftarrow \underline{f}(t_{k+1}) - \underline{r}(\underline{u}_{k+1}, \dot{\underline{u}}_{k+1}) - \underline{M} \ddot{\underline{u}}_{k+1} \quad (2.66)$$

end while

Additionally, the restoring force and stiffness matrix are updated to represent nonlinear effects according to the Newton-Raphson iteration method, see below.

Restoring force $\underline{r}(\underline{u}_{k+1}, \dot{\underline{u}}_{k+1})$:

for $i = 1$ to n **do:**

$$1. \quad r_{i,k+1} \leftarrow \underline{Z}_i \underline{u}_{k+1} \quad (2.67)$$

$$2. \quad \underline{r}_{k+1} \leftarrow \underline{r}_{k+1} + \underline{Z}_i^T r_{i,k+1} \quad (2.68)$$

end for

Element stiffness \underline{K} :

for $i = 1$ to n **do:**

$$1. \quad \underline{K}_{i,k+1} \leftarrow \underline{Z}_i \underline{u}_{k+1} \quad (2.69)$$

$$2. \quad \underline{K}_{k+1} \leftarrow \underline{K}_{k+1} + \underline{Z}_i^T \underline{K}_{i,k+1} \quad (2.70)$$

end for

Figure 2.38 shows the procedure for the Newton-Raphson iteration scheme incorporated in the Newmark method. What can be observed is that the tangential stiffness Z_i^T is applied as starting point in order to reduce the error $\varepsilon_{i,k+1}$ to reach convergence.

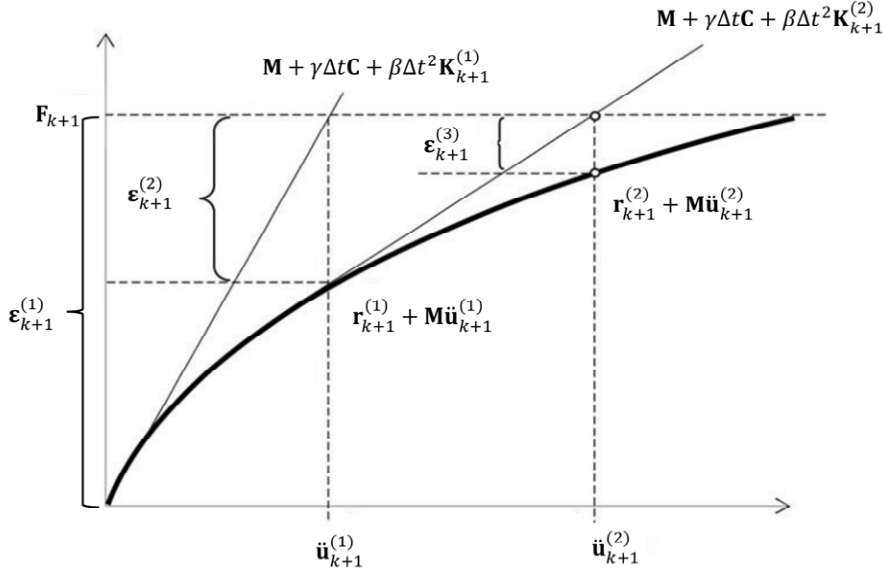


Figure 2.38: Newton-Raphson iteration scheme for Newmark Beta method [35]

2.7 External damping

External damping has various forms applied in civil engineering structures. Most notably there is three categories to be distinguished, namely: tuned mass/liquid, viscoelastic and viscous fluid dampers. Although there have been applications found of both viscous [36] and viscoelastic dampers [37] to reduce pedestrian-structure interaction in footbridges, the most widely applied method is by means of tuned mass dampers (TMD's). "Tuned mass dampers have often been selected over alternative damping devices due to their low cost, high reliability and efficiency in the mitigation of vibration under pedestrian loads. Over the last few years, research has been developed to optimize the placement and properties of multiple tuned mass dampers through deterministic approaches in order to reduce human-induced vibration in pedestrian bridges" [38]. The purpose of adding a mass damper is to limit the motion of a structure when subjected to a particular (resonant) excitation. In this section, the concept of applying a tuned mass damper to reduce pedestrian-structure interaction is outlined.

2.7.1 Den Hartog

The theory of TMD's has its root in dynamic vibration absorbers studied as early as 1909 by Frahm [39]. This was continued by the works of Den Hartog in 1950, who proposed the theory of damped absorbers attached to undamped main systems, subjected to a sinusoidal excitation [40]. The system is denoted by the second order coupled differential equations per 2.71 and 2.72. A schematic representation of the system is given by figure 2.39 .

Primary mass:

$$(1 + \mu\ddot{u}) + \omega^2 u = \frac{p}{m} - \mu\ddot{u}_d \quad (2.71)$$

Tuned mass:

$$\ddot{u}_d + 2\xi_d\omega_d\dot{u}_d + \omega_d^2 u_d = -\ddot{u} \quad (2.72)$$

Where:

$\mu = \frac{m_d}{m}$, mass ratio

$p = \hat{p} \sin(\Omega t)$, harmonic forcing

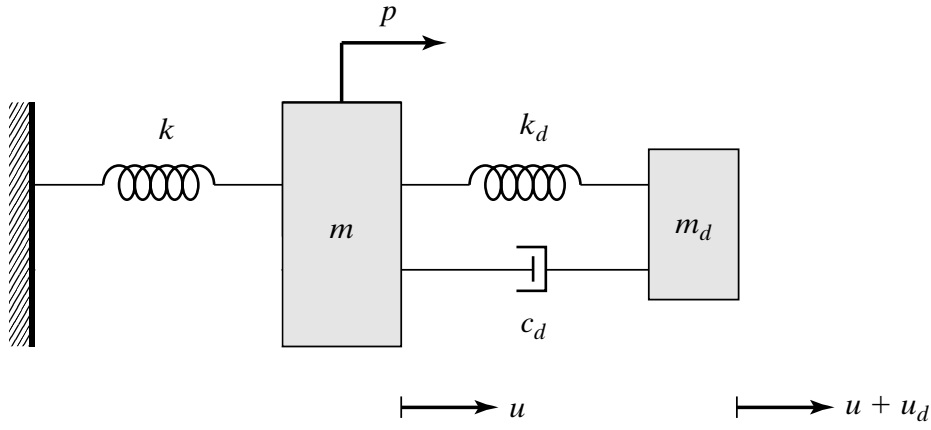


Figure 2.39: Two-mass system with undamped primary mass and TMD [41]

Since the maximum response is of interest, the steady-state solution can be designed for when applying TMD systems. Den Hartog found that the curves of the amplification factor of the primary structure intersect at two fixed points, independent of the damping ratio ξ_d of the tuned mass. This represents the "fixed point" theory, which can be observed in figure 2.40. The key design parameters are the optimum damping $\xi_{d,opt}$ and mass ratio μ .

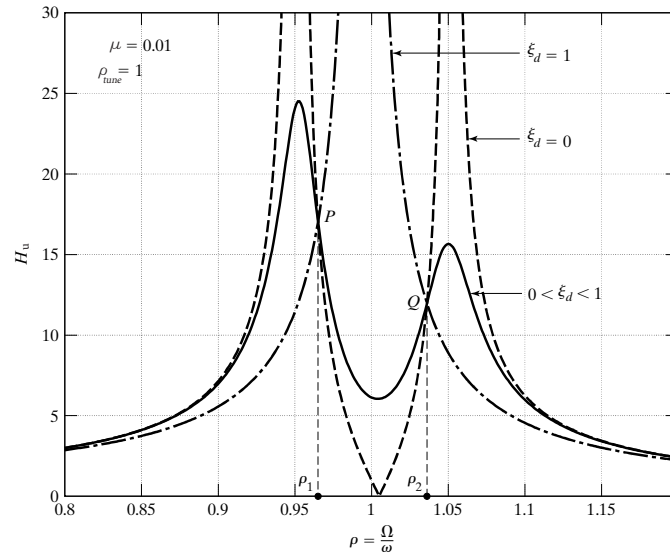


Figure 2.40: Fixed points P and Q for various damping ratio's ξ_d whilst looking at transfer function H_u of the primary mass at different forcing frequencies Ω for $\mu = 0.01$ and $\rho_{tune} = 1$ [41]

By tuning the natural frequency ratio of the damper with respect to the main system, point P and Q can be shifted up and down at $c = 0$, whereas the optimum tuning ratio ρ_{opt} is obtained whenever these points are equal, see equation 2.73. Furthermore, the optimum damping ratio $\xi_{d,opt}$ is found by ensuring a horizontal tangent through one of these points, see equation 2.74. Figure 2.41 shows the transfer function of the primary mass for a tuned damper with optimal damping ratio. Important to note is that damping values above the optimum $\xi_{d,opt}$ result in a merge of peaks and higher amplitudes, whereas below this optimum the amplitudes are once again higher.

Optimum tuning ratio

$$\rho_{opt} = \frac{1}{1 + \mu} \quad (2.73)$$

Optimum damping ratio

$$\xi_{d.opt} = \sqrt{\frac{3\mu}{(2 + 2\mu)^3}} \quad (2.74)$$

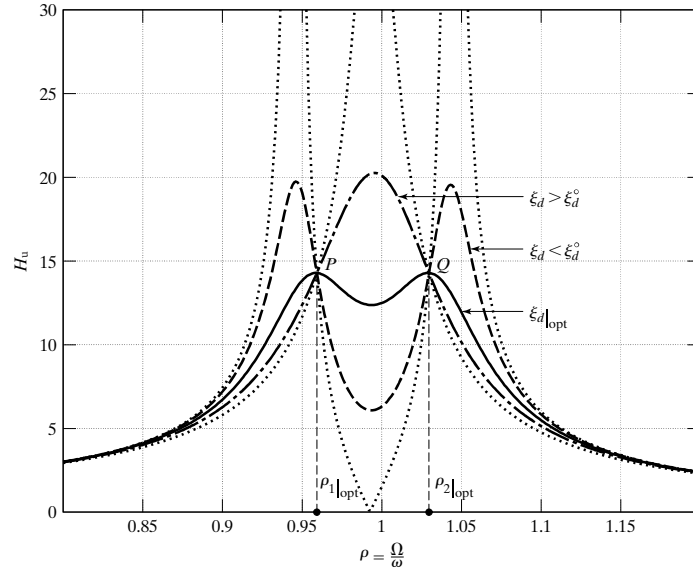


Figure 2.41: Amplification factors H_u for optimised tuning ρ_{opt} and damping ratio $\xi_{d.opt}$ [41]

2.7.2 Damping of the primary structure

In continuation of Den Hartog's work, Ghosh and Basu [42] found that for moderate damping ratio's $\xi \leq 3\%$ of the primary system, the "fixed point" theory still holds, but for a different optimal tuning frequency ratio as per equation 2.75. Since a structure always exhibits some form of structural damping, a more accurate system is acquired. It can be observed that when $\xi \rightarrow 0$, the optimal tuning frequency previously proposed by Den Hartog is once again obtained, see equation 2.73.

Improved optimum damping ratio

$$\rho_{i.opt} = \sqrt{\frac{1 - 4\xi^2 - \mu(2\xi^2 - 1)}{(1 + \mu)^3}} \quad (2.75)$$

Where:

ξ = Structural damping of the main system, assumed $\xi \leq 3\%$

Important to note is when a tuned mass is added to the structure, a new degree of freedom is introduced. The original natural frequency is reduced in amplitude by the damper and two new resonance peaks are formed, or merged to one if damping of the TMD is increased. The lower natural frequency $\rho_{1.opt}$ represents a natural frequency in which primary system and tuned mass are in-phase, whereas the upper natural frequency $\rho_{2.opt}$ depicts a natural frequency where primary structure and tuned mass move in anti-phase $\varphi = \pi$.

To further explain the concept of a TMD, the coupled second order differential equations for a two-mass fully damped system, see equation 2.76 and 2.77, is presented. Figure 2.42 gives a graphic representation of the system.

Primary mass:

$$(1 + \bar{m}\ddot{u}) + 2\xi\omega\dot{u} + \omega^2u = \frac{p}{m} - \bar{m}\ddot{u}_d \quad (2.76)$$

Tuned mass:

$$\ddot{u}_d + 2\xi_d\omega_d\dot{u}_d + \omega_d^2u_d = -\ddot{u} \quad (2.77)$$

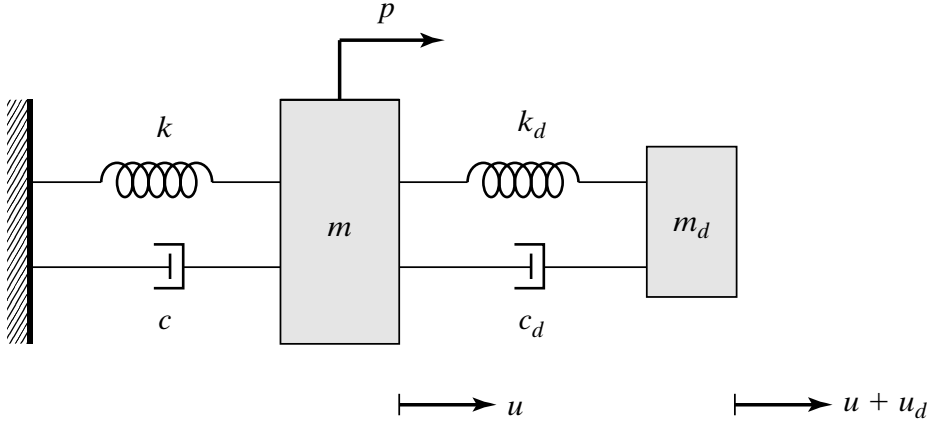


Figure 2.42: SDOF representation of primary mass and TMD for a fully damped system as proposed by Ghosh and Basu [41]

Assumed is a near-optimal approximation for the frequency of the damper where $\omega_d = \omega$. The steady-state response for the resonant condition $\Omega = \omega$ of the primary system and tuned mass is given by equations 2.78 and 2.79 respectively.

$$u = \frac{\hat{p}}{k\bar{m}} \sqrt{\frac{1}{1 + \left(\frac{2\xi}{\bar{m}} + \frac{1}{2\xi_d}\right)^2}} \sin\left(\omega t - \tan^{-1}\left[\frac{2\xi}{\bar{m}} + \frac{1}{2\xi_d}\right]\right) \quad (2.78)$$

$$u_d = \frac{1}{2\xi_d} \frac{\hat{p}}{k\bar{m}} \sqrt{\frac{1}{1 + \left(\frac{2\xi}{\bar{m}} + \frac{1}{2\xi_d}\right)^2}} \sin\left(\omega t - \tan^{-1}\left[\frac{2\xi}{\bar{m}} + \frac{1}{2\xi_d} + \frac{\pi}{2}\right]\right) \quad (2.79)$$

Observed can be that the steady-state response of the tuned mass is $\varphi = 1/2\pi$ out of phase with the response of the primary mass. This difference in phase produces energy dissipation, which is contributed by the damper inertia force. With the introduction of the improved system, an additional design parameter is introduced, namely the damping ratio of the primary structure ξ . Meaning that for design, the mass ratio μ , tuning frequency ρ_{tune} and primary structure's damping ratio ξ form the main parameters. Furthermore, practical matters are to be considered, such as the relative displacement of the tuned mass u_d , ensuring an unrestricted motion by preventing contact with the main structure. Lastly, practical limits rest on the tuned mass, guaranteeing it can be lifted into place and is resisted locally by the primary structure. Figure 2.43 gives a graphical representation of a tuned mass damper, showcasing its main components. Adjustable masses in the form of steel plates are utilised to tune the TMD to the main structure's observed natural frequency. Damping of the TMD is in the form of a viscous oil damper, tuned to the optimal damping ratio.

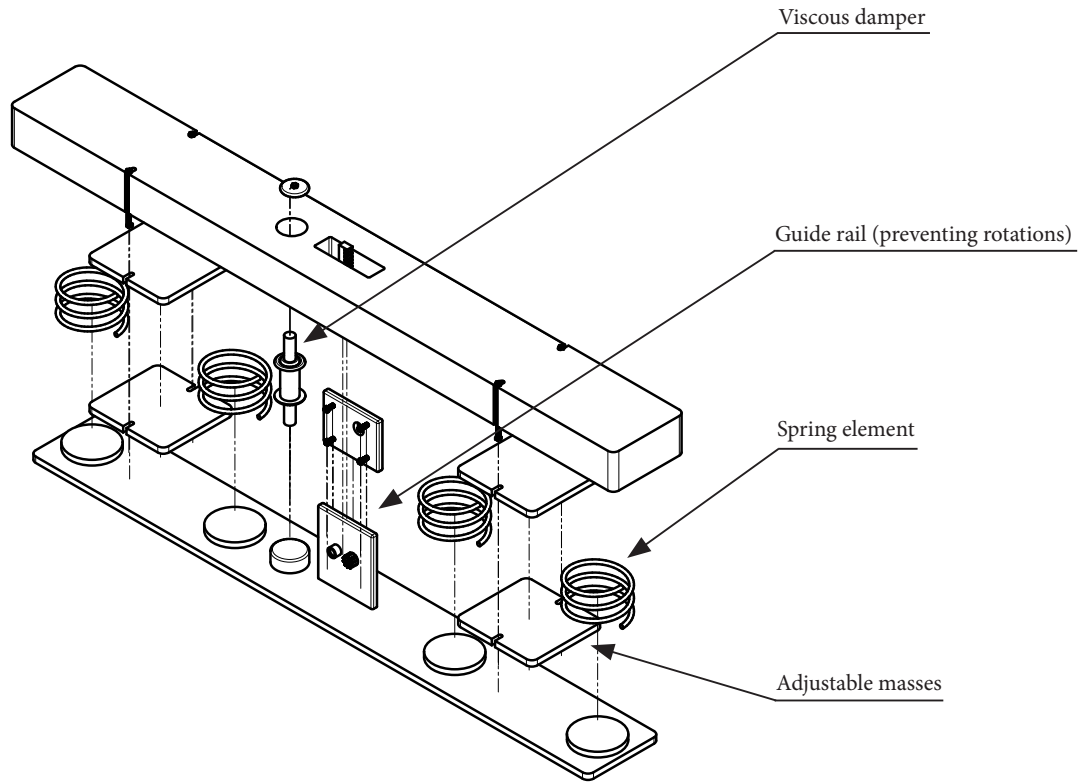


Figure 2.43: Schematic overview TMD with its main components

2.7.3 Application to footbridges

To apply TMDs in footbridge design, the theory of the two-mass system is extended. Eigenfrequencies within critical range giving a non-compliant response to serviceability (and ultimate) limits, are dampened. Through modal analysis, the properties of the frequencies of interest are determined. By equalling these modal properties to those of the primary mass, the two-mass system theory is applicable once more. It should be noted however that primarily the response of the frequency in question is dampened, not the total response. This means that if eigenfrequencies are in close range of one another, a frequency shift could mean a new (damped) resonant response of the total structure by damping out the (initial) critical mode.

Part II

Voldijk bridge - design & assessment

3

Case study design

3.1 Introduction

The case study applied in this research for the validation of assessment methods and optimisation of external damping is a self-anchored suspension bridge designed for cyclist and pedestrian traffic, see figure 3.1. The bridge is located in Tilburg, the Netherlands, and crosses the Wilhelminakanaal connecting residential neighbourhood Tuindorp de Kievit with the industrial area Vossenbergh. The bridge has a main span of 69 meters and a total length of 103 meters. It consists of three prefabricated deck elements which are welded and connected to prefab pylons using cables. The pylons are rigidly connected to the foundation and the horizontal component of the main suspension cable is (self)anchored to the girders. To prevent uplift, the main suspension cable's vertical component is anchored to the abutment. At the pylons, the bridge girders are supported via a bevelled corbel connection with a tension cable connecting both pylons to prevent buckling in the transverse direction.

After performing an eigenvalue analysis, three vertical modes and one torsional mode were obtained to be within the critical frequency range, making the structure prone to pedestrian-structure interaction. In order to prevent excessive vibrations, four tuned mass dampers (TMD's) were incorporated in the final design.

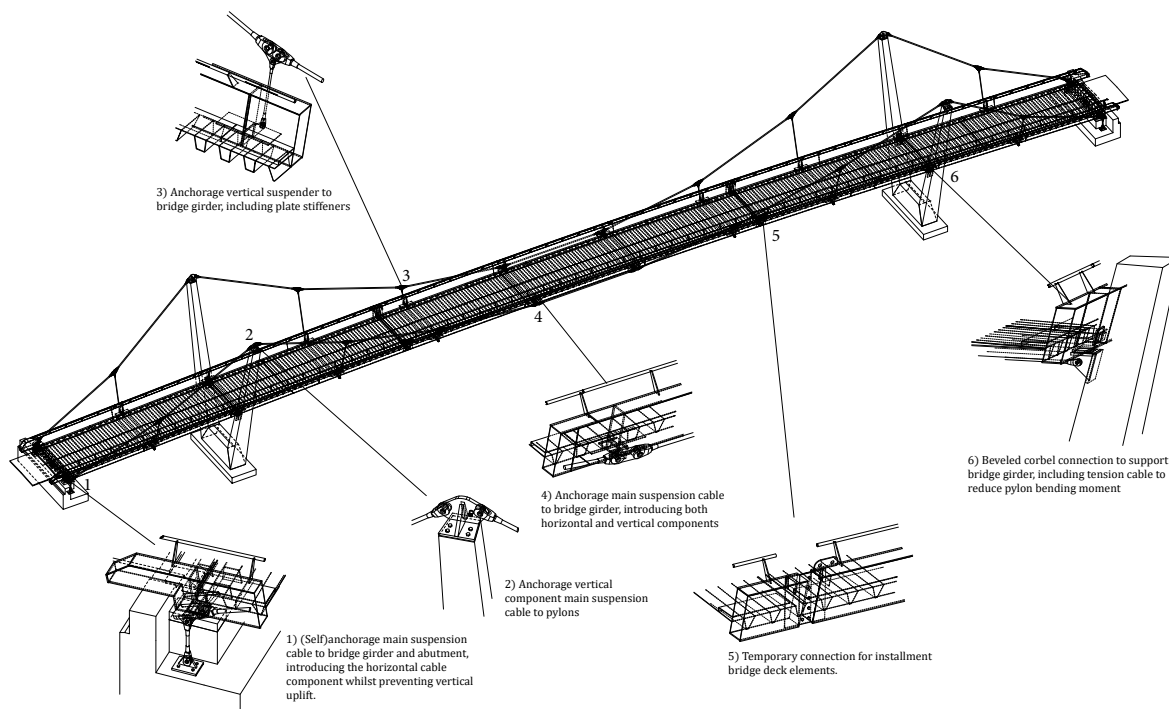


Figure 3.1: Voldijk bridge Tilburg - Graphical representation of the case study [43]

3.2 Suspension bridges

Before the structural design of the case study is presented, context is provided for suspension bridge design. The structural system has a large impact on the outcome of external damping in the context of pedestrian-structure interaction.

Suspension bridges are renowned for their ability to span long distances while maintaining structural efficiency. Their design is influenced by critical factors such as side span configuration, span continuity, and anchoring methods, each of which plays a vital role in determining the distribution of forces and overall performance. Designs feature either a self-supporting or suspended side span, identified as variant A or B respectively. A suspended side span helps reduce bending moments in the side span and hogging moments at the pylons. However, this configuration decreases the stiffness of the main span's deck, resulting in increased bending moments in the main span. Additionally, suspension bridges are classified based on whether they have continuous spans or separate main and side spans. In continuous spans, hogging moments at the pylons are reduced, but sagging moments in all spans increase. Furthermore, the horizontal reaction forces from the main span must be absorbed by the pylons and/or abutments, impacting the structural requirements of these components.

Self-anchored suspension bridges present a distinctly different configuration. The horizontal component of the main cable acts as a compressive force within the bridge deck, eliminating the need for anchor blocks. However, this design prohibits the use of separate spans due to the significant compressive forces in the deck, which can lead to local buckling issues from pronounced $P - \Delta$ effects.

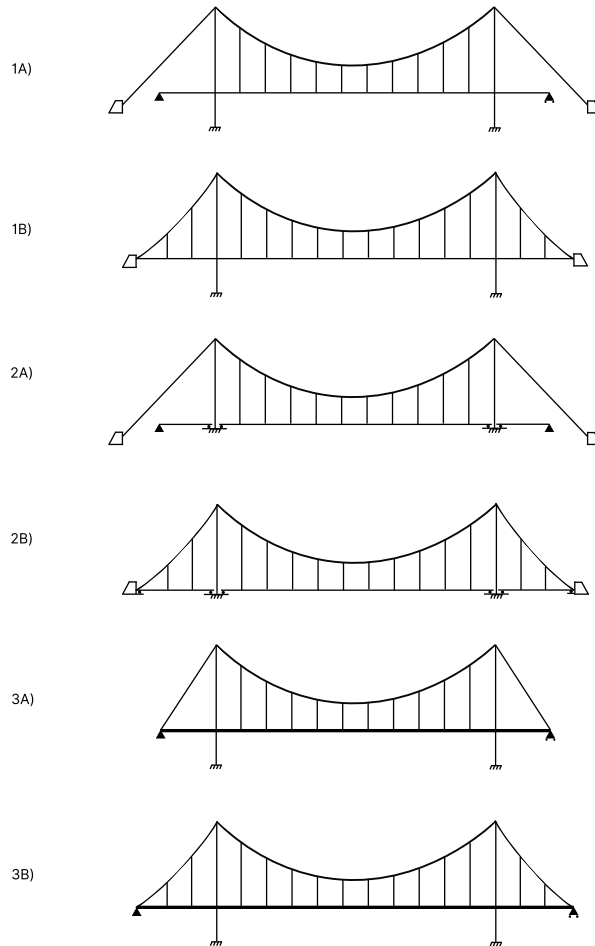


Figure 3.2: Types of suspension bridges and their structural schematics

With the insight of the various suspension bridge systems, it should be stated that the design choice of a self-anchored suspension bridge for the analysed case study mainly resulted from poor soil conditions, making it too costly to implement earth-anchorage for the main cables. Additionally, suspended side spans were chosen in order to reduced bending moments and keep one girder design throughout the entirety of the bridge.

Parallel system

The parallel system of an earth-anchored one-span suspension bridge is explained by:

$$q_c + q_g = w_L + w_D \quad (3.1)$$

Where:

- w_L = Additional loads
- w_D = Permanent loads
- q_c = Cable loads
- q_g = Girder (and deck) loads

Rewriting into the equivalent differential equation gives:

$$E_g I_g \frac{d^4 \eta}{dx^4} - (H_w + \Delta H) \frac{d^2 (y + \eta)}{dx^2} = w_L + w_D \quad (3.2)$$

Where:

- E_g, I_g = Girder bending stiffness
- H_w = Horizontal tension component of the main cable under **permanent loads**
- ΔH = Additional horizontal tension component of the main cable under **live loads**
- $y(x)$ = Vertical profile of the main cable
- $\eta(x)$ = Vertical displacement of the main girder

Resulting in a system of the final form:

$$E_g I_g \frac{d^4 \eta}{dx^4} - (H_w + \Delta H) \frac{d^2 \eta}{dx^2} = w_L - \frac{\Delta H}{H_w} w_D \quad , \text{ with : } - H_w \frac{d^2 y}{dx^2} = w_D \quad (3.3)$$

Equation 3.3 shows that live loads w_L lead to an increase in the horizontal cable force ΔH , resulting in a stiffer earth-anchored suspension bridge system. Furthermore, does the second term of the right hand side show that additional horizontal tension ΔH results in upward vertical dead load w_D .

Bending moments

To gain insight into the bending moment behaviour of the system, integrating equation 3.3 twice and applying boundary conditions $y(0) = \eta(0) = 0$, $M(0) = M_q(0)$, $y(l) = \eta(l) = 0$, $M(l) = M_q(l)$, gives the expression:

$$M(x) = M_L(x) - y(x)\Delta H - (H_w + \Delta H)\eta(x) \quad (3.4)$$

Where:

- $M(x)$ = Total bending moments in the main girder
- $M_L(x)$ = Bending moments of the main girder subjected to live loads only

Showing that bending moments resulting from live loads are reduced by cable contributions in relation to the arm of the cable $y(x)$ as well as deflections of the main girder $\eta(x)$. It should be noted that this bending moment behaviour differs from self-anchored suspension bridges, as per stated in equation 3.16. Self-anchored suspension bridges do not experience a reduction of bending moment due to deflections of the main girder $\eta(x)$, since the horizontal components H_w and ΔH are being reintroduced. Figure 3.5 is presented as clarification for the bending moments present in an earth-anchored suspension bridge.

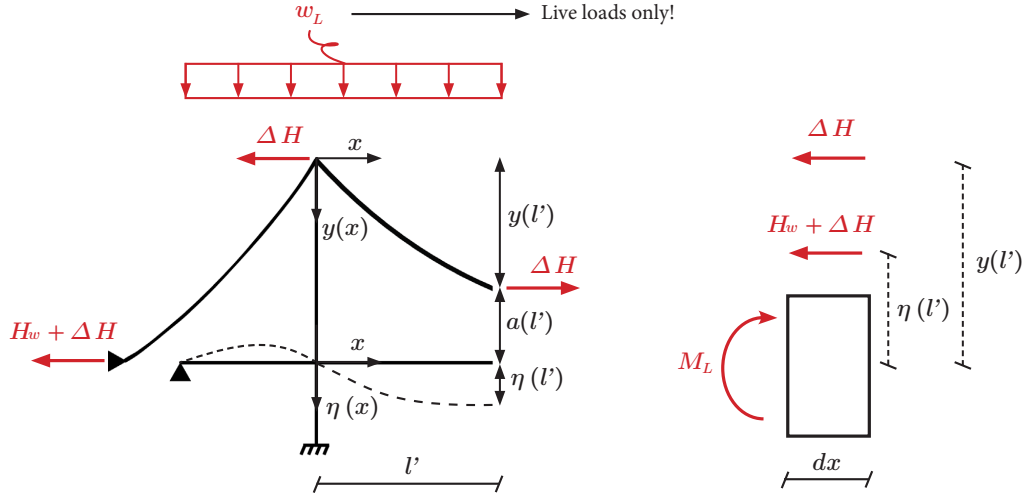


Figure 3.5: Bending moments for a main girder segment dx subjected to live loads w_L only

Axial deformation

To solve the system shown in equation 3.3, an additional equation is required due to unknowns η and H_p . The equation relates to the axial deformation of the main cable:

$$\int_{x=0}^{x=l} \frac{\Delta H L}{E_c A_c} dx = \int_{x=0}^{x=l} \frac{dy}{dx} \frac{d\eta}{dx} dx \quad (3.5)$$

Where:

E_c, A_c = Cable axial stiffness

L = Cable length along the initial parabolic shape of the main cable

Final notes

It should be emphasized that this system represents an earth-anchored suspension bridge which does not account for the horizontal main cable component as compressional force in the main girder. Furthermore, does the current system not account for any fabrication camber to prevent drainage problems and/or meet deflection criteria. What can be observed is that an increase of the additional load ΔH will result in an increased stiffness. Lastly, will a stiffer main cable likewise increase overall stiffness.

One solves for the system described in equations 3.3 and 3.5, by first determining the horizontal tension component H_w of the main cable under permanent loads, by solving the differential equation with boundary conditions $y(0) = 0$ and $y(l) = 0$. The shape of the cable $y(x)$ is obtained by assuming a central sag f at mid-span $x = 1/2 l$. This is followed by solving for the horizontal tension component ΔH of the main cable under additional loads, and the vertical displacement of the girder $\eta(x)$ whilst assuming boundary conditions $\eta(0) = 0$, $\frac{d^2\eta(0)}{dx^2} = 0$, $\eta(l) = 0$ and $\frac{d^2\eta(l)}{dx^2} = 0$.

3.3.2 Self-anchored theory

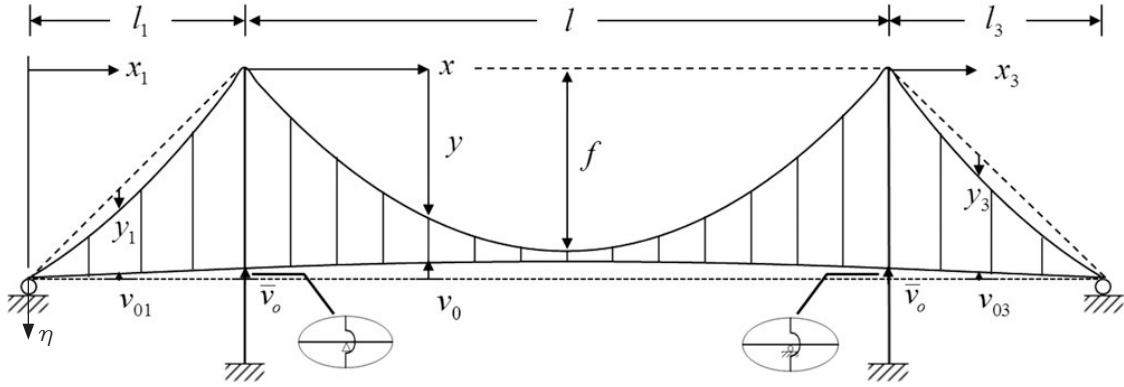


Figure 3.6: Structural system of a three span self-anchored suspension bridge including fabrication camber [44]

The parallel system of a self-anchored three-span suspension bridge with fabricated camber is once again explained by:

$$q_c + q_g = w_L + w_D \quad (3.6)$$

Solving for permanent loads

In order to better understand the parallel system of a suspension bridge, the individual components of the main cable and stiffening girder are presented for the equilibrium equation for the initial state. Continued by the derivation of the total system for permanent w_D and live loads w_L .

$$-H_w \frac{d^2 y}{dx^2} = w_m + w_h/2 + T_h \quad (3.7)$$

Where:

- H_w = Horizontal tension component main cable under permanent loads
- w_m, w_h = Uniformly distributed self-weights of the main cable and hangers
- T_h = Distributed tension of the hangers

The equilibrium equation for the initial state of the main girder is given by:

$$E_g I_g \frac{d^4 (v + v_0)}{dx^4} + H_w \frac{d^2 v}{dx^2} = w_g + w_h/2 - T_h \quad (3.8)$$

Where:

- E_g, I_g = Bending stiffness of the main girder
- $v(x)$ = Vertical deflection of the main girder, positive in downward direction
- $v_0(x)$ = Initial camber of the main girder, positive in upward direction
- w_g = Uniformly distributed self-weight of the main girder

Comparing equation 3.2 of the earth-anchored theory to equation 3.8 of the self-anchored theory, it can be observed that there is an additional contribution of the horizontal tension component H_w of the main cable related to the shape of the girder $v(x)$, due to the cable being reintroduced to the girder by terms of self anchorage. Since the horizontal tension component is denoted with a positive term on the left-hand-side, will an increase of H_w lead to a reduction of stiffness in the main girder system.

In order to minimize bending moments due to permanent loads, as well as reach a target deflection of zero, the realization should be made that the vertical profile of the main girder should equal that of the

initial camber ($v = -v_0$), meaning that the distributed tension in the hanger cables T_h is determined as follows:

$$T_h = w_g + w_h/2 + H_w \frac{d^2 v_0}{dx^2} \quad (3.9)$$

Optimised initial state

The optimised initial state for self-anchored suspension bridges under permanent loads is provided by the main cable shape and horizontal tension component. Both aspects are outlined below and will result in a balanced bending moments for the main girder with close to zero deflections. This is a continuation of the self-anchored suspension theory presented above and provides the components of interest. These components were likewise implemented for the case study.

Main cable shape

Inserting equation 3.9 into 3.7, and solving for boundary conditions $\bar{v}_0 = v_0(0) = v_0(l)$, will result in a definition for the main cable of the **central span** as:

$$y(x) = \frac{w_D}{2H_w} x(l-x) v_0(x) + \bar{v}_0 \quad (3.10)$$

Where:

- $y(x)$ = Cable shape of the central span
- $w_D = w_m + w_h + w_g$, total permanent load
- \bar{v}_0 = Camber of the main girder profile at the pylons

Solving for boundary conditions $v_{01}(0) = 0$ and $v_{01}(l_1) = \bar{v}_0$ provides the shape of the main cable for the **side spans**, such as:

$$y_{1,3}(x) = \frac{w_D}{2H_w} x_1(l_1 - x_1) - v_{01}(x_1) + \bar{v}_0 \frac{x_1}{l_1} \quad (3.11)$$

Horizontal tension component

By assuming a sag f at mid span, the horizontal tension component H_w of the main cable under permanent loads can be determined by:

$$H_w = \frac{w_D l^2}{8[f + v_0(l/2) - \bar{v}_0]} \quad (3.12)$$

Where:

- f = Sag of the parabolic cable shape at $x = l/2$
- $v_0(l/2)$ = Camber of the main girder profile at the middle of the central span

Solving for live loads

Extending the theory to incorporate live loads, requires a new definition of both main cable and main girder system. Due to the additional loads, the horizontal tension component is increased by ΔH , as well as incrementally increasing the hanger tension distribution ΔT_h . The equilibrium equation for the new main cable system is given by:

$$-(H_w + \Delta H) \left[\frac{d^2 y}{dx^2} + \frac{d^2 \eta}{dx^2} \right] = w_m + w_h/2 + T_h + \Delta T_h \quad (3.13)$$

Where:

ΔH = Additional horizontal tension of the main cable

ΔT_h = Incremental hanger tension as a result of live loads

The equilibrium equation for the main girder system subjected to additional loading is given by:

$$E_g I_g \frac{d^4 \eta}{dx^4} + (H_w + \Delta H) \left[-\frac{d^2 v_0}{dx^2} + \frac{d^2 \eta}{dx^2} \right] = w_g + w_h/2 - T_h + w_L(x) - \Delta T_h \quad (3.14)$$

Eliminating $T_h + \Delta T_h$ from equation 3.13 and 3.14, leads to a total system of:

$$E_g I_g \frac{d^4 \eta}{dx^4} = w_L(x) - w_D \frac{\Delta H}{H_w} \quad (3.15)$$

Bending moments

Integrating the previously obtained equation 3.15 twice and applying boundary conditions $M(0) = M_L(0)$, $y(0) = 0$, $M(l) = M_L(l)$, $y(l) = 0$ gives the following bending moment expression:

$$M(x) = M_L(x) - y(x)\Delta H \quad (3.16)$$

Equation 3.16 depicts the bending moment expression for self-anchored suspension bridges. It is apparent that an increase in cable tension ($H_w + \Delta H$) does not lead to a bending moment reduction of the live loads in the main girder. This is different from the bending moment expression found for earth-anchored suspension bridges, as per equation 3.4. To explain no additional bending moment reduction due to deflection of the main girder via $H_w + \Delta H$, figure 3.7 is presented.

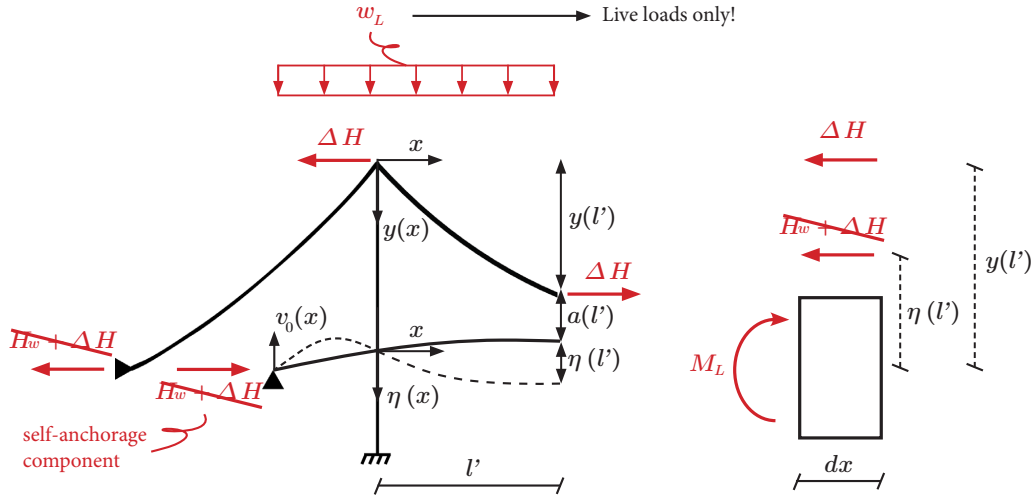


Figure 3.7: Bending moments for a main girder segment dx subjected to live loads w_L only

Axial deformation

To solve the system shown in equation 3.15, an additional equation is required once again due to unknowns η and ΔH . The equation relates to the axial deformation of the main cable:

$$\int_{x=0}^{x=l} \frac{\Delta H L}{E_c A_c} dx = \int_{x=0}^{x=l} \frac{dy}{dx} \frac{d\eta}{dx} dx \quad (3.17)$$

Solving the system of the main span is achieved by first determining the horizontal tension component

$H_{i.w}$ of the main cable under permanent loads, by solving the differential equation with boundary conditions $y(0) = 0$ and $y(l) = 0$. The shape of the cable $y(x)$ is obtained by assuming a central sag f at mid span $x = \frac{1}{2}l$. This is followed by solving for the horizontal tension component ΔH of the main cable under additional loads, and the vertical displacement of the girder $\eta(x)$ whilst assuming boundary conditions $\eta(0) = 0$, $\frac{d^2\eta_1(l)}{dx^2} = \frac{d^2\eta(0)}{dx^2}$, $\eta(l) = 0$ and $\frac{d^2\eta(l)}{dx^2} = \frac{d^2\eta_3(0)}{dx^2}$. Since the boundary conditions are linked to the side spans, solving the entire system is required and can be performed using the same approach.

Final notes

Comparing earth-anchored to self-anchored theory, one can observe that self-anchored suspension bridges have a smaller stiffness, due to the fact that the horizontal cable component is reintroduced to the deck, meaning that not additional contribution by the (horizontal) stiffness $H_w + \Delta H$ as result of main girder deflections η is obtained. This should be considered the main trade off, reducing the total system's response whilst providing better loading conditions for the foundation since the requirement of significant anchor blocks is reduced due to the horizontal cable component being absent.

3.3.3 Procedure for self-anchored suspension bridges

Structural design of a suspension bridge starts by finding an optimized initial state, in which bending moments under dead loads are minimized and a target deflection close to zero is met. This should include cable tensioning, since cables cannot be defined in a stress-free state. Many different methods exist to determine this initial shape, which in practice is mainly obtained through use of non-linear FE methods due to its high geometric non-linearity. Methods worth mentioning are the initial force method [45], target configuration under dead load (TCUD) method [46], and the unstrained length method (ULM) [47]. An iterative process is applied in order to find the initial shape, either by changing the prestress in the cables or finding the unstrained length for set targets.

Additionally, the closed-form analytical solution presented in section 3.3.2 can be utilised. In order to effectively apply the solution to a finite element model, one should consider that the initial state is obtained by use of the horizontal tension component H_w and description of the main cables shape $y_i(x)$. Meaning that the optimal shape and pretension in the cable elements is found when this horizontal component is introduced to the main cable, and the same magnitude as compression in the main girders. These stress states can not be applied to the model directly, which is why the horizontal force is introduced as pair to provide both tension and compression, as per figure 3.8. The methodology to obtain the closed-form optimised state for self-anchored suspension bridges in FEM is explained on the next page.

Methodology for the closed-form optimised state in FEM

1. Obtain the horizontal tension component H_w and cable shape $y_i(x)$ for each span from the closed-form solution.
2. Apply horizontal cable force component H_w to the ends of the main cable, for which translation in longitudinal direction is permitted (roller support).
3. The same force is applied in opposite direction to the main girder, reintroducing the tension component into the deck as compression.
4. Acquire the cable stresses of each individual cable component and store these values.
5. Create rigid links between the ends of the main cable and connection point of the main girders (remove roller support).
6. Reduce the cable stiffness close to zero^a and apply the previously stored cable stresses to each element.
7. Apply the geometry and cable stresses of the previously determined step whilst reintroducing cable stiffness back to its original value.

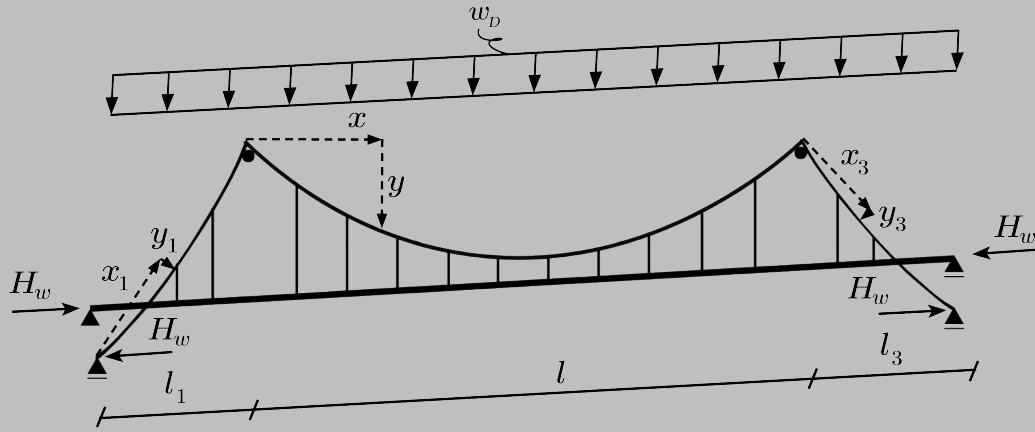


Figure 3.8: Graphic representation of steps 2 and 3 of the methodology, in which the horizontal forces of the main cables are parallel to those of the main girder, allowing for translation of the cable roller supports in longitudinal direction

^aBy reducing the stiffness of a cable element ($EA \approx 0 \text{ kN}$), equilibrium is found under pure tension, resulting in an optimised shape with minimal internal forces

3.4 Structural design

The cross section of the bridge deck is made up of steel elements, consisting of two boxed girders connected via secondary beams via troughs and a deck plate on top. This configuration allows for high torsional rigidity due to the closed form cross section of the main girder, whilst providing an arm to connect these elements via the troughs and deck. Additionally, the (shear) stability is substantial due to the 2D element by means of the steel deck connecting to the main girders. Coupler cables provide support to these girders by transferring vertical loads to the main cable which in turn transfers horizontal loads to the ends of the girders and vertical loads to the pylons. The deck is constructed out of three prefabricated elements, which are connected via bolts and afterwards welded together with steel plates to form a rigid connection. Figure 3.9 presents the cross section of the deck with the aforementioned components.

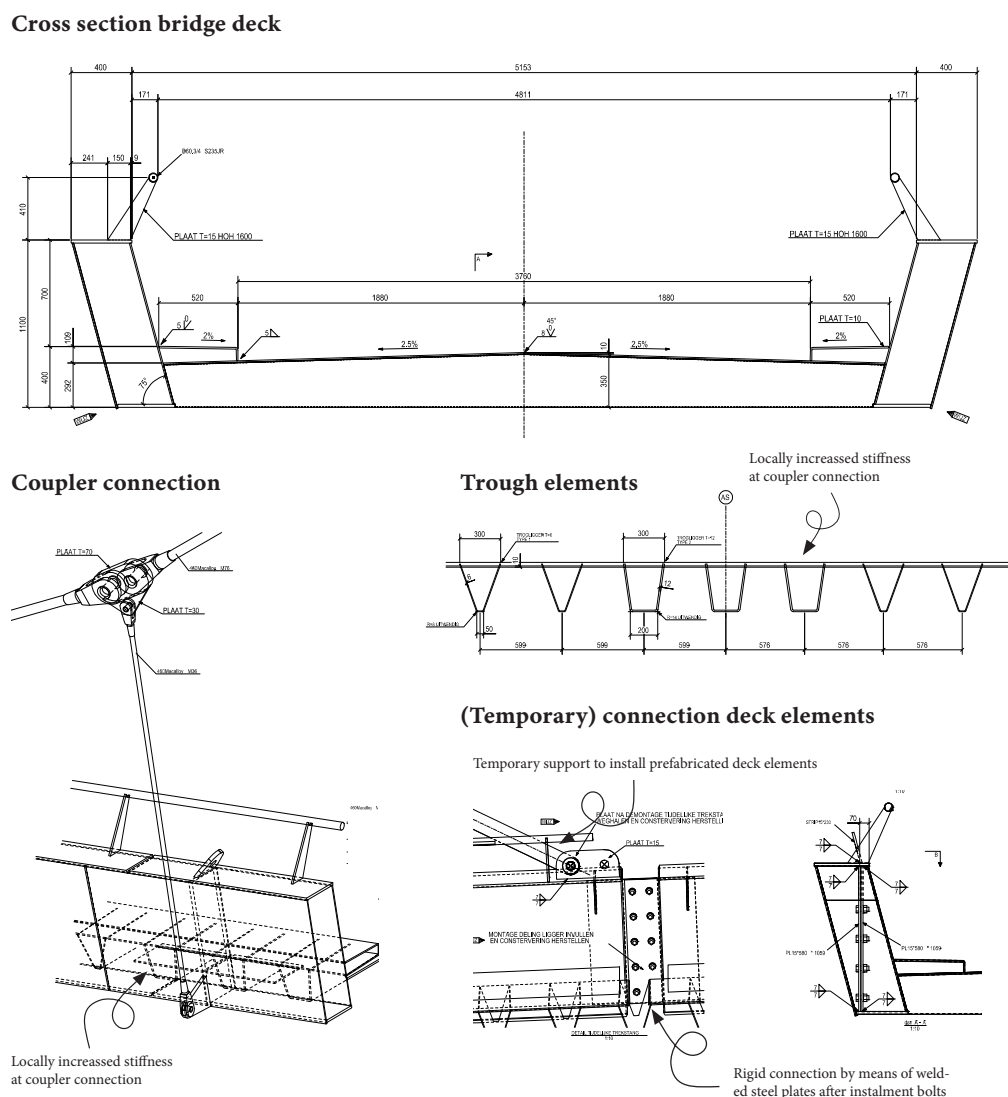


Figure 3.9: Case study representing the cross section of the deck, troughs elements, coupler and deck element connection [43]

Prefabricated reinforced concrete pylons provide an arm to effectively reduce bending moments due to the parallel system properties of a suspension bridge. The pylons have a tapered geometry and connect from abutment to the main cable system. They support the bridge deck via a bevelled corbel connection. A cable element is positioned below the corbel connecting both pylons, reducing buckling length in transverse direction since translation is prevented by either the cable (tension) or the deck as a whole (compression). In longitudinal direction the contribution of the main cable system allows for a positive effect on buckling instability of the pylons. Figure 3.10 presents the design of the pylons.

[illegible]

Supporting the bridge is done by a pinned and roller support at the abutments located at the ends of the bridge, whereas the pylons support the deck by means of a bevelled corbel connection, allowing for both rotation and translation in longitudinal direction utilising a sliding plate. The roller support consists of an expansion joint to primarily deal with temperature loads, whereas the pinned support transfers horizontal loads to the South side abutment, mainly resultant from wind, breaking forces generated by a vehicle or (dynamic) horizontal pedestrian loads. Figure 3.11 presents the connection design of the case study.

[illegible]

Technical drawing of a cross-section of a bridge structure. The drawing shows the concrete overhang profile (labeled 'CONCRETE OVERHANG PROFILE' and 'CONCRETE OVERHANG 500') and the expansion joint (labeled 'Expansion joint'). The drawing includes dimensions for the concrete overhang (500) and the expansion joint (1000). The drawing is a cross-section view, showing the internal structure of the bridge and the expansion joint.

[illegible][illegible]

59

The case study uses a self-anchored suspension bridge system, as previously mentioned. Figure 3.12 shows the force transfer of the bridge, for which elements in tension are denoted in red and elements in compression blue. Reason for applying a self-anchored suspension bridge design as opposed to an earth-anchored system is to exclude the need for large anchor blocks to effectively transfer the horizontal tension component from the main cable to the foundation. The horizontal cable component is anchored, reintroducing this force to the main girders resulting in compression forces. Additionally, the vertical component of the main cable at the ends of the bridge is anchored to the abutment. This is done to prevent the uplift of the main girder (and deck) due to increased loading.

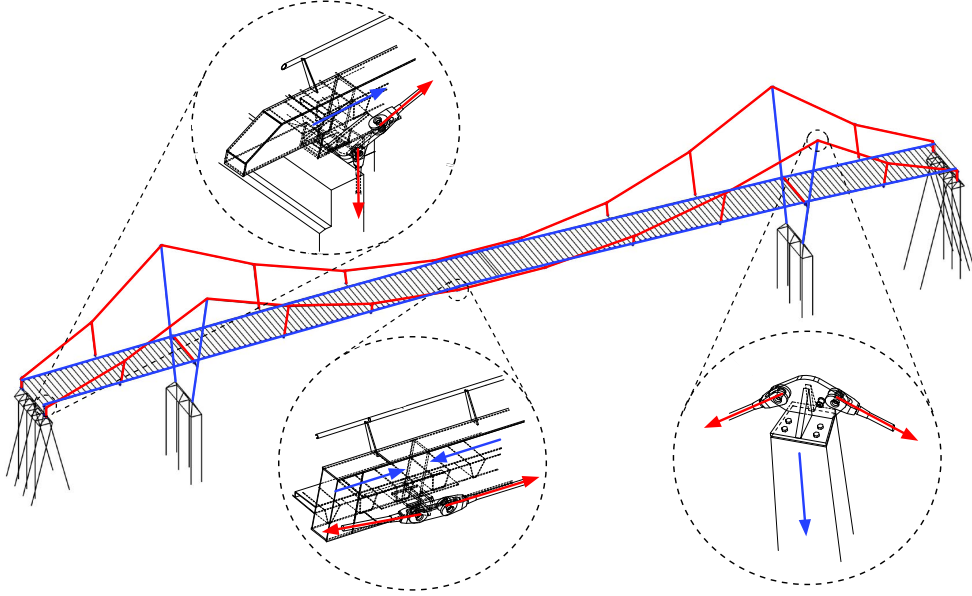


Figure 3.12: Self-anchored suspension bridge concept and force transfer in reference project [43]

Furthermore, observed can be that the design slightly deviates from a conventional self-anchored suspension bridge due to the connection of the main cable to the main girders close to mid-span. Reducing compression forces in the main girder between the cable-to-girder connections. The reason for this design choice was mainly architectural, although it results in a change of structural system.

Figure 3.13 illustrates the bending moment line (BML) for the main girders in the case study, comparing the original configuration to the form-finding configuration. A significant reduction of nearly 90% in bending moments under permanent loads (self-weight and additional dead load) is observed. Suspension bridges are subject to large geometric non-linearities, which must be considered during the design process. This form-finding approach is particularly valuable for geometric parameter studies, as it reduces bending moments and deflections while ensuring the finite element (FE) model achieves geometric equilibrium, enabling smooth iterations required for optimization.

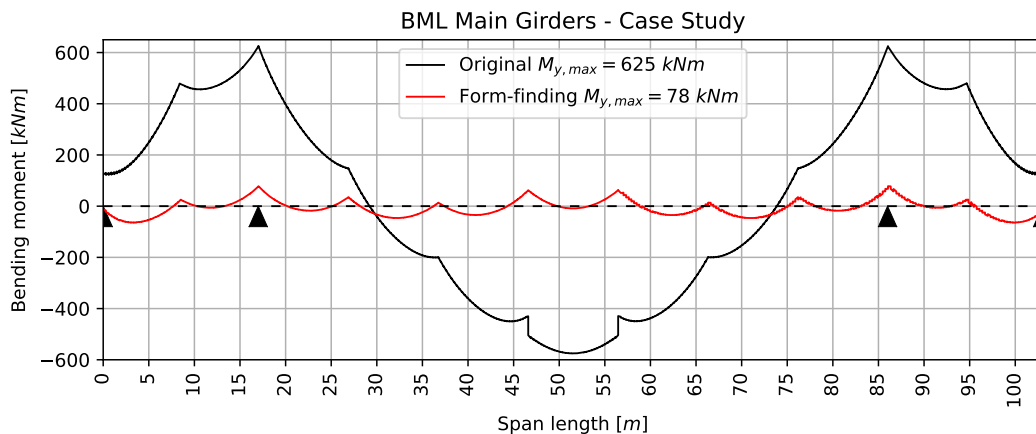


Figure 3.13: Bending moment line of original and form-finding design for permanent loads (self-weight and additional dead load) of the case study Voldijk bridge

3.5 Finite element model

The original design was validated by verification via finite element analysis, using SCIA Engineer 2012 [48] for static analysis and Oasys GSA [49] to assess the structure's dynamic behaviour. These models serve as a starting point for geometric optimisation to mitigate external damping of the case study.

Furthermore, a structural model of the case study is created in SOFiSTiK [50], which serves as a parametric model with its primary goal of iterating to an optimal solution. SOFiSTiK is used for its wide application in bridge engineering, providing solutions for statics, shape finding, non-linear geometric and material, modal and transient analysis. Additionally, the database of the FEM software can be exploited by use of its Python API, allowing for scripting of various parameters and running optimisation algorithms for multiple iterations. Lastly, does the software incorporate verification models based on Eurocodes and national annexes, simplifying structural verification. The finite element model uses beam, truss, cable and plate elements. A focus is given on supporting conditions, mesh sizing and connectivity of certain elements to provide insight of the model.

Since eigenvalue analysis plays an important role in the assessment of human-induced vibrations, are results of both FE models results compared to one another. Additionally, field measurements of the eigenfrequencies after installment of the bridge by means of sledgehammer tests are likewise compared to FE results, see section 3.6.1 Eigenfrequencies in FEM. Furthermore, the TMD design of the original case study is compared to the results of the parametric model.

3.5.1 Substructure

The substructure, consists of foundation piles, piers and abutments. Both abutments and piers are modelled as a 3D structure consisting of 1D elements in which a primary beam connects to the superstructure and secondary beams connect to the foundation piles. These primary and secondary beams are connected via diagonal elements, allowing to effectively transfer vertical and horizontal forces into the foundation. The diagonal elements are modelled with zero weight, only functioning as a connecting element without adding weight to the total foundation. Translational springs are implemented to model the vertical foundation pile behaviour, in which a geotechnical analysis was performed to derive pile characteristics and point resistance. Equation 3.18 shows how the pile behaviour was captured as a spring, whereas equation 3.19 gives the total vertical translation spring.

$$k_{pile} = \frac{EA}{L} \quad (3.18)$$

Where:

k_{pile} = translational stiffness of the foundation pile
 E = modulus of elasticity
 A = area
 L = length

$$k_{tot,vert} = \frac{1}{\frac{1}{k_{pile}} + \frac{1}{k_{tip}}} \quad (3.19)$$

Where:

k_{tot} = total vertical translational stiffness
 k_{pile} = translational stiffness of the foundation pile
 k_{tip} = tip resistance of the foundation pile

Furthermore, horizontal supports are applied at three points of the foundation to account for horizontal forces and prevent rotations in the foundation. This approach is used to reduce computational time, whilst accurately modelling the bridge response when subjected to vertical (dynamic) loads, since vertical and torsional mode shapes have been determined most dominant, see REF. Figure 3.14 shows the foundation of the pylons modelled in FEM.

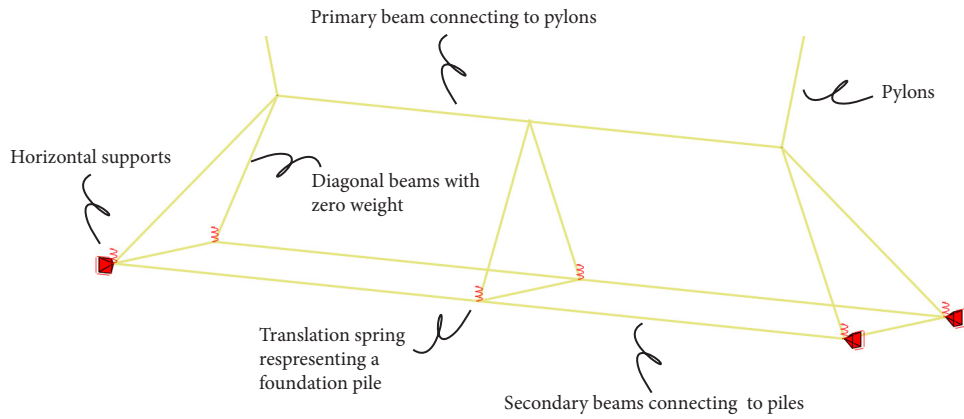


Figure 3.14: Wireframe model of the pylon foundation modelled in FEM

The abutments are modelled via the same approach, but differ at the connection to the superstructure. The pylons are rigidly connected to the primary beam, whereas the abutments connect from the primary beam to main girders via beam elements which do not allow for transfer of tension forces. This accurately models the support condition, since vertical uplift of the main girder is prevented via the cable element connected to the abutment and main cable system only. Additionally, the abutment at the South side is free to translate in horizontal direction, which is why the beam element connecting to the main girder and primary beam is hinged on both ends. Figure 3.15 shows the foundation of the abutments in FEM

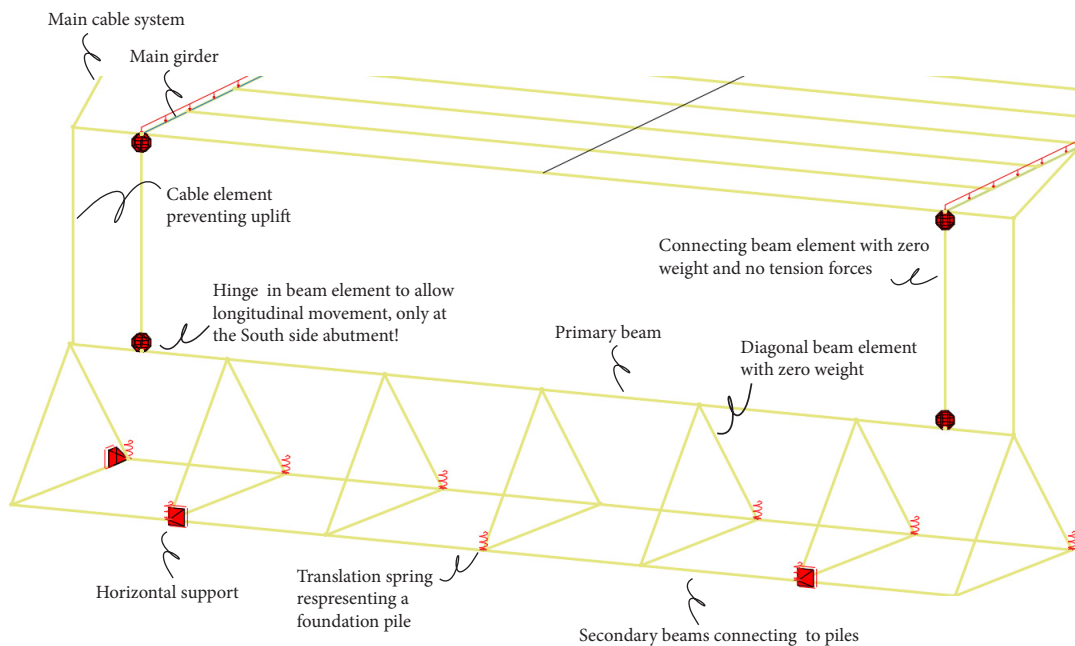


Figure 3.15: Wireframe model of the abutment foundation modelled in FEM

3.5.2 Superstructure

The superstructure model includes rigid connections between the main girder, troughs, and deck plate, with an applied eccentricity of 325 mm in the z-direction relative to the main girder. This eccentricity ensures a more accurate representation of the load transfer from the deck plate and troughs to the main girder, as these components will be welded to the lower part of the girder during production. The main cable system is linked to the abutment through an additional cable element, which prevents uplift, while the main girders are connected to the abutment using a beam element, as shown in Figure 3.16.

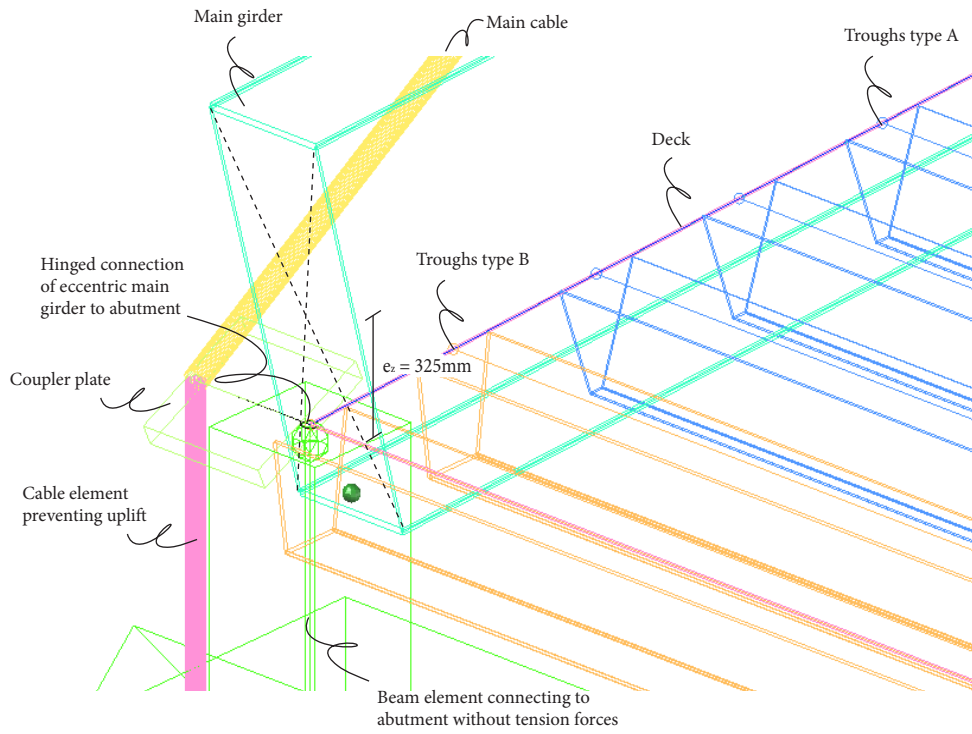


Figure 3.16: Connection detail of the superstructure at the abutment in FEM

The bevelled corbel connection between the main girders and pylons is modelled by means of a HE300A profile, which has pinned circular profiles connecting from the main girder to the pylon directly, or to the cantilevering HE300A profile. This configuration allows for longitudinal translations (sliding), whilst effectively transferring vertical and transverse loads to the pylons, see figure 3.17.

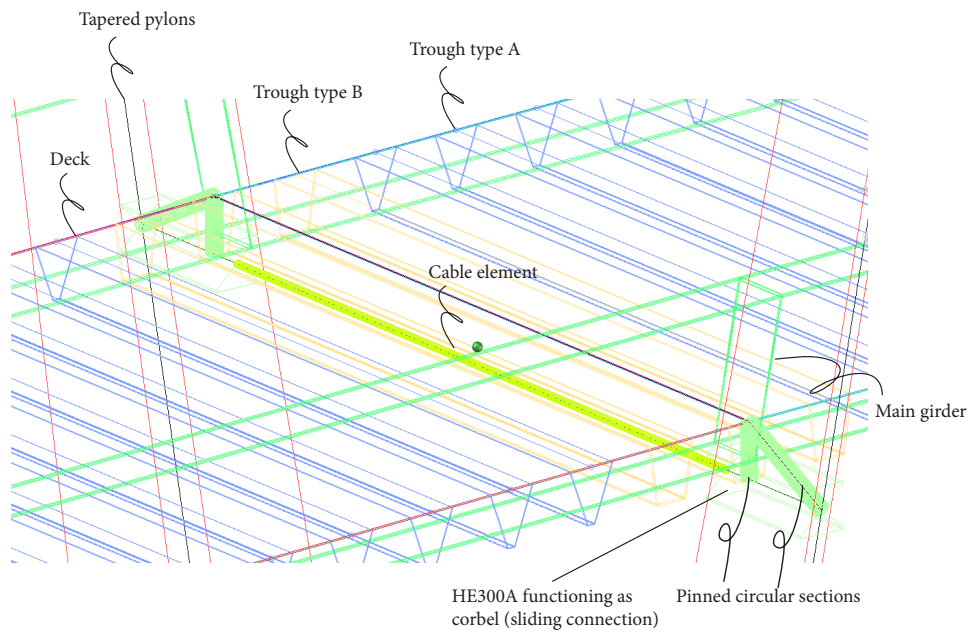


Figure 3.17: Connection detail of the superstructure at the pylons in FEM

The superstructure mainly consists of 1D beam elements, whereas the deck consists of 2D shell elements. A maximum mesh size of 600 mm is used for all elements in the model, for which extra elements are automatically generated to obtain connections, figure 3.18 shows the meshing of the model.

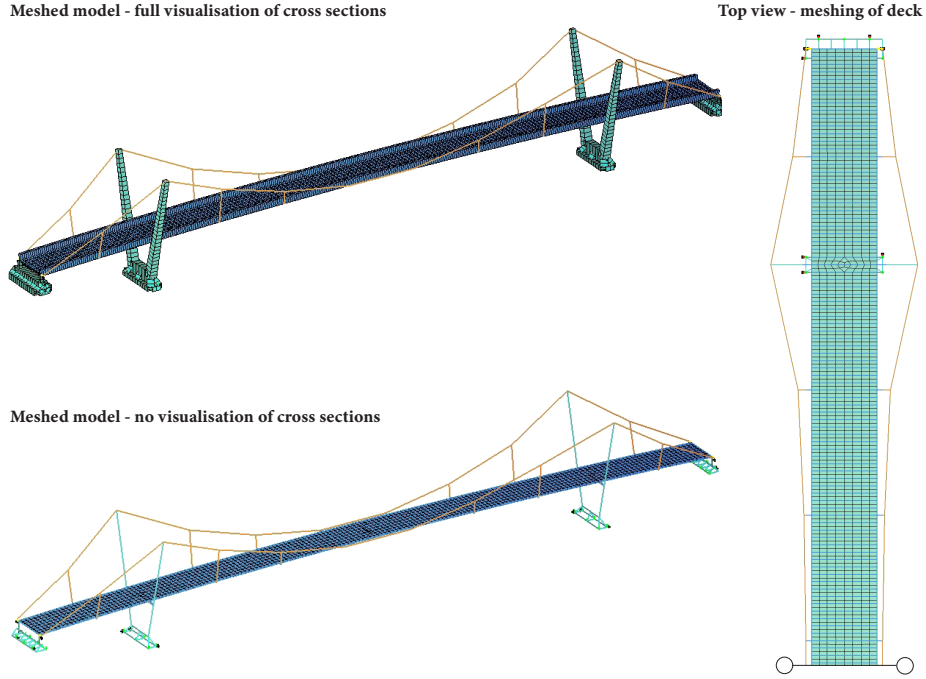


Figure 3.18: Meshing of the FE model for a mesh size of 600 mm

3.6 Eigenfrequencies

Eigenfrequencies play a crucial role in defining the magnitude and application of dynamic loading, which makes accurate determination essential. The eigenfrequencies obtained from eigenvalue analysis of the original design in Oasys GSA are outlined, followed by eigenvalue analysis in SOFiSTiK used for parametric optimisation. Furthermore, sledgehammer tests are presented to verify the calculated eigenfrequencies in FEM via measurements. By comparing these results, one can conclude if the model is an accurate representation and can be used for dynamic analysis for various parametric iterations. Lastly, the implementation of tuned mass dampers (TMDs) in the original design is presented, focusing on the type of dampers used, their functionality, and their placement within the structure.

Before conducting an eigenvalue analysis, it is essential to recognize that adding mass to a structure, whether from additional dead load or dynamic live load, typically causes a downward shift in the eigenfrequency of vertical and torsional modes. In contrast, the impact on lateral and longitudinal vibration modes is minimal, as the added mass primarily acts vertically under the influence of gravity, thereby updating the stiffness matrix predominantly in this direction. Eurocode prEN1991-2 [23] advises: "It is recommended that the mass of pedestrians be considered when calculating the natural frequencies if the modal mass of the pedestrians exceeds 5% of the modal mass of the footbridge." Further context is provided in the JRC document [5], which suggests that the influence of static pedestrian mass can be readily estimated using equation 3.20.

$$m^{**} = \rho \int_{L_D} \mu_d(\Phi(x))^2 dx = \rho m^* \quad (3.20)$$

Where:

- m^{**} = modified modal mass accounting for additional pedestrian mass
- $\rho = \frac{\mu_d + \mu_p}{\mu_d}$ influence factor for additional pedestrian mass
- m^* = modal mass
- μ_d = bridge deck mass per unit span length
- μ_p = pedestrian mass per unit span length
- $\Phi(x)$ = mode shape of the eigenfrequency in question
- L_D = total length of the bridge deck

Equation 3.21 shows the influence of a 5% higher modal mass results in a decrease in eigenfrequency of 2.5%. The JRC-document states that: "this is within the accuracy of the whole model compared to the natural frequencies that will be measured in reality. Therefore, it is recommended to neglect the influence of an increased modal mass lower than 5% on the natural frequency".

$$f'(\rho = 1.05) = \sqrt{\frac{k^*}{1.05m^*}} = 0.976f \quad (3.21)$$

Furthermore, the geometric stiffness matrix of the initial state is added to the elastic stiffness matrix to calculate the eigenfrequencies accordingly. The initial state references the deformed (stressed) state which the structure experiences after applying permanent loads, before dynamic loading. Transforming these additional load cases into effective mass and applying this initial stiffness, the eigenfrequencies are calculated accordingly. Equation 3.22 shows the non-trivial solution of the newly proposed eigenvalue problem.

$$\det \left| [\mathbf{K}_e + \mathbf{K}_g] - \omega^2 [\mathbf{M}_{sw} + \mathbf{M}_{add}] \right| = 0 \quad (3.22)$$

Where:

- \mathbf{K}_e = elastic stiffness matrix of the structure
- \mathbf{K}_g = geometric stiffness matrix of the structure incorporating the initial state
- \mathbf{M}_{sw} = mass matrix of the structure's self weight
- \mathbf{M}_{add} = mass matrix obtained from conversion of add. load cases (add. dead load and weight ped.)
- ω = (angular) eigenfrequencies of the structure

3.6.1 Eigenvalue analysis in FEM

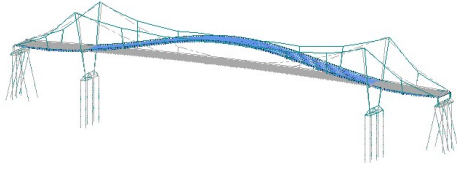
The eigenfrequencies obtained in Oasys GSA regarding the original case study within a range of $f_n = 0 \sim 5 \text{ Hz}$, are listed in the table below. It presents two loading conditions, being permanent loads (i.e. self-weight and additional dead loads) and permanent loads including the static loading of pedestrians for traffic class 3. Only the eigenfrequencies of interest are shown, meaning that local cable oscillations which do not influence pedestrian-structure interaction have been excluded. Figure 3.19 shows the mode shapes and eigenfrequencies of interest for **permanent loads only**.

Mode nr.	Perm. loads [Hz]	Perm. loads + TC3 [Hz]	Shift [%]	Mode shape
5	1.598	1.515	-5.19	1 st vertical
10	2.222	2.229	0.315	1 st lateral ¹
16	2.841	2.698	-5.03	2 nd vertical
17	2.893	2.806	-3.55	1 st torsional
18	3.398	3.393	-0.147	2 nd lateral
27	4.274	4.072	-4.73	3 rd vertical
34	4.725	4.614	-2.349	2 nd torsional
38	4.833	4.651	-3.76	3 ^{rd*} vertical

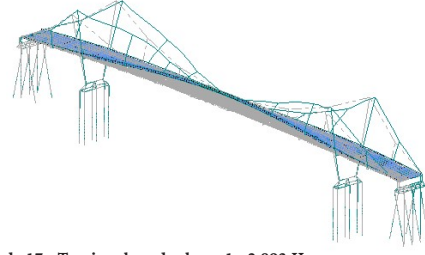
Table 3.1: Eigenvalue analysis in Oasys GSA of the original design for permanent loads and permanent loads including the static load of pedestrians for traffic class 3 (TC3).

¹One can observe small torsional contributions for the mode shape of mode 10, however, since lateral movement is dominant, mode 10 is classified as a lateral mode shape.

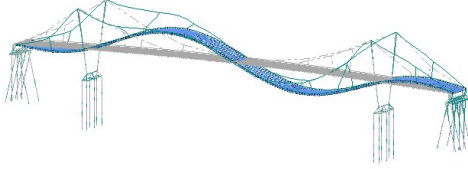
Mode 5 - Vertical mode shape 1 - 1.598 Hz



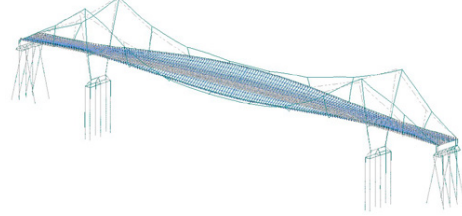
Mode 10 - Lateral mode shape 1 - 2.222 Hz



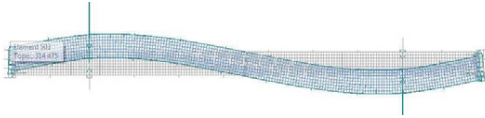
Mode 16 - Vertical mode shape 2 - 2.841 Hz



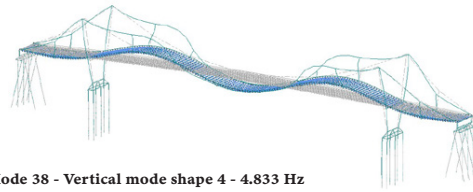
Mode 17 - Torsional mode shape 1 - 2.893 Hz



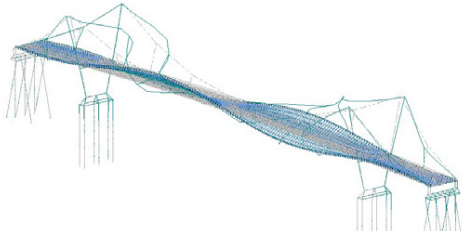
Mode 18 - Lateral mode shape 2 - 3.398 Hz



Mode 27 - Vertical mode shape 3 - 4.274 Hz



Mode 34 - Torsion mode shape 2 - 4.725 Hz



Mode 38 - Vertical mode shape 4 - 4.833 Hz



Figure 3.19: Mode shapes and eigenfrequencies obtained from eigenvalue analysis in Oasys GSA regarding the original design for **permanent loads**

The eigenfrequencies obtained in SOFiSTiK of the original design for a range $f_n = 0 \sim 5 \text{ Hz}$, are listed in the table below. These eigenfrequencies are presented to show how they compare to the eigenfrequencies obtained in Oasys GSA. The original design serves as a benchmark from which parametric iterations can be applied in order to optimise external damping. It once again presents two loading conditions, permanent loads (i.e. self-weight and additional dead loads) and permanent loads including static loading for pedestrians of traffic class 3. Figure 3.20 presents the mode shapes and eigenfrequencies of interest for **permanent loads only**.

Mode nr.	Perm. loads [Hz]	Perm. loads + TC3 [Hz]	Shift [%]	Mode shape
1	1.610	1.527	-5.155	1 st vertical
2	2.062	2.098	1.746	1 st lateral
7	2.801	2.661	-4.998	2 nd vertical
8	2.915	2.887	-0.961	1 st torsional
17	3.630	3.668	1.047	2 nd lateral
19	4.610	4.550	-1.302	2 nd torsional
21	4.666	4.473	-4.136	3 rd vertical

Table 3.2: Eigenvalue analysis in SOFiSTiK of the original design for permanent loads and permanent loads including the static load of pedestrians for traffic class 3 (TC3).

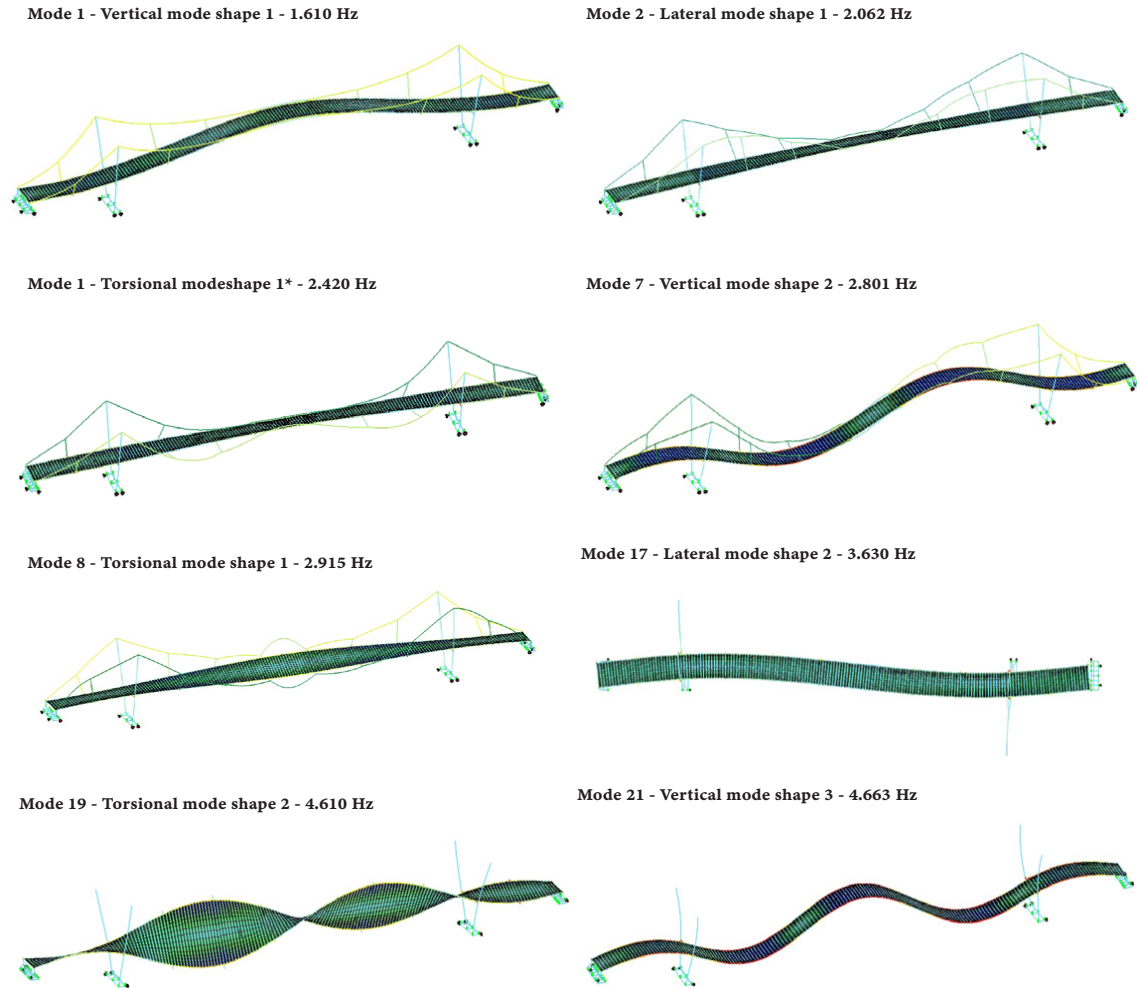


Figure 3.20: Mode shapes and eigenfrequencies obtained from eigenvalue analysis in SOFiSTiK regarding the original design for **permanent loads**

The eigenvalue analysis reveals that the FEM software produces comparable eigenfrequencies and mode shapes, as summarized in Table 3.3. Notably, SOFiSTiK identifies only one third-order vertical mode shape (mode 19), whereas Oasys GSA identifies two such mode shapes (modes 27 and 38).

Perm. loads [Hz]			Difference [%]	Perm. loads + TC3 [Hz]			Mode shape
Oasys	GSA	SOFiSTiK		Oasys	GSA	SOFiSTiK	
1.598		1.610	0.625	1.515		1.527	1 st vertical
2.222		2.062	-7.20	2.229		2.098	1 st lateral
2.841		2.801	-1.79	2.698		2.661	2 nd vertical
2.893		2.915	-0.933	2.806		2.887	1 st torsional
3.398		3.630	7.210	3.393		3.668	2 nd lateral
4.274		-	9.80	4.072		-	3 rd * vertical
4.725		4.601	2.695	4.614		4.518	2 nd torsional
4.833		4.663	-	4.651		4.550	3 rd vertical

Table 3.3: Comparison of eigenvalue analysis for permanent loads and permanent loads including traffic class 3 (TC3) obtained in Oasys GSA and SOFiSTiK

3.6.2 Sledgehammer tests

Besides eigenvalue analysis in FEM software, sledgehammer tests have been performed. The case study was excited by a sledgehammer positioned three meters from midspan at the edge of the bridge deck. The response was measured by 64 piezoelectric accelerometers, spaced 3.2 m across the entire length of the bridge. Figure 3.21 shows the location of the sledgehammer excitation and the position of the accelerometers.

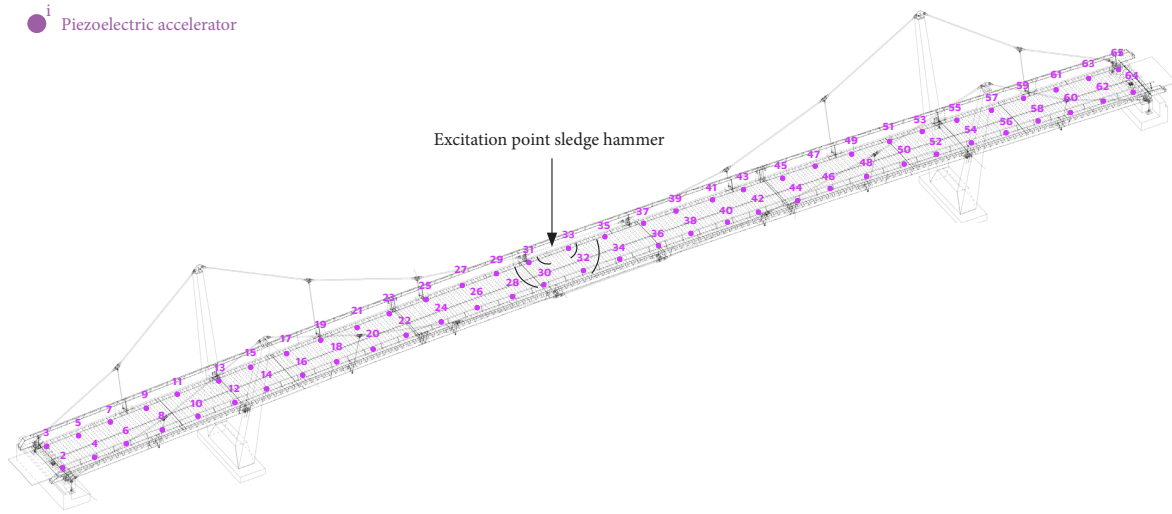


Figure 3.21: Excitation point of sledgehammer and position accelerometers for performed sledgehammer tests

Fast Fourier Transform (FFT) analysis was performed within a frequency range of 0 Hz to 100 Hz . Results have been processed via modal analysis software DIAMOND [51] and the measured eigenfrequencies are presented in table 3.4 and the corresponding mode shapes in figure 3.22.

Mode nr.	Eigenfrequency [Hz]	Mode shape
1	1.681	1 st vertical
2	2.789	2 nd vertical
3	3.137	1 st torsional
4	4.587	3 rd vertical

Table 3.4: Eigenfrequencies derived from sledgehammer tests

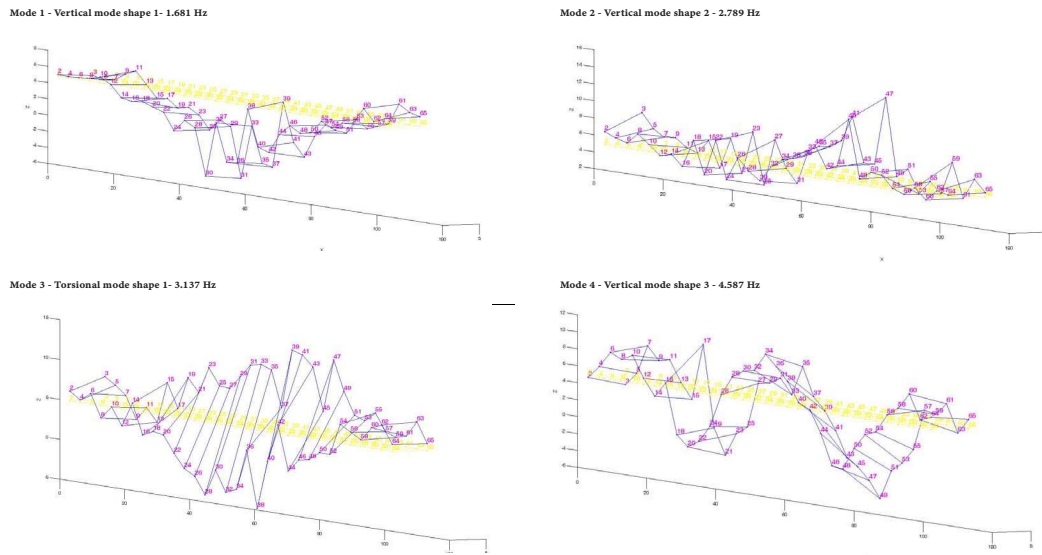


Figure 3.22: Mode shapes obtained from sledgehammer tests

3.6.3 Comparison of eigenfrequencies

Before comparison of results, an account should be made for the fact that the eigenfrequencies obtained from hammer impact tests only consider self weight and dead load, excluding the static load from pedestrians since no crowd loading is present during testing. The comparison for eigenfrequencies under **permanent loads** is presented in table 3.5.

Sledge hammer [Hz]	Oasys GSA [Hz]	SOFiSTiK [Hz]	Mode shape
1.681	1.598	1.610	1 st vertical
2.789	2.841	2.801	2 nd vertical
3.137	2.893	2.915	1 st torsional
4.587	4.833	4.663	3 rd vertical

Table 3.5: Comparison of eigenfrequencies obtained from eigenvalue analysis and hammer impact tests for **permanent loads**

Since the sledgehammer was excited vertically, only account for vertical and torsional modes of vibration was made, leading to the exclusion of lateral frequencies in the measured results. Furthermore, can be observed that the eigenvalue analysis from FEM accounts for the second torsional mode as well as the fourth vertical mode, whereas the sledgehammer tests do not.

Eurocode pr-EN1991-2 states that "when assessing the footbridge natural frequencies, an allowance should be made for possible differences observed on the constructed structure, which often occur due to different than predicted boundary conditions or to other modelling insufficiencies. In the absence of a sensitivity test, a frequency variation of 5 % can be considered".

The difference in eigenfrequencies from Oasys GSA and SOFiSTiK in comparison to sledgehammer tests is presented in table 3.6. Showing that the results exceed the 5% variation stated earlier. Various explanations could be given for these differences. Primarily, shortcomings in FEM modelling are to be identified.

Sledge Hammer - Oasys GSA	Sledge Hammer - SOFiSTiK	Mode shape
4.94%	4.34%	1 st vertical
1.86%	0.42%	2 nd vertical
7.78%	7.708%	1 st torisonal
6.82%	1.63%	3 rd vertical

Table 3.6: Difference between measured and FEM-derived eigenfrequencies

3.7 Structural damping

For the case study, a minimum structural damping value for welded steel bridges of $\xi_{s,min} = 0.2\%$ is considered, according to EUR23984 [5]. The reason for applying a minimum structural damping value is to obtain conservative acceleration results. Since no modal damping values have been derived from testing and the literature does not mention damping values of comparable structures, has been opted for the lower bound value.

The newly proposed Eurocode prEN1991-2-2021 [23] states that "in the absence of other information, damping ratios for serviceability conditions with average amplitudes of vibration may be taken in accordance with Table G.4". The table is based on construction material and enlists a critical damping ratio of $\xi_{s,avr} = 0.4\%$. Whereas the Dutch national annex of Eurocode EN1991-2-2019 [32] refers to a logarithmic decrement of $\delta_{s,weld} = 0.02$ for welded steel bridges and truss structures. Equation 3.23 gives the relation between critical damping value and logarithmic decrement, which converts to a critical damping value of $\xi_{s,weld} = 0.318\%$.

$$\xi = \frac{\delta}{\sqrt{\delta^2 + (2\pi)^2}} \quad (3.23)$$

The FE model uses Rayleigh damping with one damping value ($\xi_{s,min} = 0.2\%$) as outlined in section 2.4.4 - Assessment of structural damping, which is applied to all modes within the critical range.

3.8 Design situations

With the determination of eigenfrequencies can be concluded that the structure is prone to pedestrian-structure interaction, as outlined in section 2.4.2 - Critical range of natural frequencies. Giving implication to account for dynamic effects according to 2.4 - Design steps. Assessment of the design situations is performed to determine if maximum deck accelerations are within the given limit. This regards a serviceability limit state, for which the comfort of its users is verified.

What follows is the determination of design situations based on associated traffic and comfort classes. These account for the anticipated traffic whilst reflecting the comfort requirements for its users. Four specific design situations are considered in this case study, as outlined in table 3.7. Each scenario must adhere to the specified maximum acceleration limits to satisfy the serviceability limit state criteria. However, analysis has shown that external damping is necessary to mitigate pedestrian-induced vibrations effectively.

Situation	Description	Traffic class	Occurance	Comfort class
1	Small group of hikers	TC1	Daily	CL1
2	Commuting traffic	TC2	Daily	CL1
3	Dense traffic	TC3	Weekly	CL2
4	Inauguration	TC4	Once	CL3

Table 3.7: Design situations to be considered for the case study

3.9 External damping

Tuned mass dampers (TMDs) have been applied to limit pedestrian-induced vibrations to an acceptable level. Transient analysis is performed in Oasys GSA for all critical modes to verify the maximum response. The original analysis is presented and compared to the parametric model in SOFiSTiK, followed by a description of the TMD design, its belonging characteristics and placement within the structure.

3.9.1 Evaluation of response

The evaluation of the response is only performed for pedestrian streams since moving loads have been deemed non-dominant in the original design. The analysis of moving loads, i.e. pedestrians/hikers and joggers as enlisted in 2.5.5 - Moving harmonic point load model, is presented in for completeness. The eigenfrequencies inside the critical range considered for the analysis are shown in table 3.8. The third vertical mode $f_{vert,3} = 4.274$ Hz is excluded since its frequency is within range of the second harmonic, giving a reduction factor of $\psi = 0.25$, greatly reducing its response and excluding the need for verification². No account for horizontal accelerations has been made since the lateral modes of vibration $f_{lat,1} = 2.229$ Hz and $f_{lat,2} = 3.393$ Hz fall outside the critical frequency range.

Mode nr.	Frequency	Mode shape
5	1.598	1 st vertical
16	2.698	2 nd vertical
17	2.806	1 st torsional

Table 3.8: Eigenfrequencies obtained in Oasys GSA, subjected to transient analysis for tuned mass damper design of the case study

Transient analysis is used to determine the maximum vertical acceleration of each critical mode, in which dynamic loading for pedestrian streams is applied according to its mode shape for a frequency range of $1 \sim 3$ Hz. Plotting a range of frequencies can account for modal coupling if present, although it should be stated this only occurs when frequencies are in close proximity to one another. The total response of the original analysis is obtained by taking the square root of the sum of squares (SRSS), see equation 3.24. However, use of the SRSS seems excessive since modal coupling is unlikely to happen and using transient analysis accounts for the contribution of all modes simultaneously.

²A single transient analysis was performed at resonance frequency to determine if the third vertical mode could be of potential risk to pedestrian-structure interaction, which was deemed non-dominant.

$$a_{tot,vert} = \sqrt{\sum_{i,vert}^n a_{i,vert}^2 + \sum_{i,tor}^n a_{i,tor}^2} \quad (3.24)$$

where:

- $a_{tot,vert}$ = total vertical accelerations
- $a_{i,vert}$ = acceleration from vertical critical mode i
- $a_{i,tor}$ = acceleration from torsional critical mode i

Structural damping is assumed with a lower bound value of $\xi_{s,min} = 0.2\%$. The reduction factor ψ has not been applied yet, giving better insight into the dynamic behaviour. The response has been measured at four points of the bridge deck, namely at midspan and one-fourth from midspan (points A1 to B2). The placement of these points is in the middle and outer edge of the bridge deck to account for vertical and torsional vibrations. Figure 3.23 shows these points used for measurement.

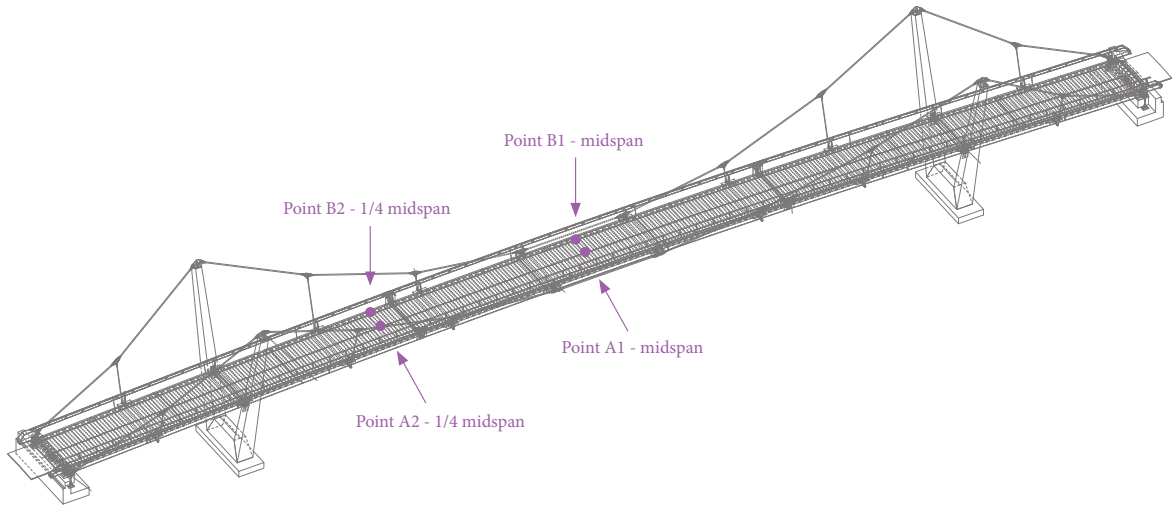
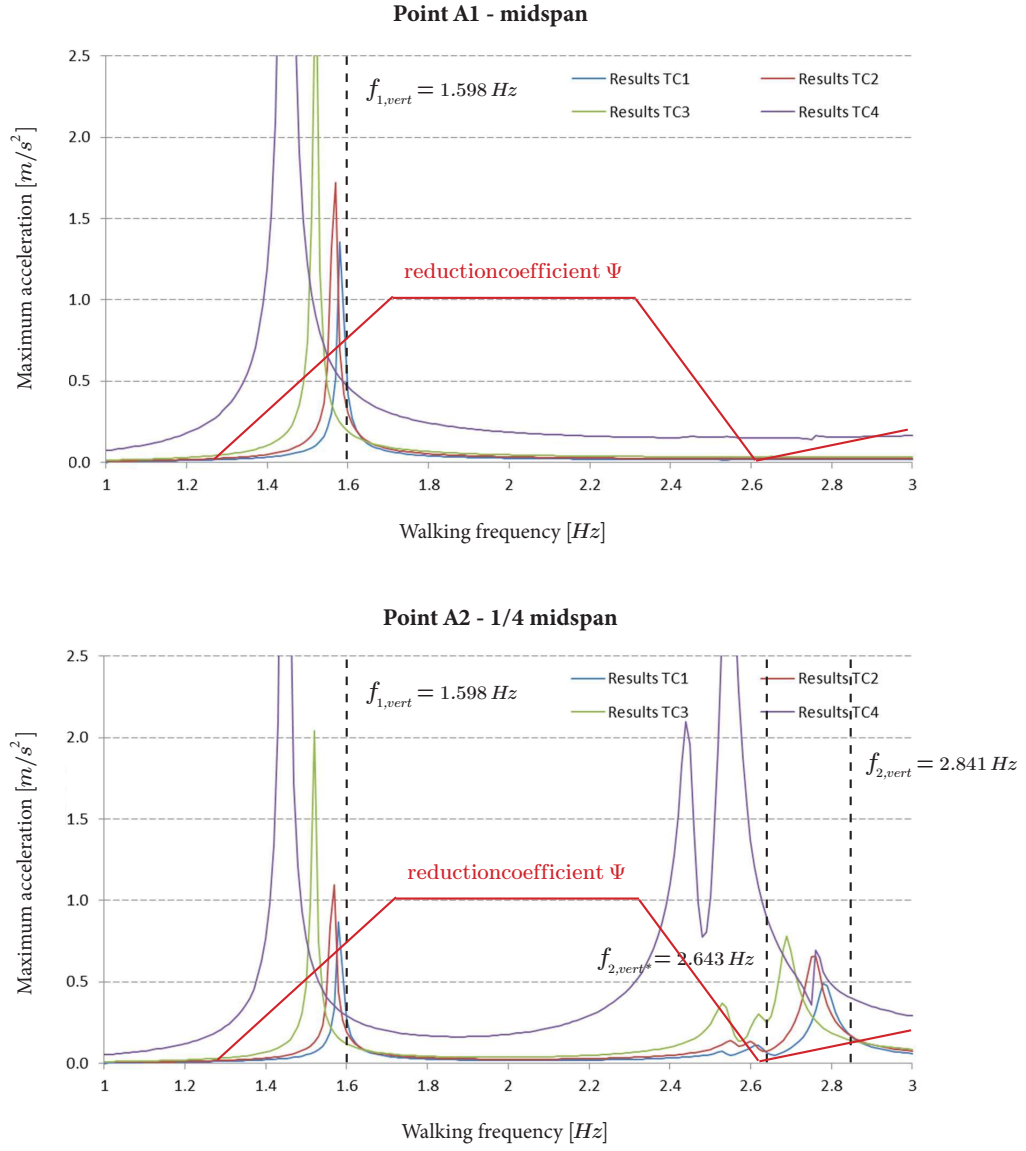


Figure 3.23: Measurement points - Transient analysis Oasys GSA

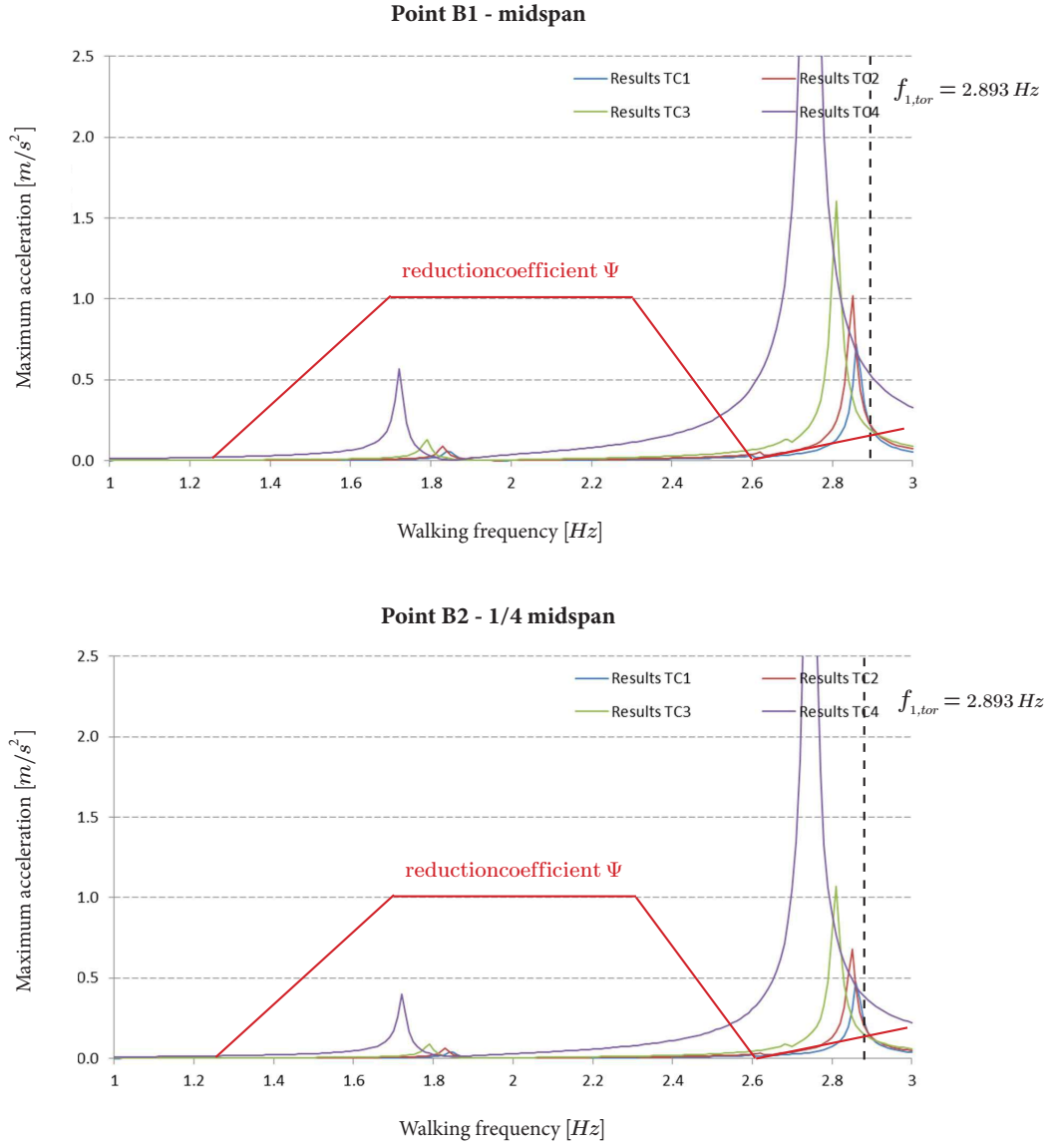
The results for vertical modes of vibration in point A1 and A2 are shown in figure 3.24. Observed can be that the walking frequency exhibits resonance phenomena at the eigenfrequencies of the structure. The second vertical mode shape does not resonate in point A1, since measurements are taken mid span which forms a nodal point for this mode.

Furthermore, the eigenfrequency shifts downward for increased traffic classes, resultant from increased static loading, as explained in 3.6 Eigenvalue analysis in FEM. Additionally, point A2 shows that a new resonance peak is obtained at $f_{2,vert*} = 2.643 \text{ Hz}$. This was deemed a local cable oscillation at first, having no contribution to the overall response of the structure, but is better described as an alteration of the second vertical mode. Determining the response for a range of frequencies prevents the underestimation of critical mode contributions and their magnitude.



Obtained vertical accelerations in point A1 and A2 via transient analysis via Oasys
Figure 3.24: GSA, depicting the reduction coefficient still to be applied for dynamic loading based on walking frequency

Figure 3.25 presents torsional vibrations for points B1 and B2. Observed can be the resonance peak of the first torsional mode shape. Furthermore, does a small peak form around $f_w = 1.7 \text{ Hz}$. Since the comfort limit for traffic class 4 (TC4) is set at comfort class 3 (CL3, minimum comfort) with a limit of $a_{max} < 2.5 \text{ m/s}^2$, the contributions are ignored.



Obtained torsional accelerations in point B1 and B2 from transient analysis via
Figure 3.25: Oasys GSA, depicting the reduction coefficient still to be applied for dynamic loading based on walking frequency

To obtain the total response, the contributions of the vertical and torsional accelerations presented in figure 3.24 and 3.25 are combined to a total response via the SRSS. However, based on the accelerations from all four points (A1 to B2), can it be concluded that only the first vertical mode shape $f_{vert,1} = 1.598 Hz$ is considered dominant, excluding the need to obtain the total response. The contribution of the other mode shapes is diminished due to the reduction factor present. Since the first vertical eigenmode results in accelerations outside the stated limits, external damping is required to mitigate the pedestrian-induced motion.

3.9.2 TMD design

The design process for TMDs begins with the selection of a mass ratio μ , which represents the damper's mass relative to the target modal mass. A mass ratio of $\mu = 4\%$ was selected for the case study, resulting in a damper mass of $m_d = 2000$ kg. However, due to potential transportation and installation challenges, four TMDs with a mass of 500 kg each was considered in the final design. Besides the mass ratio, are there two key parameters to be identified, namely the optimum tuning ρ_{opt} and damping ratio ξ_d , as explained in 2.7 - External damping. Tuning the design according to these parameters will yield an optimum damper design.

The TMDs target the first vertical eigenmode, having a frequency of $f_{1,vert} = 1.680$ Hz as derived from sledgehammer tests. Results of the TMD design in FEM are slightly contrary, since a lower eigenfrequency of $f_{1,vert} = 1.598$ Hz and $f_{1,vert} = 1.610$ Hz in Oasys GSA and SOFiSTiK has been derived respectively. Figure 3.26 presents the TMDs and its positions within the structure.

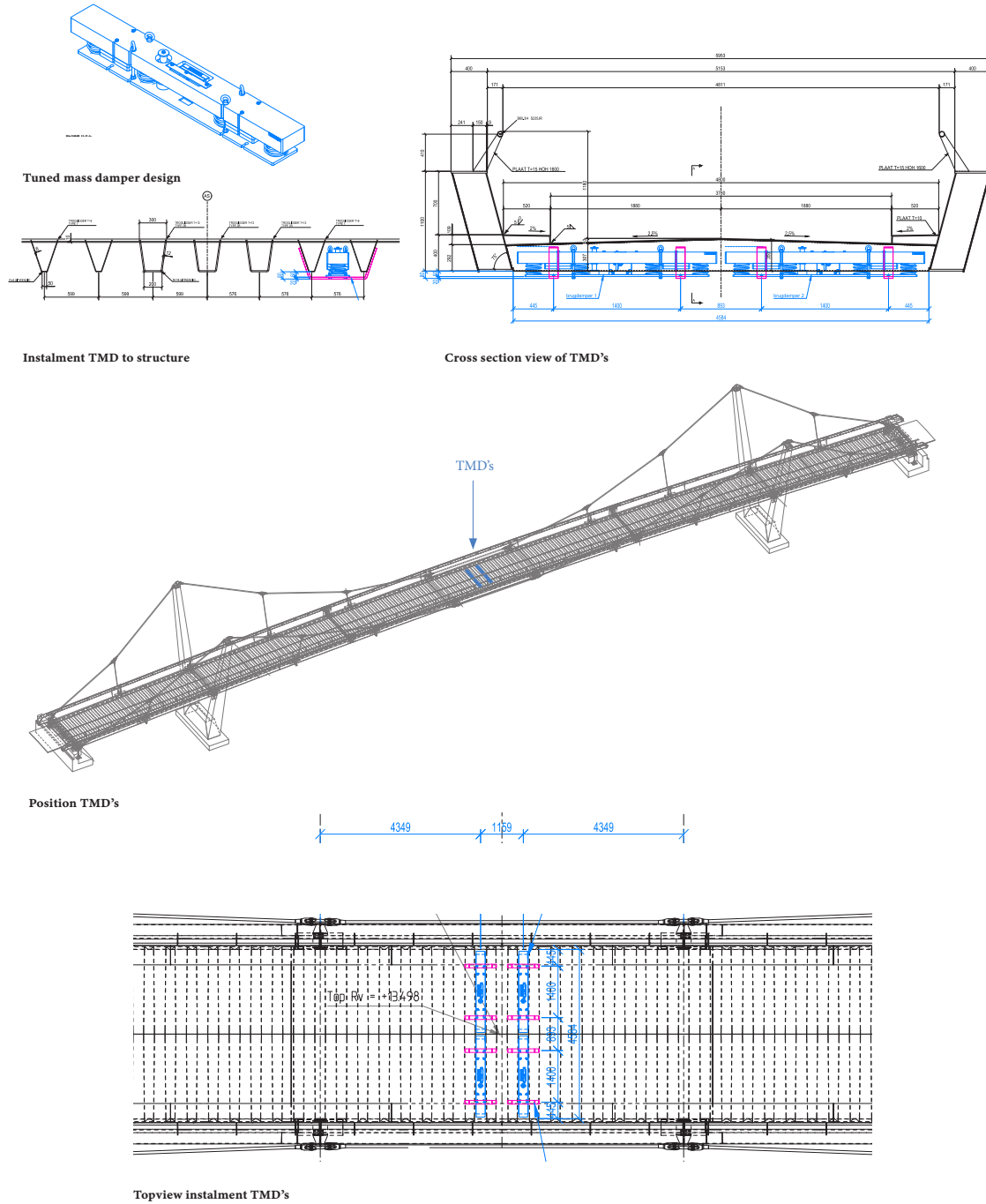


Figure 3.26: Positioning of four TMDs according to final design

Properties TMDs

$m = 50 \text{ t}$ - modal mass of the damped mode
 $f_{1,vert} = 1.680 \text{ Hz}$ - target frequency
 $\xi = 0.2\%$ - structural damping
 $\mu = 4\%$ - mass ratio
 $\xi_{d,opt} = 11.5\%$ - optimal damping ratio of the damper, see 2.75
 $m_d = 0.5 \text{ t}$ - damper mass
 $n = 4$ - number of dampers
 $f_{d,opt} = 1.492$ - optimal frequency of the damper, see 2.73
 $c_d = 1.173 \text{ Ns/mm}$ - damping constant of the damper
 $k_d = 12.9 \text{ N/mm}$ - spring constant of the damper

To validate TMD design, transient analysis was performed using FEM. Figure 3.27 gives a schematic overview of how the TMDs are modelled. Results are presented exclusively from the parametric model in SOFiSTiK, as the original TMD design in Oasys GSA featured a different configuration. Transient analysis is performed to account for the local damping effects the TMDs have on the structure.

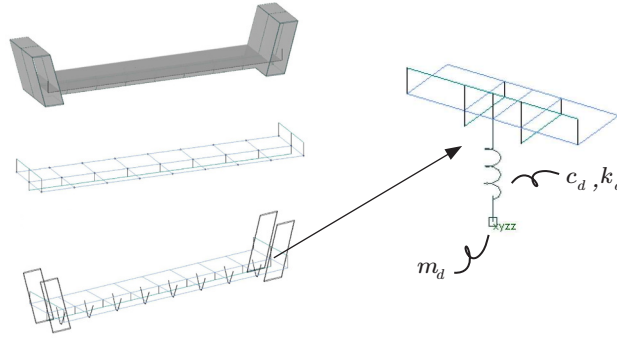


Figure 3.27: Modelling of TMDs in FEM

By adding mass to the structure in the form of a TMD, an additional degree of freedom is obtained. TMD instalment present an additional eigenfrequency and shifts the targeted frequency downwards due to added mass. The lower eigenfrequency consists of TMD movement in phase with the structure, whereas the newly introduced eigenfrequency has TMD movement in anti-phase relative to the structure. Figure 3.28 shows these eigenfrequencies after TMD instalment.

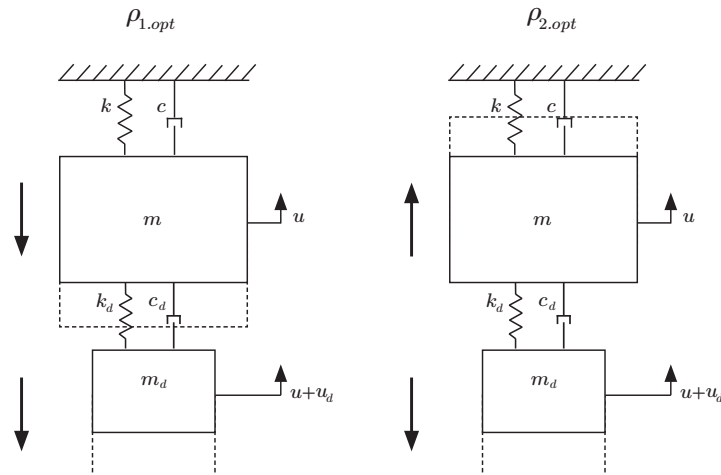


Figure 3.28: Eigenfrequencies obtained after instalment TMDs

Following the measurements from sledgehammer testing of eigenfrequencies after instalment of the footbridge, a revised TMD design was proposed. The parametric model was verified using the final TMD design. It is important to note that the target frequency of the dampers is tuned to the first vertical eigenfrequency obtained from FE analysis rather than the measured eigenfrequency. Using the measured eigenfrequency would yield unsatisfactory results, as the design philosophy focuses on damping an eigenfrequency, not an arbitrary frequency.

Figure 3.29 presents the transient response spectrum for ranging traffic classes of the final TMD design with its proposed target frequency $f_{1,vert} = 1.610 \text{ Hz}$. Observed can be the increased maximum acceleration for higher traffic classes and a shift in eigenfrequencies due to the added mass resultant from increased pedestrian density. The fixed points P and Q are shown for each traffic class. According to the TMD design theory is an optimal design achieved when the magnitude of these points is equal to one another. This holds best when the eigenfrequency of the traffic class in question is closest to the target frequency, being TC 1. TC4 shows a sub optimal design, where the first resonance peak (TMD and strucure moving in-phase) has a greater acceleration content than the second resonance peak (anti-phase movement).

One could make the claim that by tuning the TMDs to a lower target frequency, results will be improved for the highest traffic class. However, since the increased mass due to pedestrians leads to a downward shift of the first resonance peak, will the reduction coefficient ψ be of such magnitude that the response will be considerably lower. Reduction of the dynamic loading has not been implemented in figure 3.29, giving better insight into the dynamic response.

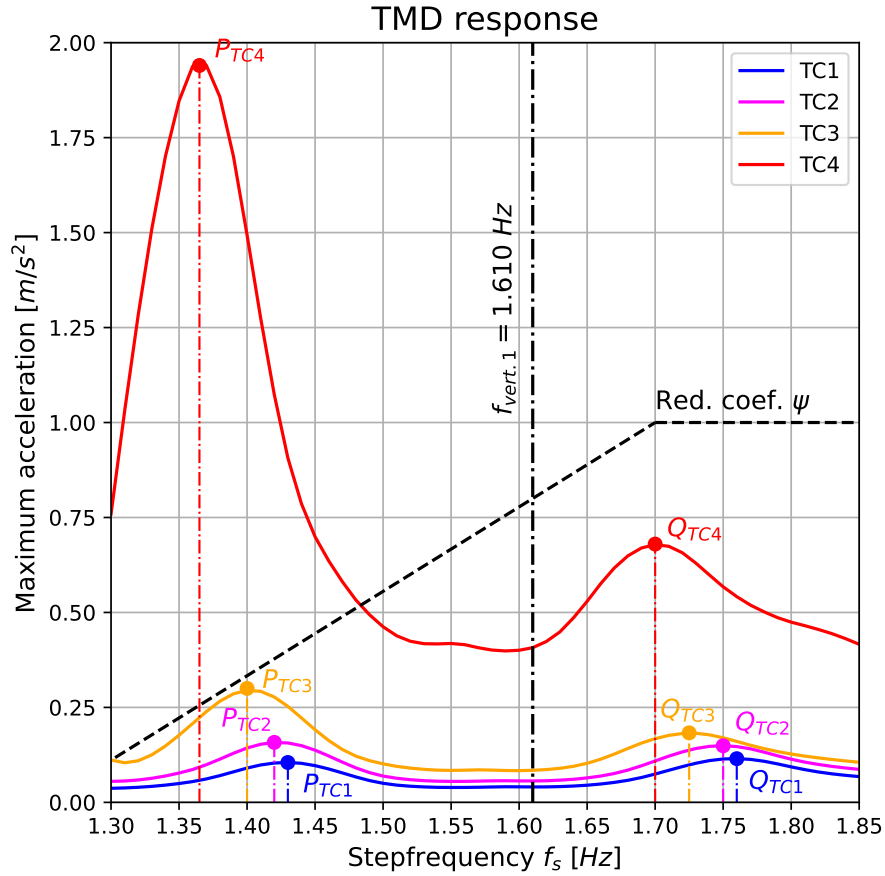


Figure 3.29: Transient response of the TMD design for ranging traffic classes for a target frequency of $f_{vert.1}=1.610 \text{ Hz}$

Table 3.9 presents the results of the transient analysis of the parametric model in SOFiSTiK as opposed to the set limits regarding the design situations stated earlier. Observed can be that the serviceability limit state in terms of comfort is satisfied.

Design Situation	Traffic class	Comfort class	Limit [m/s^2]	Max. acc. [m/s^2]
1	TC1	CL1	0.50	0.12
2	TC2	CL1	0.50	0.16
3	TC3	CL2	1.00	0.30
4	TC4	CL3	2.50	1.83

Table 3.9: Verification TMD design case study excluding reduction factor ψ

Since test results of the response after instalment of the TMDs is missing, no comparison can be made between the results from transient analysis and the measured response. However, even with response measurements will it be hard to quantify the behaviour. An approach would be to obtain the acceleration response after stationary loading by pedestrians according to the modeshape of the first vertical eigenmode, for which no account for the reduction coefficient ψ is made in FE analysis.

It is important to recognize that a TMD design is never perfectly tuned, as the bridge deck is subjected to varying traffic classes, causing the eigenfrequency to shift away from the tuned frequency. This variability in loading introduces significant challenges in maintaining optimal performance, as the dynamic characteristics of the structure evolve over time. Figure 3.29 illustrates a suboptimal response for TC4, where one peak is higher than the other, highlighting the impact of this frequency shift on the effectiveness of the TMD.

The limitations of TMD tuning frequency mean that achieving ideal performance across all design scenarios is impractical. Instead, designers must make decisions about which loading conditions or traffic scenarios to prioritize. For instance, scenarios with the highest likelihood of inducing resonance or those representing the most critical design cases will dominate.

4

Case study assessment

Accurately assessing the maximum acceleration across the bridge deck is critical for mitigating the external damping of pedestrian bridges. Part I, Assessment of pedestrian-structure interaction, introduces various methods for evaluating this behaviour. This chapter centres around the case study, demonstrating the application of these methods and comparing their results. Analytical expressions for predicting the response of girder bridges are excluded, as the case study focuses on a suspension bridge with a parallel structural system, which differs significantly from girder bridge designs.

The response spectrum method is likewise excluded since its design spectrum is once again tailored to girder bridges. As noted in EUR23984: "The design method was elaborated with beam bridge models. If the structural behaviour of a bridge differs significantly from that of a beam bridge, limits of application of the spectral method may be reached" [5]. The 4FFH, SDOF and transient analysis methods are applied for the assessment of modes in the critical frequency range as shown in the table below. Every method is subjected to a minimum structural damping value of $\xi_{s,min} = 0.2\%$, as outlined in 3.7 - Structural damping.

Mode nr.	Frequency	Mode shape
5	1.610	1 st vertical
4	2.420	1 ^{st*} torsional
7	2.801	2 nd vertical
8	2.915	1 st torsional

Table 4.1: Critical eigenfrequencies obtained in SOFiSTiK used for assessment

Both crowd-loading and moving loads are assessed according to codes and guidelines. The parametric model created in SOFiSTiK of the original design is used for verification since it will likewise serve in the optimisation. According to the design procedure, an assessment should be performed for various dynamic load types. The response to a single pedestrian (4FFH), pedestrian crowds and moving loads such as hikers and joggers is presented. Lastly, measurements obtained from sledgehammer testing are presented and conclusions are drawn on the most suitable assessment methods. Table 4.2 gives an overview of the relevant methods and their purpose.

Load type	Method	Use
Single moving pointload	4FFH	Qualitative
Crowd-loading	SDOF Direct time	Quantitative
Moving loads (pedestrians/joggers)	SDOF Direct time	Quantitative

Table 4.2: Assessment methods utilised for the case study

4.1 Four Footfall Harmonics

The Four Footfall Harmonics (4FFH) method, outlined in 2.6.3 - Four Footfall Harmonics, provides a spectrum of acceleration responses generated by a single pedestrian walking across the footbridge. This method incorporates higher harmonics up to the fourth, enabling the analysis of vertically induced vibration responses. Originally developed for indoor flooring environments, the method can be adapted for outdoor pedestrian bridge design. Figure 4.1 gives a schematic representation of the analysis regarding the case study.

It is important to note the 4FFH method has limitations. It does not account for pedestrian streams or joggers, which restricts its ability to accurately predict the maximum acceleration as required. Additionally, it only evaluates vertically induced loads, leaving laterally induced vibrations to be analysed using an alternative assessment method.

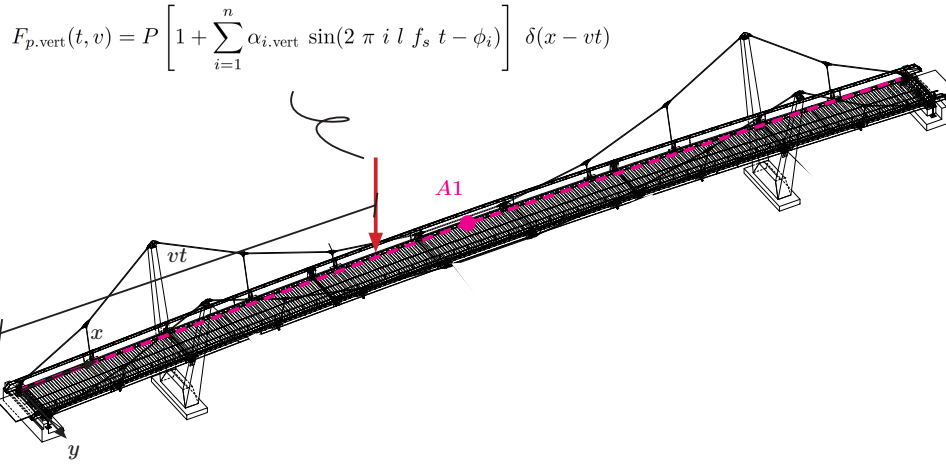


Figure 4.1: Representation of 4FFH analysis of the case study

EUR23984 states that "A vertical vibration excitation by the second harmonic of pedestrian forces might take place. Until now there is no hint in the literature that significant vibration of footbridges due to the second harmonic of pedestrians have occurred" [5]. This contrasts the 4FFH method, which incorporates effects up to the fourth harmonic. Figure 4.2 presents the full response spectrum of the 4FFH analysis, for which point A1 at midspan has been analysed. The placement of the point at the outer edge of the bridge deck accounts for both vertical and torsional vibrations. Resonance peaks are observed at the eigenfrequencies previously identified in table 3.2 - Eigenvalue analysis in FEM. Mode VM3 and TM3 show modal coupling, for which two resonant peaks are centered around the eigenfrequency. A second torsional mode TM2 is identified in close occurrence of VM3, whereas a fourth vertical mode VM4 shows coupling with TM3. Even though these modes represent antinodes at the observed point A1, modal coupling can lead to interference and closely spaced resonant peaks.

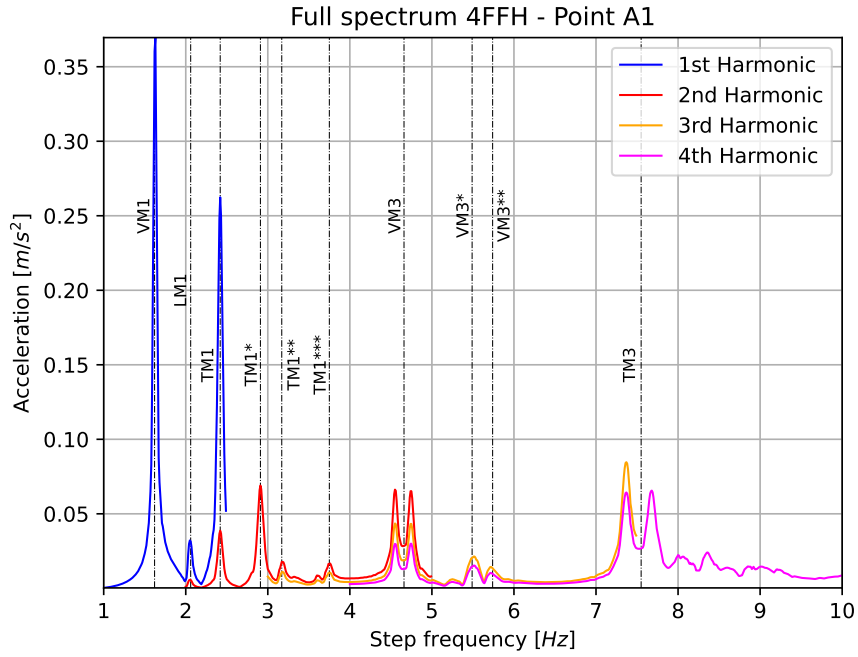


Figure 4.2: Full spectrum of the 4FFH analysis observed at point A1 of the bridge deck , assuming $\xi_{s,min} = 0.2\%$

Figure 4.3 presents the combined contributions according to the square root of sum of squares (SRSS) principle, presenting the response factor according to the ISO 10137 guideline [12] as specified. It is evident that the response to human-induced vibrations is dominated by the first vertical mode VM1, followed by the first torsional mode TM1. The influence of higher harmonics is mainly present via the second harmonic. The conclusion can be drawn that the effect of higher harmonics is small compared to the first harmonic, confirming the statement of EUR23984 as afore mentioned. Furthermore, the contribution of higher modes is small compared to the first modes due to a lower dynamic load factor at higher frequencies and the squared total response.

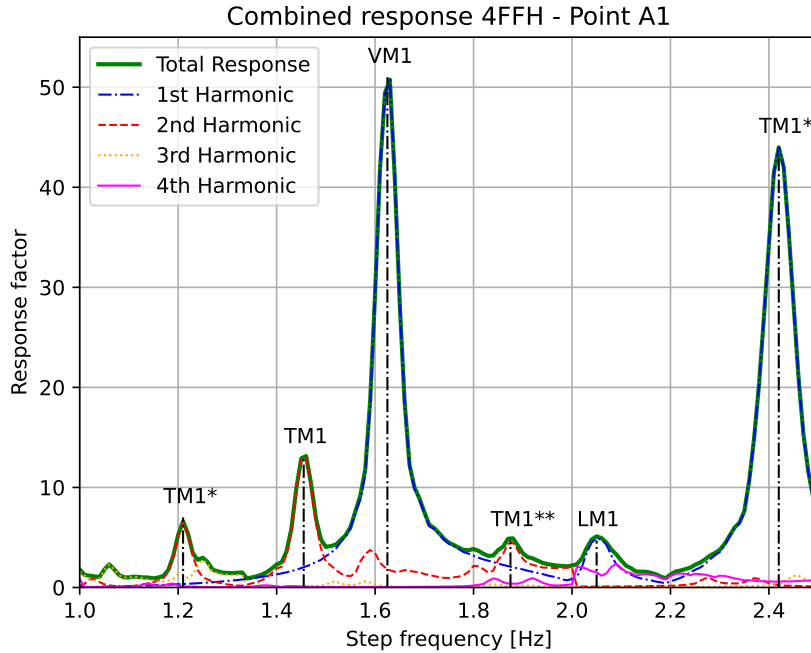


Figure 4.3: Combined response of the 4FFH analysis observed at point A1 of the bridge deck, assuming $\xi_{s,min} = 0.2\%$

According to the 4FFH analysis, a response factor limit of $R \leq 64$ is specified for outdoor bridges, making the case study compliant. However, it should be noted that the method accounts for single moving loads, for which primarily the perception of other users of the structure is verified. An example would be an indoor office environment, where vibrations induced by individual co-workers do not lead

to a loss of comfort. The methodology does not address vibrations caused by large groups, nor does it account for their comfort under complex stochastic loading conditions. Instead, the 4FFH method is designed to evaluate the influence of a structure's mode shapes and their contribution to higher harmonics, making it a qualitative tool for assessing human-induced vibrations.

4.2 Pedestrian crowds

The dynamic loading induced by crowds is presented here. Both the SDOF and transient analysis serve as assessment methods. It is important to note that the loading is stationary since the assumption is made that the amount of people that enter and leave the bridge is equal. The load is applied according to its mode shape, representing a harmonic area load that matches the eigenfrequency of the structure according to the equivalent amount of pedestrians.

4.2.1 SDOF

The Single Degree of Freedom (SDOF) method, as described in 2.6.1 - SDOF method, offers a straightforward approach to assessing human-induced vibrations. Through modal analysis, eigenfrequencies within the critical range are identified, and the maximum response under full resonant behaviour is calculated. The method simplifies the analysis by assuming that the response of the SDOF system under full resonance exceeds the total response of the structure. However, this assumption seems shortcoming, since the total response can either be constructive or destructive via the superposition of the eigenmodes. Consequently, the SDOF method is primarily used to estimate the magnitude of accelerations resulting from pedestrian-structure interaction, rather than serve as a means of actual verification. It can be utilized during early design stages to determine whether pedestrian-structure interaction is significant and requires further consideration in later design stages.

Furthermore, does the use of modal analysis require that the system behaves linearly, meaning that geometric and material non-linear effects are not accounted for. Additionally, obtaining accurate modal damping values can be challenging, requiring testing (i.e. logarithmic decrement or half-power bandwidth method), measurements from literature of comparable structures, or the use of sophisticated FEM simulations. The maximum deck accelerations for the critical modes, subjected to loading by pedestrian streams under design situation three, are summarized in Table 4.3.

Mode shape	Traffic class	p^* [N]	m^* [kg]	a_{max} [m/s ²]
1st vertical	TC1	381.61	47,454	2.010
	TC2	539.73	48,724	2.769
	TC3	853.02	51,797	4.117
	TC4	16,599	56,904	20.57
2nd vertical	TC1	220.56	47,489	0.290
	TC2	316.18	48,736	0.405
	TC3	499.96	51,797	0.603
	TC4	2,315.7	56,902	2.583

Table 4.3: Maximum deck accelerations for the critical vertical modes, assuming $\xi_{s,min} = 0.2\%$

Only vertical modes are presented, as the FEM software does not support determining the modal load for torsional mode shapes. This limitation arises because, in SOFiSTiK, eigenvectors are normalized such that the generalized modal mass equals one to simplify algebraic equations. To calculate the modal load, it must be scaled by a proportionality factor α_k , as defined in equation 4.1. However, when the maximum rotation is used to derive the proportionality factor α_k for torsional modes, the results become unreliable, producing responses up to ten times higher than the vertical accelerations. This could be explained by the large contribution of lateral displacements in addition to rotational effects.

$$\alpha_k = \frac{1.0}{\max_{1 \leq j \leq n} (\phi_{j,k})} \quad (4.1)$$

where:

α_k = proportionality factor
 $\max_{1 \leq j \leq n} (\phi_{j,k})$ = maximal component of the original eigenvector for node j at mode k

4.2.2 Direct time

The most elaborate means of vibration assessment is via transient, numerical (direct) time integration. An extensive outline of the assessment method is provided in 2.6.4 - Transient numerical analysis in FEM. The total response of all modes is obtained, contrary to the SDOF or response spectrum method utilising modal analysis. Furthermore, it enables geometric and material non-linear analysis, as linear modal analysis is no longer necessary. The Newmark Beta method with coefficients $\gamma = \frac{1}{2}$, $\beta = \frac{1}{4}$ is selected, whilst the account is made for geometric non-linearity of the suspension bridge system.

The selection of an adequate time step is of great importance to accurately determine the dynamic response. Transient simulations should generally have twenty time steps for the highest frequency of interest. However, the SYNPEX guideline [6] states a sensitivity analysis regarding step size Δt should be performed to ensure accurate results. Figure 4.4 presents the acceleration response for the first vertical mode of the case study for various time steps. Observed can be that a large time step $10/f_{max}$ results in no clear resonance peak, whereas a decrease in step size provides convergence to a steady state response. For all transient analyses in this study, a step size of $30/f_{max}$ is applied to ensure accurate results whilst keeping computational time limited.

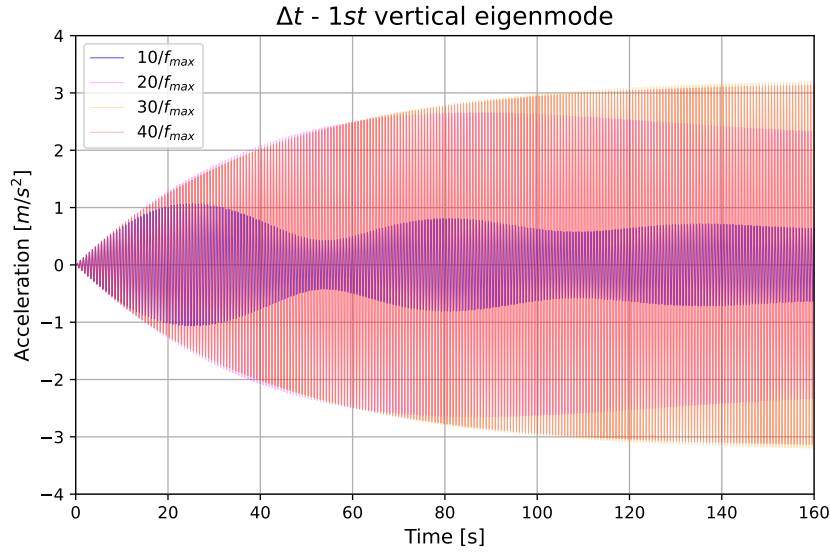


Figure 4.4: Time step sensitivity analysis of the first vertical eigenmode, assuming $\xi_{s,min} = 0.2\%$

In addition to determining the maximum acceleration for each mode of interest, provides the analysis insight into the (non-linear) behaviour of the structure over time. According to the theory, the loading should be applied under resonance conditions, where $f_{step} = f_n$, thereby exciting the structure in the mode shape of interest. Figures 4.5 to 4.8 illustrate the transient response and Fast Fourier Transform (FFT) of the critical modes for traffic class three (TC3).

The torsional mode shapes exhibit a beating behaviour. To explain this, a Fast Fourier Transform (FFT) analysis is performed to investigate whether the response arises from modal coupling and/or non-linear effects. The analysis reveals that both torsional modes display a secondary amplitude at approximately $f = 2.91 [Hz]$. This is attributed to the excitation of the torsional eigenfrequency of the pylons by the torsional crowd loading. The transient analysis further shows interference effects caused by the phase lag between the deck and the pylons, leading to the observed beating pattern. Conversely, the vertical modes are nearly in complete resonance, with no significant energy content at other frequencies.

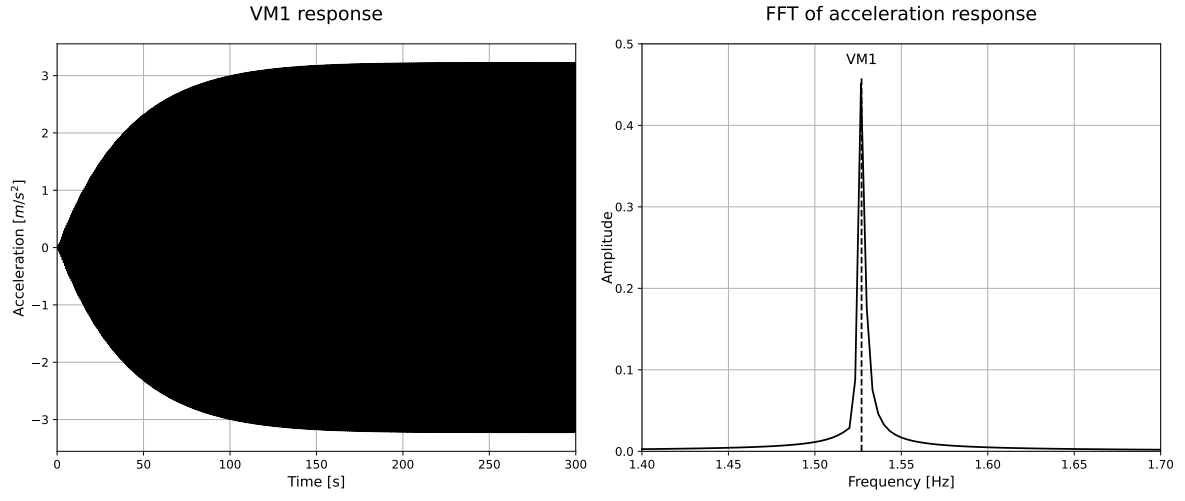


Figure 4.5: Transient response of the first vertical mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$

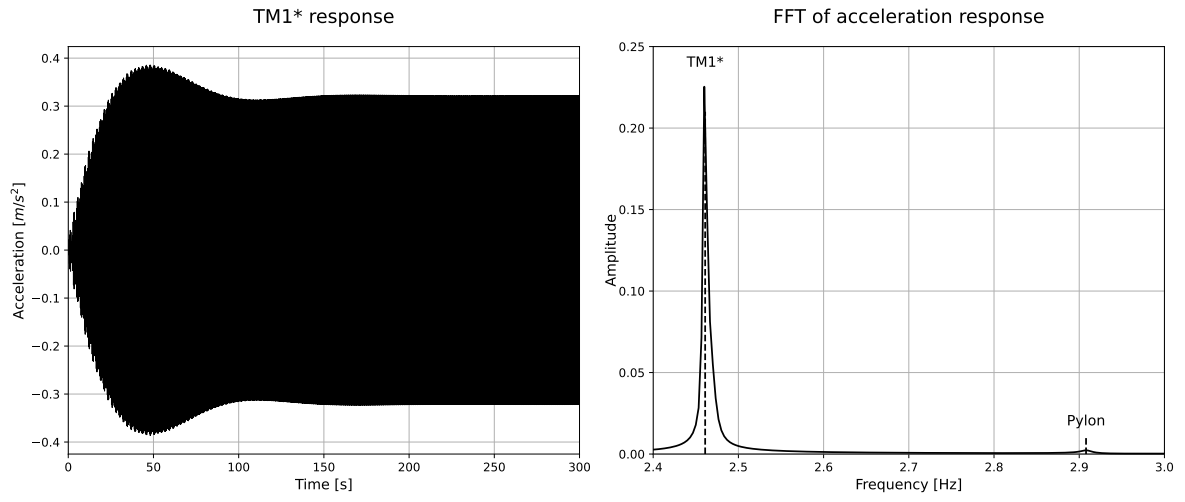


Figure 4.6: Transient response of the first torsional* mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$

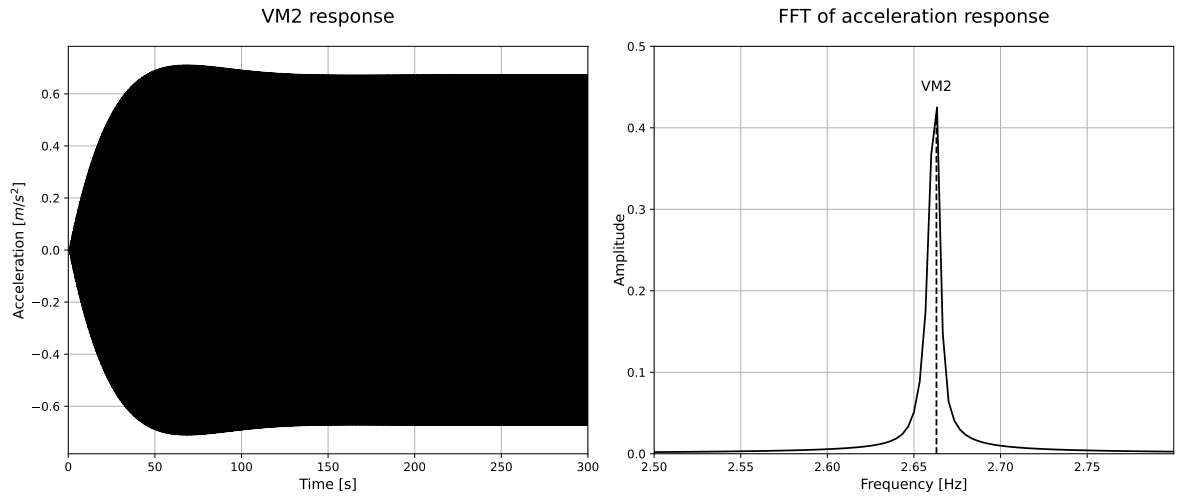


Figure 4.7: Transient response of the second vertical mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$

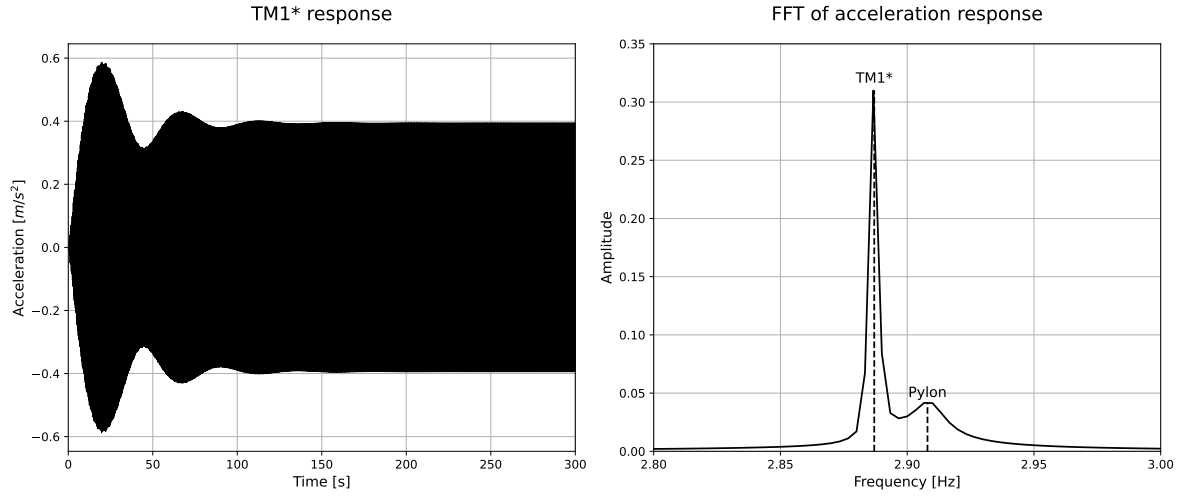


Figure 4.8: Transient response of the first torsional mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$

Table 4.4 presents the maximum accelerations obtained in the original design via Oasys GSA and the parametric model in SOFiSTiK regarding transient analysis for all eigenmodes assuming $\xi_{s,min} = 0.2\%$ damping. The first torsional* mode shape is excluded since this mode was not obtained through modal analysis in Oasys GSA. It shows that the original design and the parametric model yield comparable results.

mode shape	traffic class	org. design - OasysGSA		param. model - SOFiSTiK	
		eigenfrequency [Hz]	max. acc. [m/s ²]	eigenfrequency [Hz]	max. acc. [m/s ²]
1st vertical	TC1	1.594	1.41	1.592	1.46
	TC2	1.568	1.73	1.576	1.91
	TC3	1.515	3.23	1.529	3.34
	TC4	1.437	15.2	1.458	15.45
2nd vertical	TC1	2.784	0.49	2.770	0.48
	TC2	2.751	0.61	2.740	0.62
	TC3	2.681	0.66	2.662	0.73
	TC4	2.556	3.145	2.540	3.31
1st torsional	TC1	2.872	0.71	2.878	0.53
	TC2	2.849	1.01	2.852	0.63
	TC3	2.801	1.54	2.843	1.35
	TC4	2.752	13.20	2.780	10.8

Table 4.4: Results of the transient analysis of both FE models due to dynamic loads of pedestrian streams for all specified traffic classes, assuming $\xi_{s,min} = 0.2\%$

4.2.3 Sledgehammer tests

In addition to the SDOF and direct time method, sledgehammer tests were performed at three meters from midspan. Accelerometers have been placed at 64 nodes evenly spaced across the bridge deck and the acceleration response for a unit force has been measured. A reference area of $A_{ref} = 58m^2$ is assumed to have the same acceleration as the excited node. By applying the equivalent crowd-loading to the acceleration response for a unit load in accordance with the reference area, a total response is obtained. Table 4.5 presents the response of the measured critical modes at the reference node.

mode shape	a_{max} [m/s ²]
1st vertical	2.561
2nd vertical	0.375
1st torsional	0.680

Table 4.5: Acceleration response derived from sledgehammer tests

4.2.4 Comparison

Table 4.6 shows the acceleration responses for both assessment methods. Observed can be that results are not comparable. This is expected since the SDOF method is deemed shortcoming, only to be used to get an estimate for the magnitude of accelerations. The response of the first critical vertical mode shows an overestimation, whereas the second vertical mode depicts an underestimation of the acceleration response.

mode shape	traffic class	SDOF [m/s ²]	direct time [m/s ²]	difference [%]
1st vertical	TC1	2.010	1.462	37
	TC2	2.769	1.908	45
	TC3	4.117	3.340	23
	TC4	20.57	15.45	33
2nd vertical	TC1	0.290	0.481	-39
	TC2	0.405	0.623	-34
	TC3	0.603	0.727	-17
	TC4	2.580	3.310	-22

Table 4.6: Comparison SDOF and direct time assessment methods, assuming $\xi_{s,min} = 0.2\%$

The measured results from sledgehammer tests can only be compared to the direct time method, as the response is recorded at a node excited three meters from midspan, while the SDOF method assumes full resonant response at the antinodes based on the mode shape. For the second vertical mode, maximum displacement occurs approximately at a quarter of the span, making direct comparison impractical. However, the direct time analysis allows for comparison with measured results, as it can evaluate the response at any specific node of interest—in this case, three meters from midspan.

Table 4.7 presents the results of the direct time method and the measurements. For the direct time method, results are provided for minimum structural damping, $\xi_{s,min} = 0.2\%$ and average structural damping, $\xi_s = 0.4\%$. The findings show that for the first vertical mode, there is close agreement between the acceleration response obtained using the average damping value ($\xi_{s,avr} = 0.4\%$) and the measured results. In contrast, results for higher modes are not directly comparable.

These results indicate that applying the average structural damping value is sufficient for practical applications of steel bridges. However, the use of Rayleigh damping is less accurate for higher modes, as it overestimates the response.

mode shape	sledgehammer [m/s ²]	direct time [m/s ²]	
		$\xi_{s,avr} = 0.4\%$	$\xi_{s,min} = 0.2\%$
1st vertical	2.561	2.515	3.430
2nd vertical	0.375	0.583	0.731
1st torsional	0.680	1.353	1.721

Table 4.7: Comparison direct time method and measured results sledgehammer testing

4.3 Moving loads

Besides crowd loading, moving loads in the form of pedestrians/hikers and joggers should be assessed. The codes and guidelines [5], [4], [23] mention the SDOF and direct time method to be utilised. It should be noted that moving loads are primarily dictated by the passing time to cross the bridge, see equation 4.2. A slower walking velocity will lead to a longer passing time in which greater resonance build-up can occur. The assumption is made that the effects of synchronised crowd-loading will have a more profound effect on maximum deck accelerations than the response of moving loads induced by pedestrians and joggers.

$$t_{pass} = L/v \quad (4.2)$$

where:

t_{pass} = passing time to cross the bridge
 L = length of the bridge deck
 v = walking velocity of the user ($v_{ped} = 1.7 \text{ m/s}$, $v_{jog} = 3.0 \text{ m/s}$)

Critical velocity can play a major role in moving load analysis. The critical velocity relates to the wavelength of the dynamic response, which is a spatial property. If the load's movement causes the structure's deflection pattern to match this wavelength, (spatial) resonance occurs. Equation 4.3 denotes the critical velocity of the structure. It becomes evident that the moving load velocity is not near the first vertical critical value ($v_{cr,1} = 210 \text{ m/s}$) and the influence of (spatial) resonance is marginal. However, the load model does account for full resonance conditions for the harmonic forcing equalling the eigenfrequency $f_s = f_n$, meaning resonance still occurs being influenced by the force in time.

$$v_{cr,n} = \frac{2L}{n} * f_n \quad (4.3)$$

where:

$v_{cr,n}$ = critical velocity of the bridge for mode n
 L = length of the bridge deck
 n = wave number of mode n
 f_n = eigenfrequency of mode n

4.3.1 SDOF

It should be noted that the SDOF method assumes a steady-state response under full resonant conditions, excluding the effect of a moving load. Table 4.8 presents the results of the SDOF method for the first vertical mode under traffic class three (TC3). Only one traffic class and critical mode are considered, as it is assumed that crowd-loading dominates, only providing context to the effect of moving loads.

mode shape	type	$p^* [N]$	$m^* [kg]$	$a_{max} [m/s^2]$
1st vertical	pedestrians	866.45	46,158	4.693
	joggers	990.23	46,158	5.364

Table 4.8: Steady-state response of moving loads for the first vertical mode, assuming $\xi_{s,min} = 0.2\%$

Observed can be that the response for pedestrians and joggers yields higher results than the equivalent crowd-loading as proposed previously. However, the assumption is made that the steady-state response is reached, which is not directly apparent in the case study. It would suffice to either expand the SDOF theory to account for the full transient response and obtain the maximum response for the moving load during its passing time t_{pass} , or use the direct time method to obtain the transient response directly as outlined in the following section.

4.3.2 Direct time

In a moving load analysis, the direct time method takes into account the effects of the moving load and the passing time of the bridge. Figure 4.9 shows the transient response for pedestrians and joggers. It can be seen that the maximum acceleration response is obtained at different time values because the walking speed of the pedestrians and joggers differs, the response is asymmetric since damping is present. It can likewise be seen that the acceleration response is greater for pedestrians than for joggers. This could be explained by the fact that the pedestrians have a longer time to pass, providing a longer period for the resonance to build up.

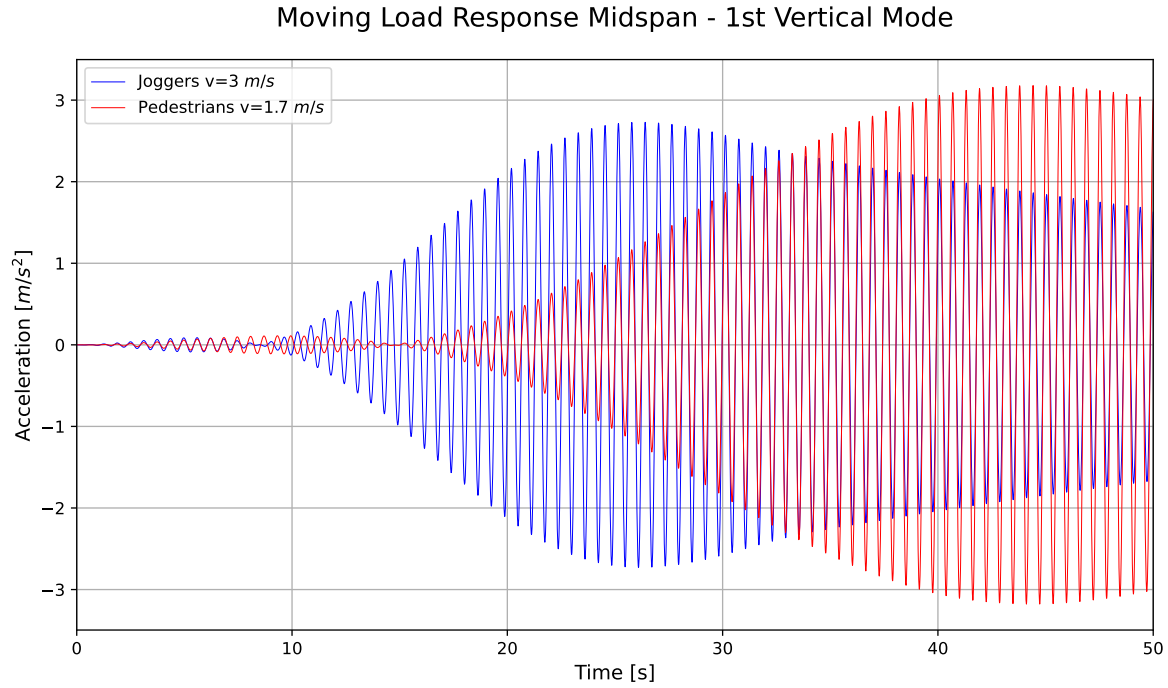


Figure 4.9: Transient response of pedestrians and joggers for the first vertical mode according to TC3, assuming $\xi_{s,min} = 0.2\%$

4.3.3 Comparison

Table 4.9 shows the results of the SDOF and direct time methods for moving loads and provides results for the equivalent crowd load concerning TC3. It can be seen that the SDOF method gives an overestimation of the response as it does not take into account the moving nature of the load, which explains the higher acceleration response for joggers than pedestrians. For the direct time method, the responses are smaller, with the response for pedestrians being greater than that for joggers. When the responses are compared with the equivalent crowd load, it is clear that they are within close range, meaning that the crowd load is dominant, but not by a significant margin. This leads to the recommendation to include moving load analysis for all traffic classes in the optimisation procedure outlined in section 5.5 - Optimisation procedure case study.

type	load [kN]	number	DLF	velocity [m/s]	SDOF [m/s ²]	direct time [m/s ²]
pedestrians	2.24	8	0.35	1.7	4.693	3.182
joggers	2.5	2	1.56	3.0	5.364	2.721
crowd	2.19 (eq.)	-	0.35	-	4.117	3.430

Table 4.9: Comparison moving load analysis SDOF and direct time method plus equivalent crowd-loading analysis for TC3, assuming $\xi_{s,min} = 0.2\%$

Part III

Geometric optimisation to reduce human-induced vibrations and external damping reliance

5

Optimisation

5.1 Methodology

Engineers must exercise caution when formulating optimisation problems, as the structure of the problem significantly impacts the effectiveness. To ensure meaningful and reliable results, the methodology outlined in figure 5.1 is adopted. This approach is rooted in the principles of Engineering Design Optimization [52], which offers an extensive framework for guiding the optimization process toward effective solutions.

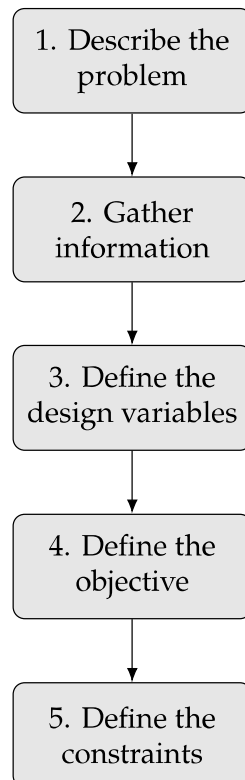


Figure 5.1: Optimisation methodology [52]

One starts by describing the system and outlining the goals and requirements. The case study represents a footbridge with a suspension bridge system incorporating TMDs to mitigate pedestrian-structure interaction to an acceptable level. Four design situations are specified to assess the pedestrian's dynamic behaviour, which occurs via transient analysis of dynamic loads according to the eigenfrequencies within the critical range. Rather than performing analysis for all design situations, only design situation three is considered as outlined in the table below. This scenario represents dense pedestrian traffic under medium comfort conditions, providing a practical and realistic basis for analysis. Restricting the assessment to a single design scenario reduces the number of dynamic analyses required for verification.

Situation	Type	Class	Characteristics	Context
3	Traffic	TC3	0.5 persons / m ²	Still unrestricted walking, overtaking inhibited
	Comfort	CL2	0.50 m/s ² ~ 1.00 m/s ² ² 0.10 m/s ² ~ 0.30 m/s ² ³	Medium comfort

Table 5.1: Design situation three considered for assessment of the case study, see 3.8 - Design situations.

A structural damping value of $\xi_{s,min} = 0.2\%$ is considered as outlined in 3.7 - Structural damping. The damping has been applied in the form of Rayleigh damping with one damping value to all modes within the critical range.

The main goal is to reduce or eliminate the need for TMDs by adjusting the structure’s geometric parameters, primarily through mass reduction. While minimizing acceleration may seem impractical due to compliance constraints, understanding to what degree deck acceleration can be reduced and which geometric parameters are of influence. Another objective is to reduce the structure’s mass, benefiting both cost and sustainability. The structure must remain compliant to ultimate limit state requirements in the optimal solution to ensure no physically unfeasible designs are produced. Information for the case study is gathered from the original design, including a FEM model, static and dynamic analysis reports, and TMD manufacturing data. Additionally, a parametric model is developed to support the geometric optimization. The design variables, objectives, and constraints significantly influencing the optimization outcome, are discussed in more detail.

5.2 Design variables

Understanding geometric modifications by their static and dynamic effects is crucial for effective optimization. Human-induced vibrations are primarily influenced by the reduction coefficient ψ , as it quantifies the magnitude of dynamic loading, which is governed by the eigenfrequencies of the structure. Successful optimization depends on identifying design variables that influence these eigenfrequencies and understanding how to modify their characteristics. For instance, increasing the structural mass lowers eigenfrequency, whereas increasing the stiffness has the opposite effect. Beyond eigenfrequencies and the resulting dynamic loads, there are other design variables which can have profound effects. Notably, changes in the static scheme of the structure might have minimal impact on eigenfrequencies but can effectively reduce deck accelerations, think of an increase in the amount of vertical hangers.

In this section, the design variables which have been selected for optimisation are presented. The selection provides a broad range in both static and dynamic properties of the structure, since one is intrinsically linked to another. The effectiveness of the parameter selection will be assessed based on the evaluation of the optimization results, potentially providing insights for more refined analyses and recommendations.

5.2.1 Pylon height

Modification of the pylon height in a self-anchored suspension bridge system results in a change of cable sag f , influencing both static and dynamic properties. Lower pylon height will result in lower sag, increasing horizontal force in the main cable and anchorage component to the girders. The vertical component of the tension force will reduce, leading to decreased vertical cable stiffness and thus eigenfrequency. Figure 5.2 shows the effect of a change in pylon height for the case study. Further details on the characteristics of self-anchored suspension bridges are provided in 3.3.2 Self-anchored theory.

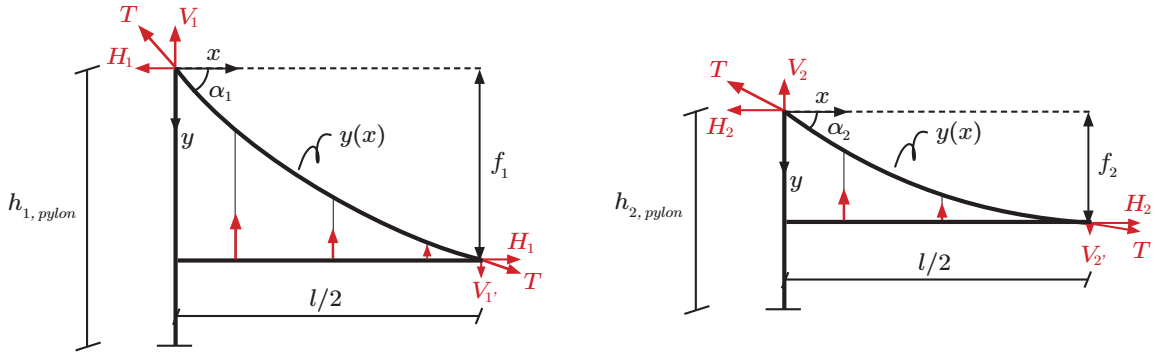


Figure 5.2: Schematic representation of pylon height influence of the case study

It can be observed that a greater pylon height $h_{1,pylon}$ results in a smaller horizontal force H_1 and greater cable sag f_1 , for which the resultant vertical force ($V_1 - V_{1'}$) in the main cable is greater. Conversely, a decrease in pylon height $h_{2,pylon}$ results in a greater horizontal force H_2 , smaller resultant vertical force ($V_2 - V_{2'}$) and decreased cable sag f_2 . Vertical eigenfrequencies will shift upward for increased pylon height due to the increased resultant vertical force in the main cable, increasing the structure's total stiffness. Additionally, due to the self-anchored nature of the suspension bridge, a reduction in pylon height increases compressive forces in the girders due to the larger horizontal force. Variations in pylon height significantly affect both the static and dynamic behaviour of the bridge, making it a critical geometric parameter. Equation 5.1 shows the design variable.

$$10 \leq h_p \leq 18 \text{ [m]} \quad (5.1)$$

5.2.2 Number of hangers

The number of vertical hangers connecting the main cable system to the girders significantly influences the deck accelerations. Resonance occurs when the step frequency aligns with the structure's natural frequency. Restricting the deck's vertical motion at specific points reduces acceleration by impeding movement in its true eigenmode. Figure 5.3 illustrates this concept schematically.

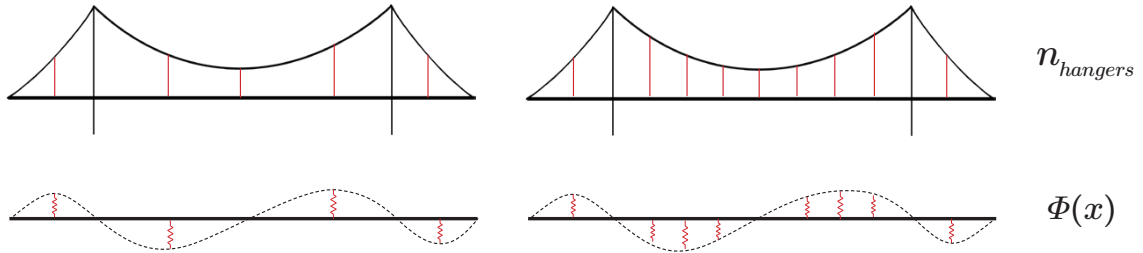


Figure 5.3: Schematic representation of the influence of hangers of the case study.

Important to note is that altering the number of hangers does not affect the structure's eigenfrequency. The main cable's shape remains unchanged, preserving its stiffness. While increasing the number of hangers introduces only minor changes to the mass and stiffness, it proves beneficial for the static behaviour of the main girders. With more contact points to the main cable system, bending moments and deflections are reduced, resulting in a more effective parallel structural system. The hanger configurations applied for the case study are presented in figure 5.4. By ensuring an even spacing of hangers will the static scheme of the main girder remain optimal.

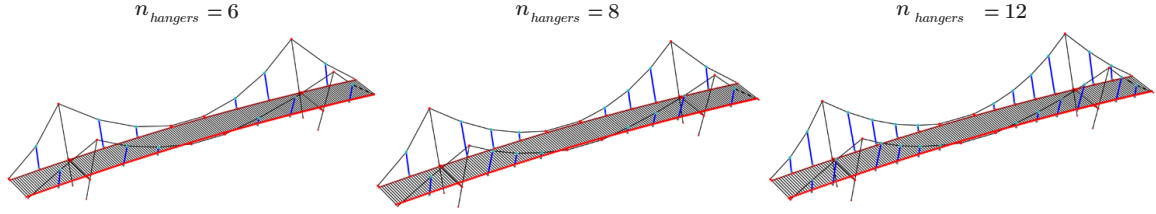


Figure 5.4: Hanger configurations considered for the case study.

The design variable $n_{hangers}$ contains discrete values and is defined by equation 5.2. The jump from eight to twelve hangers is explained by the side span containing two vertical hangers, requiring more hangers at midspan to ensure an even bending moment line distribution along the length of the bridge.

$$n_{hangers} = [6, 8, 12] \quad (5.2)$$

5.2.3 Main girder

A change in girder dimensions results in modified mass and stiffness, influencing both static and dynamic properties of the structure. Important to note is that the box girder of the original case study is identified as a class four cross-section, requiring linear-elastic material analysis. This has been applied to all iterations to simplify analysis. Figure 5.5 presents the original girder design of the case study.

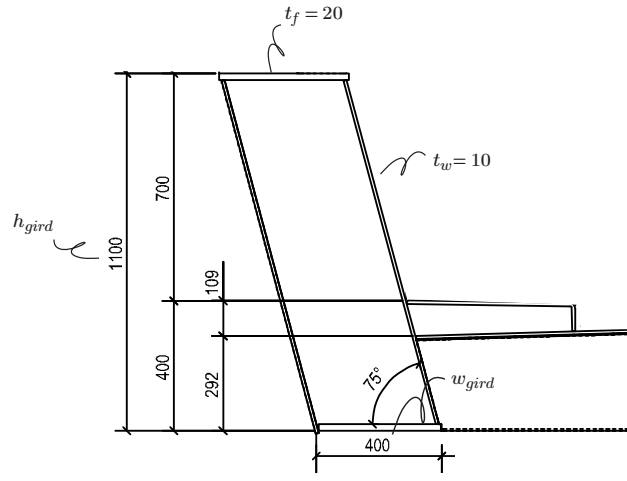


Figure 5.5: Main girder design of the original case study.

Two parameters are introduced, namely the height h_{gird} and width w_{gird} of the main girder. The ratio between height and width is remained equal to retain its original shape. Equation 5.3 presents the continuous parameters and their bounds within the design space.

$$\begin{aligned} 700 < h_{gird} < 1500 \text{ [mm]} \\ 225 < w_{gird} < 500 \text{ [mm]} \end{aligned} \quad (5.3)$$

5.3 Objectives and constraints

Before optimization can take place, a distinction must be made between objectives and constraints. Objectives define specific targets that should be either minimized or maximized. Constraints refer to the restriction of the design space, limiting certain design solutions to a specific value via an equality denoted as $h(x) = 0$ or an inequality, denoted as $g(x) \leq 0$. Engineering optimisations know many forms

of constraints enabling us to find useful solutions with physical meaning. "An example of the distinction between objectives and constraints is to minimize the stress in a structure, but this would inevitably result in an over-designed, heavy structure. Instead, we might want minimum weight (or cost) with sufficient safety factors in stress, which an inequality constraint can enforce [52]".

Multi-objective (MO) optimization aims to find the best solution for multiple objectives simultaneously. However, this is in theory not entirely possible, as only one truly optimal solution exists. Instead, MO optimization is a process of balancing trade-offs between different metrics, offering engineers valuable insights into their relationships and enabling them to identify the most suitable solution. The concept of Pareto optimality is often used in MO optimisation and refers to a concept used to identify solutions that are non-dominated for all objectives. A solution is considered Pareto optimal if there is no other solution that improves at least one objective without worsening another. In other words, a Pareto optimal solution represents the best trade-off between the objectives. Figure 5.6 presents the concept of Pareto dominance, whereas figure 5.7 presents the Pareto front, which is a depicted by all solutions being non-dominant.

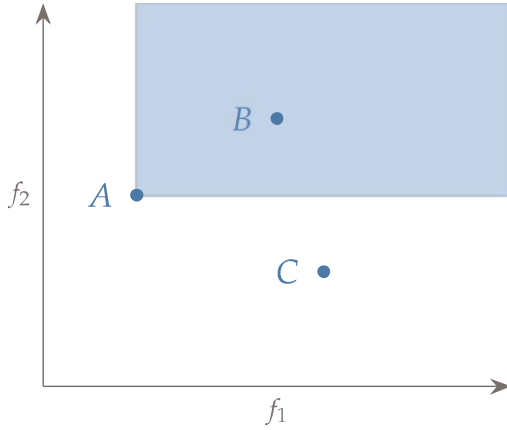


Figure 5.6: Three designs, A, B, and C, are plotted against two objectives, f_1 and f_2 . The region in the shaded rectangle highlights points that are dominated by design A [52]

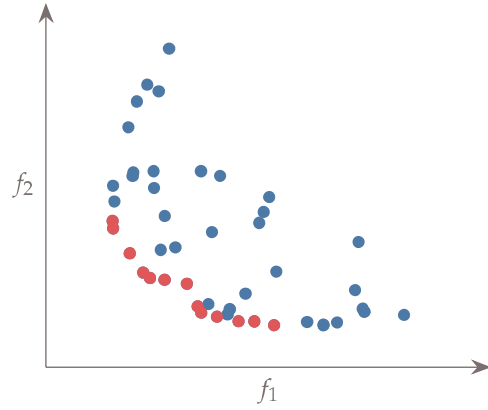


Figure 5.7: Pareto front of a MO optimisation, Pareto optimal solutions are denoted in red [52]

The total mass and maximum deck accelerations are two objectives considered to be minimized in the case study, see equation 5.4. The objectives are related since a structure's total mass has direct implications for accelerations. However, since the design variables provide stiffness and geometric changes is expected that the minimized outcome will not reveal a direct correlation.

$$\text{minimize } f(\bar{x}) = \begin{bmatrix} f_{\text{mass}}(\bar{x}) \\ f_{a.\text{max}}(\bar{x}) \end{bmatrix} \quad (5.4)$$

where:

- $f_{\text{mass}}(\bar{x})$ = total structural mass
- $f_{a.\text{max}}(\bar{x})$ = maximum deck acceleration
- \bar{x} = vector representing a unique solution for all design variables

To validate if a solution dominates, an objective function, in other words, fitness, is constructed. This function shows how the solution compares and will be used to either minimize or maximize to obtain an optimal solution. MO optimisation contains various optimisation functions, most notably the weighted sum, epsilon-constraint method and normal boundary intersection are utilised by engineers. These methods optimise one objective whilst constraining the remaining objectives in some form. To illustrate the concept of an objective function is the weighted sum method presented, combining all objectives into one, per equation 5.5. Conceptually this represents a line, for which the slope is selected by determination of weights for each objective. Iteration of the weights is required to construct the Pareto front.

$$f(\bar{x}) = \sum_{i=1}^{n_f} w_i f_i(\bar{x}) \quad (5.5)$$

where:

- w_i = weight for each objective ($w_i \in [0, 1]$)
- $f_i(\bar{x})$ = individual objective, ought to be normalised for comparison
- \bar{x} = vector representing a unique solution for all design variables

The weighted sum method knows limitations, since the weights w_i to be assigned are difficult to estimate, requiring (many) iterations to obtain the Pareto front. Furthermore, it will lead to inconsistent spacing, generating solutions only for the convex portions of the objective function, which is explained by figure 5.8.

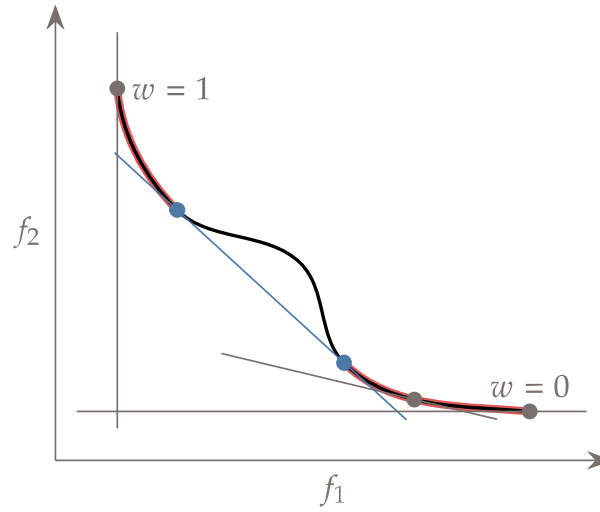


Figure 5.8: Weighted sum method as a function for objectives f_1 and f_2 , with results for the convex parts of the function, showing inconsistent spacing along the Pareto front [52]

Since the epsilon-constraint and normal boundary intersection objective functions apply constraints to one or more objectives, it is preferable to use an approach where objectives are equal. This eliminates the estimation of constraint values and costly iterations to construct the Pareto front. Evolutionary algorithms can be utilised for direct application of Pareto dominance, meaning no specific fitness definition is required resulting in a direct approach. If no domination is obtained after particle evaluation, the particle is excluded from the Pareto front. Pareto dominance logic for an evolutionary algorithm is explained below.

Pareto Dominance

For two solutions $A = (f_1^A, f_2^A, \dots, f_n^A)$, $B = (f_1^B, f_2^B, \dots, f_n^B)$

A dominates B if:

$$f_i^A \leq f_i^B \quad \forall i \quad (\text{A no worse than B in all})$$

and:

$$f_i^A < f_i^B \quad \exists i \quad (\text{A better than B at least once})$$

It is important to note that Pareto dominance requires normalisation of objectives to ensure they are treated equal. The difference in magnitude between total mass and deck accelerations would make

for an incorrect fitness evaluation. Min-max normalisation is applied to both objectives, for which the lower and upper bound values have been determined before optimisation. Equation 5.6 shows this normalisation, the optimisation considers greater values if these are obtained during the iteration process.

$$\frac{f_i(\bar{x}) - f_{i,min}(\bar{x})}{f_{i,max}(\bar{x}) - f_{i,min}(\bar{x})} \quad (5.6)$$

where:

$$\begin{aligned} f_i(\bar{x}) &= \text{objective value } i \\ f_{i,min}(\bar{x}) &= \text{minimum objective value obtained in the optimisation} \\ f_{i,max}(\bar{x}) &= \text{maximum objective value obtained in the optimisation} \end{aligned}$$

Two constraints are applied in the case study to limit design solutions to meaningful designs. The first constraint is presented in equation 5.7 and enforces strength verification of the structure in the ultimate limit state for all load cases present in the bridge by its respective unity check using a penalty function. Appendix B - Load cases & combinations provides the load cases and combinations used for ULS verification. A linear transition occurs for unity checks between 0.50 and 1.0, promoting feasible strength designs. At 1.0 a sharp transition is implemented to penalize physically impossible designs. It is important to note that alternative penalty functions could be developed, featuring higher-order and smoother transitions. However, it is advisable to first conduct optimizations using the simplest form of the penalty function and adjust constraints as necessary. The penalty value of 1.0 is used because the objectives are normalized within their respective design space, meaning that a penalty function value of 1.0 will have the same effect as the worst feasible solution.

$$p_{uc,ULS}(\bar{x}) = \begin{cases} 1.0 & \text{if } \bar{x} < 0.5 \\ 1 - \bar{x} & \text{if } 0.5 \leq \bar{x} < 1.0 \\ 1.0 & \text{if } \bar{x} \geq 1 \end{cases} \quad (5.7)$$

where:

$$\begin{aligned} p_{uc,ULS}(\bar{x}) &= \text{constraint in the form of a penalty function as ULS unity check} \\ \bar{x} &= \text{feasible design option in the specified domain} \end{aligned}$$

The second design constraint regards the number of hangers n_{hang} , see equation 5.8. The penalty function provides a higher value for hanger configurations with a greater amount of hangers. This function encourages fewer hangers, resulting in less connection detailing and a reduction of costs. Once again the maximum penalty value of 1.0 is applied since the objectives are normalized within their respective design space, meaning that a penalty function value of 1.0 will have the same effect as the worst feasible solution.

$$p_{hang}(\bar{x}) = \begin{cases} 0.0 & \text{if } \bar{x} \leq 6 \\ 0.5 & \text{if } 6 < \bar{x} < 8 \\ 1.0 & \text{if } \bar{x} \geq 8 \end{cases} \quad (5.8)$$

where:

$$\begin{aligned} p_{hang}(\bar{x}) &= \text{constraint in the form of a penalty function for hanger configurations} \\ \bar{x} &= \text{feasible design option in the specified domain} \end{aligned}$$

Optimisation of the case study is performed via an evolutionary algorithm to which the concepts of Pareto dominance are applied. Equation 5.9 is presented as fitness for dominance evaluation including the defined penalty functions.

$$\text{minimize } f(\bar{x}) = \begin{bmatrix} f_{mass}(\bar{x}) + p_{uc}(\bar{x}) + p_{hang}(\bar{x}) \\ f_{a,max}(\bar{x}) + p_{uc}(\bar{x}) + p_{hang}(\bar{x}) \end{bmatrix} \quad (5.9)$$

where:

$$\begin{aligned} f_{mass}(\bar{x}) &= \text{objective function for total mass evaluation} \\ f_{a,max}(\bar{x}) &= \text{objective function for maximum deck acceleration evaluation} \\ p_{uc,ULS}(\bar{x}) &= \text{ULS unity check penalty constraint} \\ p_{hang}(\bar{x}) &= \text{hanger configuration penalty constraint} \end{aligned}$$

5.4 Particle Swarm Optimisation (PSO)

Particle Swarm Optimization (PSO) is a stochastic, population-based genetic and evolutionary optimization algorithm which uses "swarm intelligence". Swarm intelligence describes the collective behaviour of individuals interacting locally with their environment to produce a global best. Inspired by the movement of bird flocks or schools of fish searching for food or navigating toward a destination, PSO models result in complex, coordinated group (swarm) behaviour.

The algorithm exhibits evolutionary characteristics through its mutational behaviour and genetic traits signified by particle generation. PSO is selected for this case study because it effectively explores a significant portion of the design space through the random generation of particles and individual exploration. By utilisation of crowd intelligence, it will converge toward an optimized solution efficiently.

In PSO, the "swarm" consists of design points, also known as particles which navigate an n-dimensional space in search of an optimal solution. Although these particles are essentially design points, their individual histories play a crucial role in the algorithm. Each particle moves through the design space influenced by a velocity vector which is updated based on two factors: the particle's own best position (its personal best) and the swarm's collective best position (the global best). By utilizing this combination of individual experience and shared information, particles iteratively converge toward optimal solutions [52].

A set amount of particles n is stochastically generated within a predefined design space. The position of each particle i for iteration $k + 1$ is updated according to equation 5.10 for a constant artificial time step Δt .

$$x_{k+1}^{(i)} = x_k^{(i)} + v_{k+1}^{(i)} \Delta t \quad (5.10)$$

The velocity of each particle is updated via equation 5.11. Figure 5.9 gives a graphical presentation of the velocity components.

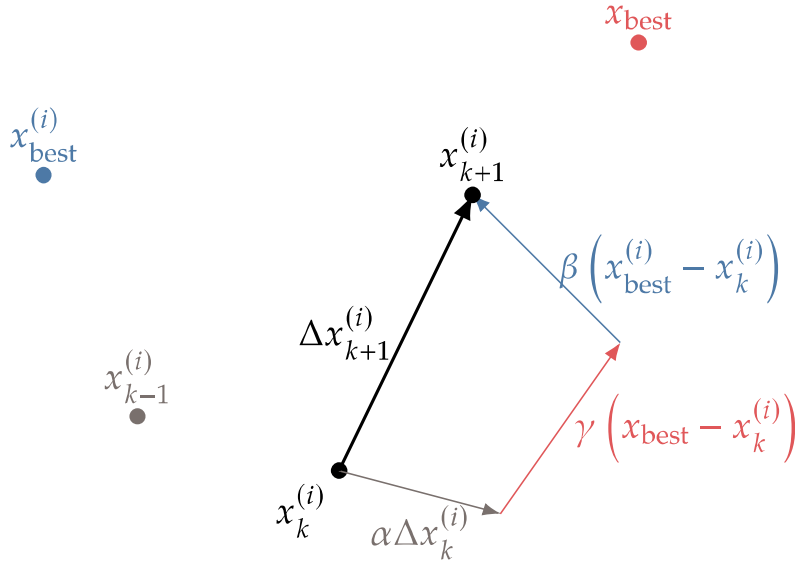


Figure 5.9: Inertia, memory and social components of the velocity vector in PSO[52].

$$v^{(i)} = \alpha \cdot v_k^{(i)} + \beta \cdot \frac{x_{best}^{(i)} - x_k^{(i)}}{\Delta t} + \gamma \cdot \frac{x_{best} - x_k^{(i)}}{\Delta t} \quad (5.11)$$

where:

- $\alpha \cdot v_k^{(i)}$ = Inertia term, where α determines how similar the new velocity is to the previous iteration. The inertia parameter α is usually in the interval of $[0.8, 1.2]$
- $\beta \cdot \frac{x_{best}^{(i)} - x_k^{(i)}}{\Delta t}$ = Memory term, pointing its velocity component to the best position particle i has seen in all its iterations (local best), β consists of a random number in the interval $[0, \beta_{max}]$.
- $\gamma \cdot \frac{x_{best} - x_k^{(i)}}{\Delta t}$ = Social term, directing towards the best point the entire swarm has found, γ is a random number between $[0, \gamma_{max}]$.

Since the time step Δt is artificial, equation 5.11 can be simplified to:

$$x_{k+1}^{(i)} = \alpha \cdot \Delta x_k^{(i)} + \beta \cdot (x_{best} - x_k^{(i)}) + \gamma \cdot (x_{best} - x_k^{(i)}) \quad (5.12)$$

For which equation 5.10 becomes:

$$x_{k+1}^{(i)} = x_k^{(i)} + \Delta x_{k+1}^{(i)} \quad (5.13)$$

The optimization algorithm is described in the following procedure, using a coding-based approach to clarify its functionality

1. Initialize particles

for $i = 1$ **to** n **do**:

a) Generate position $x_0^{(i)}$ within specified domain.

b) Initialize velocity $\Delta x_0^{(i)}$.

end for

2. Main iteration loop

while not converged **do**:

for $i = 1$ **to** n **do**:

if $f(x^{(i)}) < f(x_{best}^{(i)})$ **then**: (Individual best)

$$x_{best}^{(i)} = x^{(i)}$$

end if

if $f(x^{(i)}) < f(x_{best})$ **then**: (Swarm best)

$$x_{best} = x^{(i)}$$

end if

end for

for $i = 1$ **to** n **do**:

$$\Delta x_{k+1}^{(i)} = \alpha \cdot \Delta x_k^{(i)} + \beta \cdot (x_{best}^{(i)} - x_k^{(i)}) + \gamma \cdot (x_{best} - x_k^{(i)}) \quad \text{(Obtain velocity)}$$

$$x_{k+1}^{(i)} = x_k^{(i)} + \Delta x_{k+1}^{(i)} \quad \text{(Update position)}$$

$$x_{k+1}^{(i)} = \max(\min(x_{k+1}^{(i)}, \bar{x}, \underline{x})) \quad \text{(Set bounds)}$$

end for

$$k = k + 1$$

end while

Figure 5.10 presents a Single Objective Particle Swarm Optimisation (SOPSO) of the case study regarding the design variables pylon height h_p and amount of hangers n_{hang} for minimization of deck acceleration as objective. It shows a selection of inertia $\alpha = 0.8$, memory $\beta = 1.5$ and social $\gamma = 0.5$ components, for ten particles and one-hundred iterations. Observed can be a shift in global best, denoted by the magenta dot. Higher iterations show its evolutionary behaviour, converging to a lower deck acceleration since the global best shifts to lower accelerations.

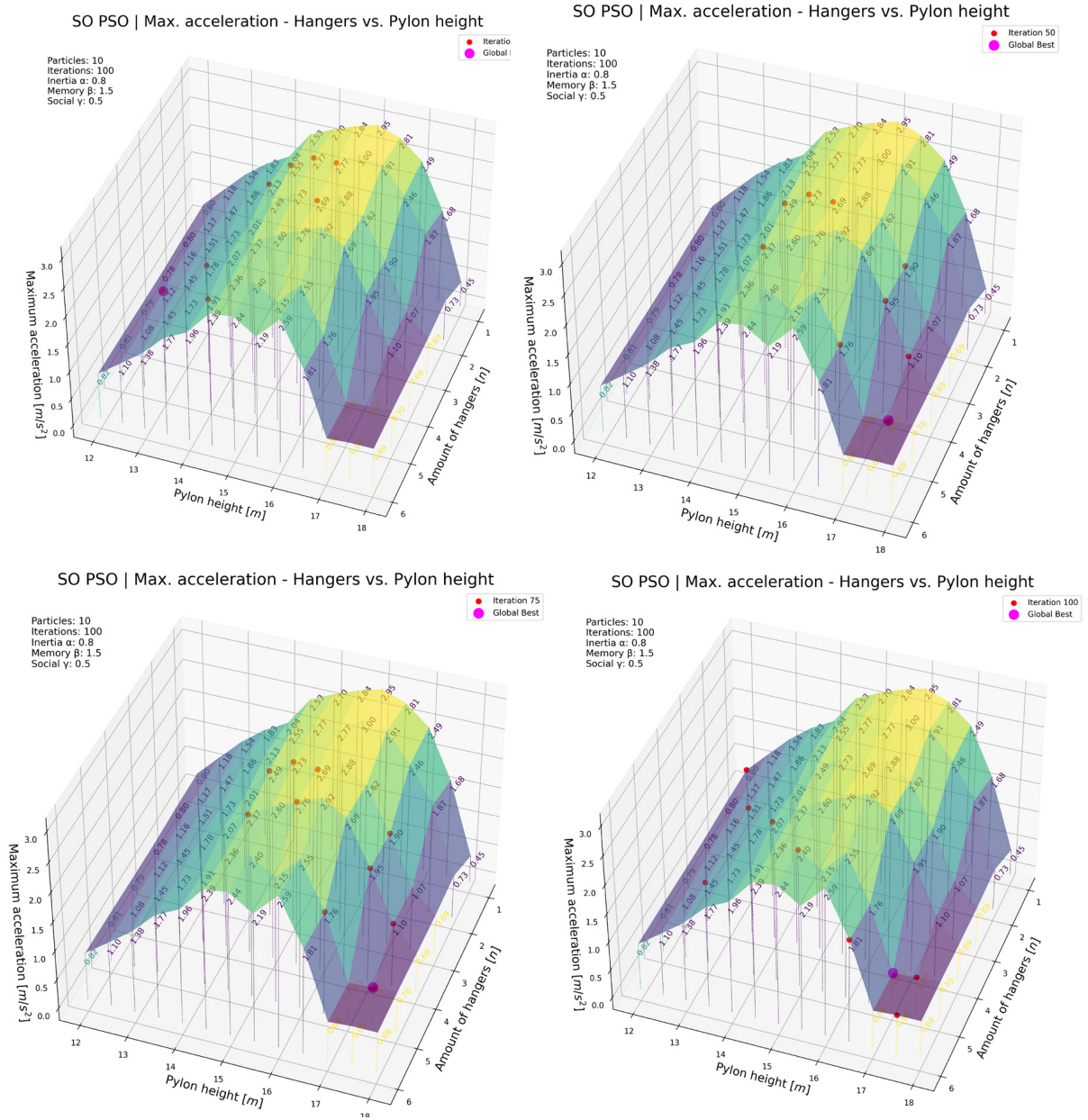


Figure 5.10: Single objective PSO of the case study presenting optimisation for two design variables at iteration 0, 50, 75 and 100 respectively

It is evident that the SOPSO method for the chosen design variables is not particularly effective. Notably, the number of hangers has no significant impact on the maximum deck acceleration, reducing the problem to a one-dimensional SOPSO. The results are presented primarily to provide the reader with an illustrative example of a single-objective approach, as it can be effectively visualized in 3D space.

5.5 Optimisation procedure case study

Figure 5.11 presents context to the optimisation of the case study, showing the Multi-Objective Particle Swarm Optimisation (MOPSO) algorithm and main optimisation loop. It provides context to the sequence of actions used to arrive at the next iteration in the optimisation process.

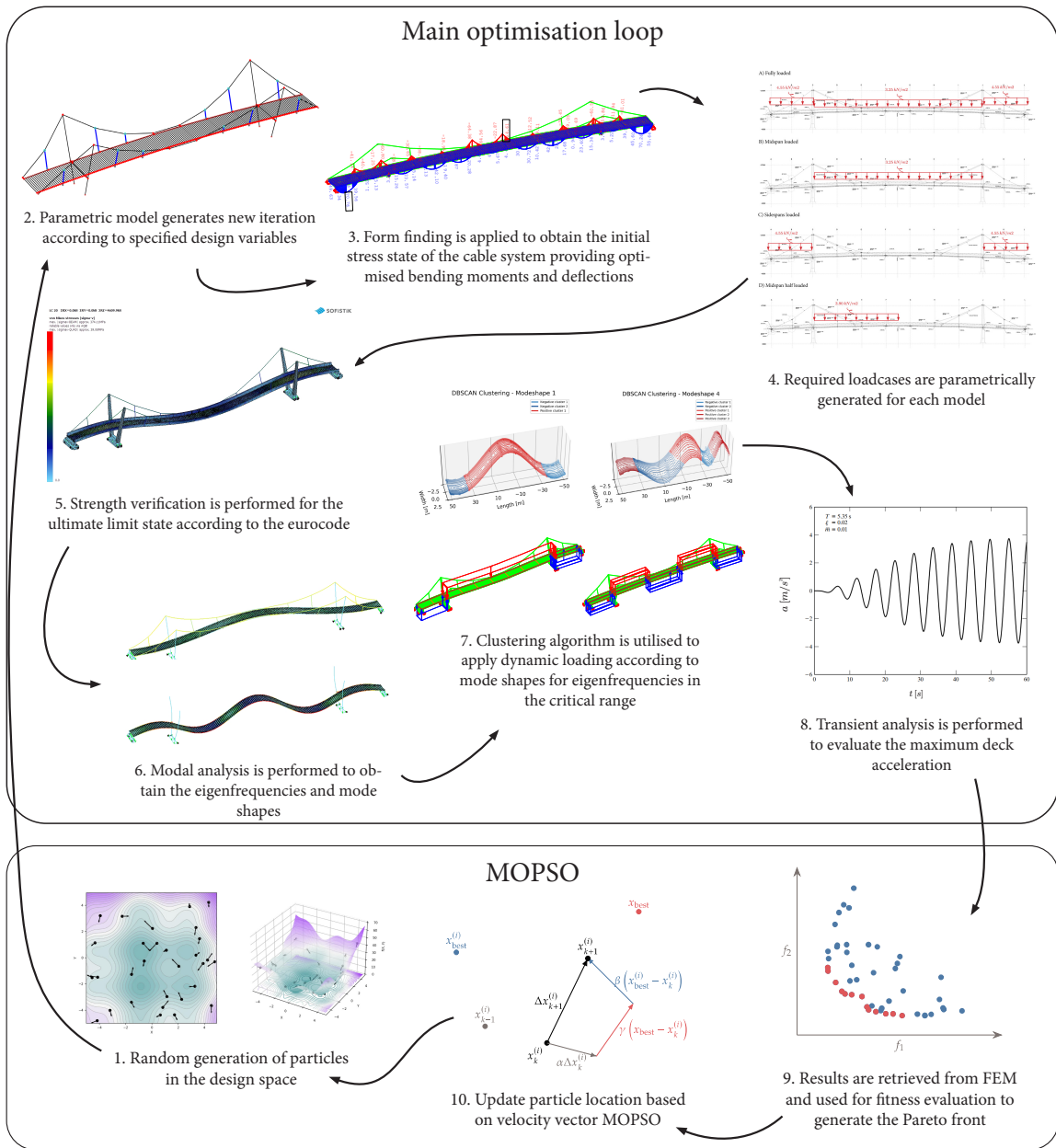


Figure 5.11: Optimisation procedure of the case study

1. **Random generation of particles in the design space**
The first step in the optimisation process generates particles that represent possible solutions in the design space, denoted by vector \bar{x} . The α -component determines similarity to the previous iteration (if applicable), whereas the β - and γ -components take random values between their specified bounds, stimulating local and global exploration respectively.
2. **Parametric model generates new iteration according to specified design variables**
A parametric model of the case study is generated based on the design variables defined in the particle generation in the previous step, starting the main optimisation loop.
3. **Form finding is applied to obtain the initial stress state of the cable system providing optimised bending moments and deflections**
Utilising the theory of self-anchored suspension bridges, an optimised initial stress state for the cable system is obtained and applied to the FE model accordingly.
4. **Required load cases are parametrically generated for each model**
Every particle with its specified design variables is subjected to the load cases required for the ultimate limit state verification. Appendix B - Load cases & combinations provides the load cases and combinations used for verification.
5. **Strength verification is performed for the ultimate limit state according to the Eurocode**
The defined load combinations are evaluated via FE analysis to verify the design. Linear analysis is performed and the maximum von Mises stress is verified for the main girders and cables.
6. **Modal analysis is performed to obtain the eigenfrequencies and mode shapes**
The FE model is subjected to modal analysis to determine the dynamic properties of the structure. The eigenfrequencies and mode shapes are obtained to simulate the crowd-loading of pedestrians. Moving load analysis is excluded, since the assumption is made that crowd loading leads to a higher acceleration response.
7. **Clustering algorithm is utilised to apply dynamic loading according to mode shapes for eigenfrequencies in the critical range**
DBSCAN clustering [53] is used for which the nodal displacements from the modal analysis in SOFiSTiK are analysed for each critical mode shape. The clustering algorithm detects areas with the same loading direction to apply loading in accordance with its mode shape.
8. **Transient analysis is performed to evaluate the maximum deck acceleration**
According to the Newmark-Beta method, direct numerical integration is applied with the factors $\gamma = 1/2$, $\beta = 1/4$ for all critical modes to evaluate maximum deck acceleration.
9. **Results are retrieved from FEM and used for fitness evaluation to generate the Pareto front**
The defined fitness function is evaluated for the mass and maximum acceleration objective. Based on Pareto dominance, the Pareto front is obtained, containing results in which solutions are dominant.
10. **Update particle location based on velocity vector MOPSO**
Based on the previous iteration, the particle's position is updated. The α component determines how similar the new velocity is to the previous iteration. The β component refers to the local best solution and the γ component to the global best solution. With this position update, the process can be repeated until the most optimal design is found.

6

Results

6.1 Effect design variables

The effect of each design parameter of the case study is presented in this section. The geometric modification of the design variables shows their influence on the dynamic characteristics of the structure and their effect on the maximum deck acceleration. It is important to note that the range of design variables presented is consistent with ULS verification. As the coupler plate length c_p had no direct effect on the torsional eigenfrequencies after a small sensitivity analysis, this design variable is excluded from the results.

6.1.1 Pylon height

Figure 6.1 illustrates the maximum deck acceleration corresponding to changes in pylon height, highlighting the mode shapes that dominate the dynamic analysis within the critical range and shows the eigenfrequency belonging to the design variable. The first vertical mode shape exceeds the comfort limit for design situation three, where $a_{max} < 1.0 \text{ m/s}^2$. Additionally, the first vertical mode shows a linear relationship between maximum acceleration and pylon height. This behaviour is explained by the linear reduction coefficient ψ , as explained in 2.5.4 Reduction coefficient ψ . To further clarify this trend, the eigenfrequencies for each mode shape are plotted for all pylon heights. The pylon height of the original design is indicated in the figure, revealing a response well above the set limit confirming the need for a tuned mass damper (TMD) design as anticipated.

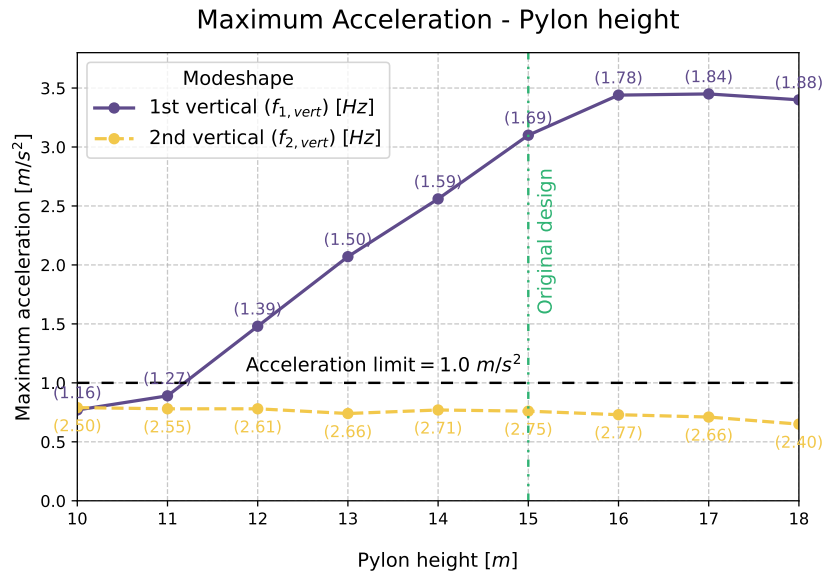


Figure 6.1: Maximum acceleration for the effect of varying pylon height plotted for dominant mode shapes within the critical range

6.1.2 Cross-section dimensions

The cross-sectional dimensions of the main girders are also considered as design variables in the optimization. Figure 6.2 illustrates the effect of modifying the girder dimensions on the maximum deck acceleration for the dominant critical mode shapes, where the height of the web and width of the flange are provided respectively. Observed can be that the second vertical mode shape falls below the acceleration limit as the girder dimensions increase. A linear relationship is evident for the second vertical mode shape, and the corresponding eigenfrequencies are plotted to emphasize this trend. The dimensions of the original design are again shown, confirming the need for a tuned mass damper (TMD) design as expected.

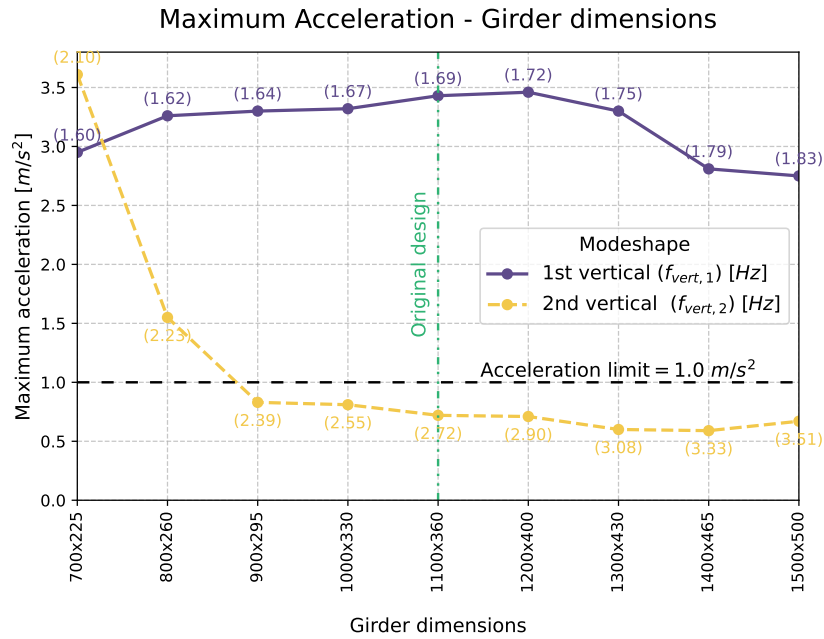


Figure 6.2: Maximum acceleration for the effect of varying cross-sections of the main girder plotted for dominant mode shapes within the critical range

6.1.3 Number of hangers

Figure 6.3 shows the effect of different hanger configurations on the maximum deck accelerations for all mode shapes within the critical range. The impact is minimal, leading to the conclusion that this design variable has no significant effect on the optimization and should be excluded. The original design of the case study, which incorporates six vertical hangers, is preferred in the optimized solution as it requires less connection detailing, reducing engineering effort and cost.

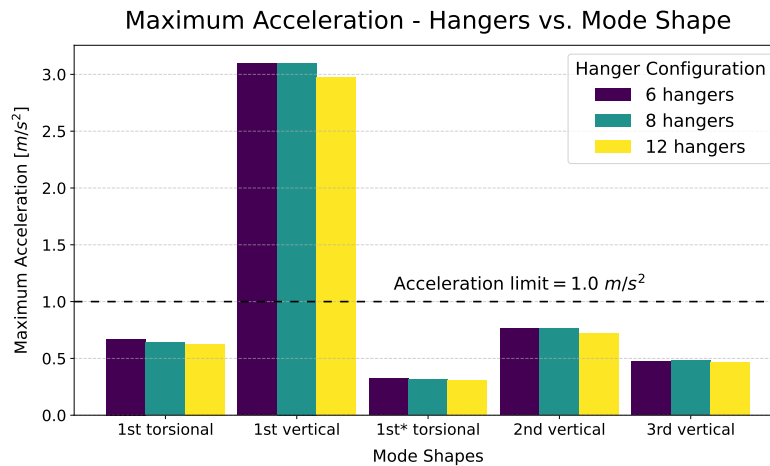
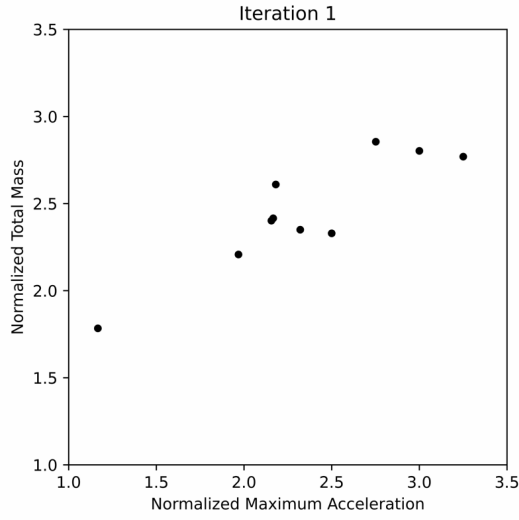


Figure 6.3: Influence hanger configuration regarding the maximum deck accelerations of the case study for each critical mode shape

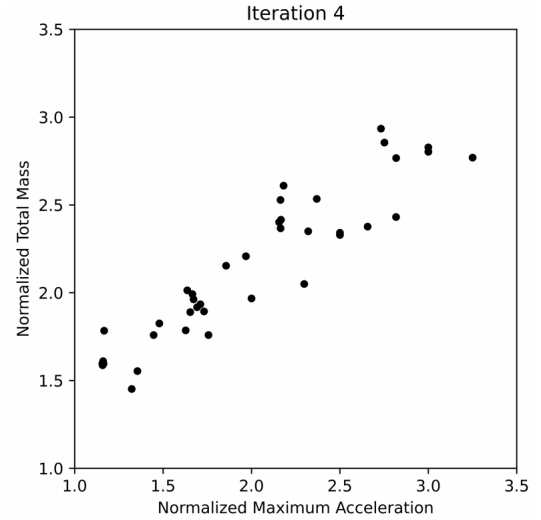
6.2 Multi-Objective Particle Swarm Optimisation (MOPSO)

Figures 6.4a to 6.4d illustrate the progression of the multi-objective particle swarm optimization (MOPSO) applied to the case study. The horizontal axis represents the normalized maximum acceleration, while the vertical axis corresponds to the normalized total mass, both based on the min-max normalization principle. Each black point indicates a potential solution derived from the defined design variables and their specified domains.

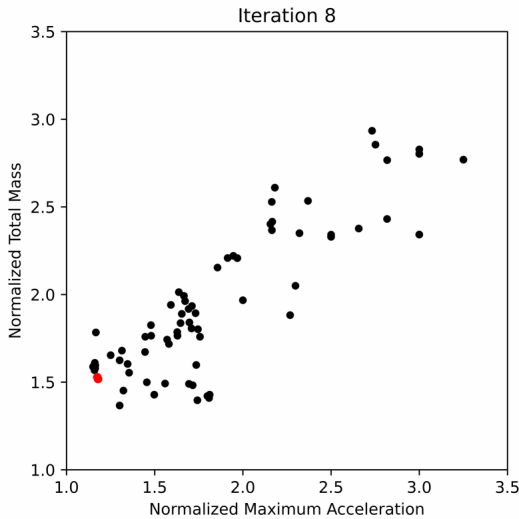
As the MOPSO algorithm advances, a transition from global to local exploration is evident, reflected in the shift from coarse to dense particle spacing across iterations. A total of ten particles were randomly initialized and iterated over twelve steps to approximate the Pareto front, with velocity components $\alpha = 0.9$, $\beta = 1.5$, $\gamma = 0.5$. This choice balances comprehensive exploration with manageable computational effort. After twelve iterations, the Pareto front was deemed satisfactory, leading to the termination of the MOPSO algorithm. It is important to note that there rest many approaches to generate a Pareto front. One could utilise fewer initial particles over reduced iterations, analyse the Pareto front and start a new MOPSO algorithm with refined bounds of the design variables. This approach will most likely result in reduced computational benefiting the user. Initialising more particles over increased iterations will explore a broader design space at the cost of significantly increased computational time.



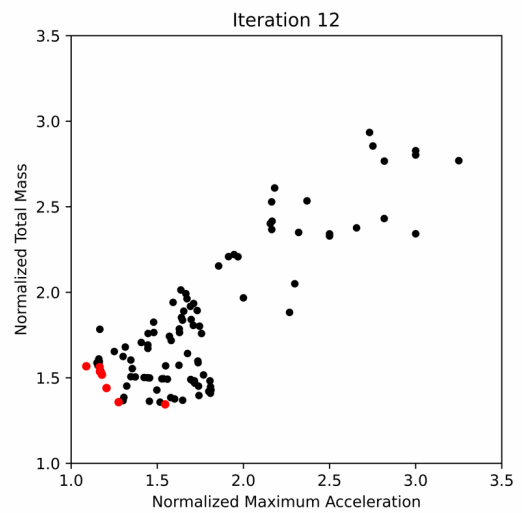
(a) First iteration for the MOPSO of the case study



(b) Fourth iteration for the MOPSO of the case study



(c) Eighth iteration for the MOPSO of the case study



(d) Twelfth iteration for the MOPSO of the case study

Figure 6.4: Comparison of different iterations for the MOPSO of the case study showing the progression of solutions. A black point represents a particle containing a design solution whereas a red point represents a Pareto dominant solution.

Each iteration shows a diagonally spaced particle field, meaning that the design variables chosen hold close to a linear relationship. The pylon height has the greatest influence, as has become evident from the results of the individual design variables, see section 5.2 - Design variables. Lower pylon height

results in lower eigenfrequencies hence lower dynamic crowd-loading due to the reduction coefficient ψ , resulting in decreased total mass and lower maximum accelerations. The girders only affect the second vertical mode shape for decreased dimensions, increasing maximum acceleration, whereas the number of hangers holds no influence.

Analyzing the effect of design variables individually before optimization can be highly beneficial. Variables with minimal impact on dynamic characteristics and maximum deck accelerations can be excluded from the optimization process, significantly reducing computational time. In the case study, the optimization procedure required just two days, compared to twenty days for evaluating the entire design space. This demonstrates the algorithm's effectiveness in generating the Pareto front. Nonetheless, further reductions in computational time are both feasible and desirable. This can be achieved by narrowing the design variables and conducting multiple MOPSO runs with domain restrictions.

6.3 Comparison of designs

Table 6.1 presents the Pareto front obtained by the MOPSO, for which four dominant solutions in the design space are depicted. The design vector \bar{x} represents a unique solution for all design variables and is constructed as shown in equation 6.1.

$$\bar{x} = \begin{bmatrix} h_w \\ w_f \\ h_p \\ n_{hang} \end{bmatrix} \quad (6.1)$$

The fitness function as defined in equation 5.9 is evaluated and returns a value for both the total mass $f_{mass}(\bar{x})$ and maximum acceleration $f_{a,max}(\bar{x})$, whilst subjected to penalties for unity checks $p_{uc}(\bar{x})$ and hanger configurations $p_{hang}(\bar{x})$. The results indicate that the stresses in the main cable dominate the case study. Since the ultimate limit state is incorporated into the MOPSO via penalties for unity checks of the main girders and cables, the algorithm identifies unfeasible design solutions. However, the lack of a design variable specifically addressing the ultimate limit state unity check for the main cable leads to exceedances.

h_w [mm]	w_f [mm]	h_p [m]	n_{hang}	a_{max} [m/s ²]	mode shape	f_n [Hz]	$f_{a,max}(\bar{x})$	$f_{mass}(\bar{x})$	uc_{girder}	uc_{cable}
1002.06	357.88	10.0	8	0.8921	1st vertical	1.1015	1.1784	1.5185	0.6947	1.0262
1028.82	367.44	10.0	8	0.8738	1st vertical	1.1152	1.1748	1.5280	0.6740	1.0173
1124.15	401.48	10.0	8	0.8272	1st vertical	1.1662	1.1654	1.5620	0.6078	0.9871
1052.94	376.05	10.0	8	0.8342	1st vertical	1.1275	1.1668	1.5366	0.6565	1.0096

Table 6.1: Pareto front obtained from MOPSO of the case study

It can be observed that one design solution remains within the stress limits of the main cables, making it the optimal solution. The maximum deck acceleration, governed by the first vertical mode shape, stays below the set limit for design situation three $a_{lim} < 1.0$ m/s². To facilitate effective comparison, the mass and acceleration objectives are normalized using min-max normalization. As previously mentioned, the height of the pylon h_p has the greatest influence on the dynamic properties of the case study. Other design variables show minimal to no deviation from the original design, as summarized in Table 6.2. A 12.9% reduction in total mass is achieved, while the maximum deck acceleration remains within the specified limit and satisfies the ultimate limit state requirements. The design effectively excludes the need of external damping by use of TMDs for the first vertical mode shape.

type	h_w [mm]	w_f [mm]	h_p [m]	n_{hang}	a_{max} [m/s ²]	mode shape	f_n [Hz]	total mass [t]	uc_{girder}	uc_{cable}
optimal	1124.15	401.48	10.0	8	0.8272	1st vertical	1.166	385	0.608	0.987
original	1100	390	16	8	3.43	1st vertical	1.610	442	0.559	0.684

Table 6.2: Comparison of optimal solution to original design case study

The mode shapes within the critical range of the optimal solution are shown in table 6.3. It becomes evident that the behaviour of the optimised solution is comparable to the original design as outlined in chapter 4 - Case study assessment. The critical modes show the same mode shapes at slightly different frequencies.

Original design			Optimised solution		
Mode nr.	Frequency	Mode shape	Mode nr.	Frequency	Mode shape
5	1.527	1 st vertical	1	1.166	1 st vertical
4	2.098	1 ^{st*} torsional	2	2.137	1 st torsional
7	2.661	2 nd vertical	3	2.549	2 nd vertical
8	2.887	1 st torsional	4	2.991	1 ^{st*} torsional

Table 6.3: Critical eigenfrequencies original design and optimised solution for design scenario 3

The transient responses obtained via direct time integration for all critical modes are shown in the figures below. Important to note, is that once again the torsional modes experience a beating-like behaviour. This is explained by the eigenfrequency of the pylons interfering with the pure torsional response.

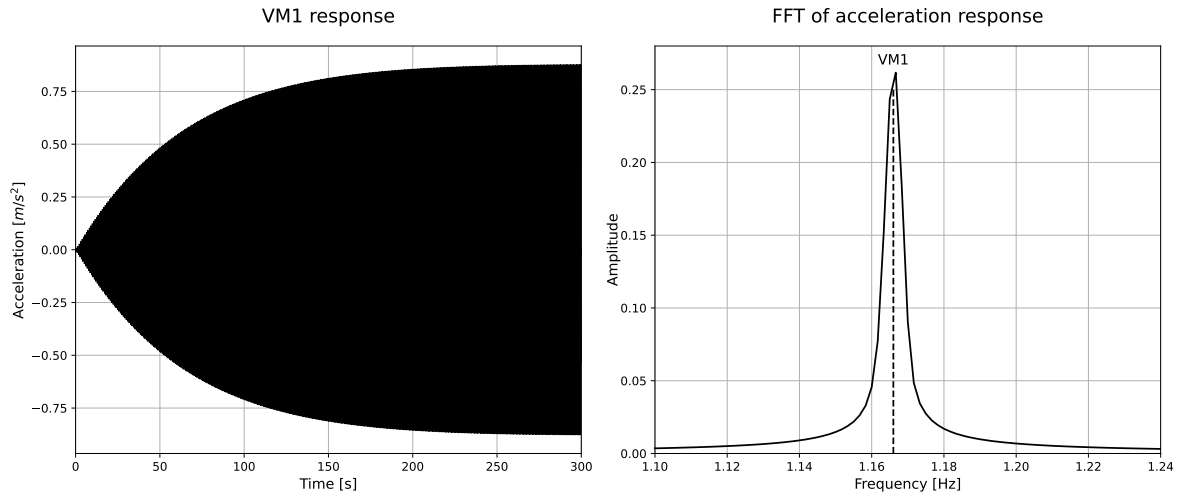


Figure 6.5: Transient response of the first vertical mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$

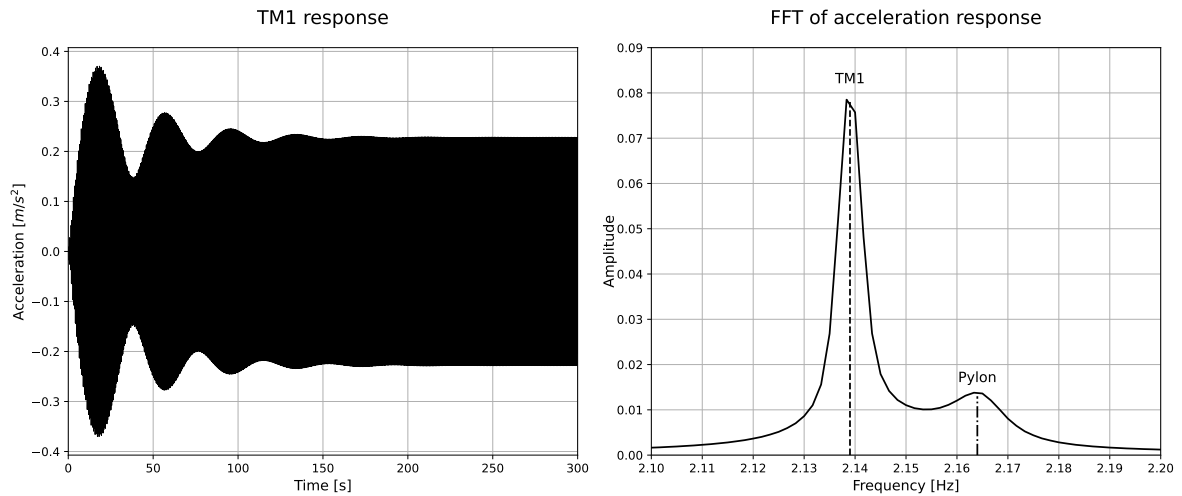


Figure 6.6: Transient response of the first torsional mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$

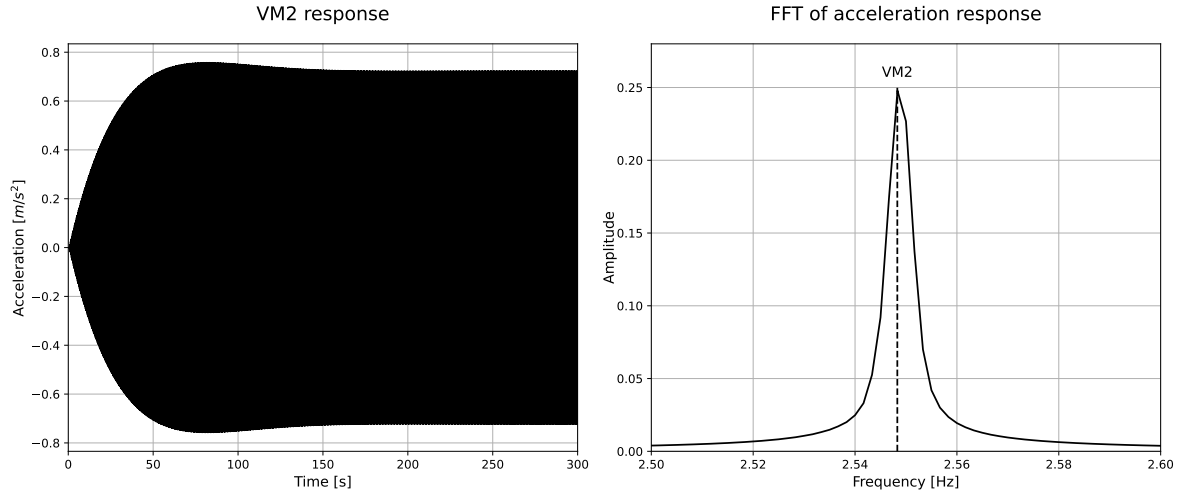


Figure 6.7: Transient response of the second vertical mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$

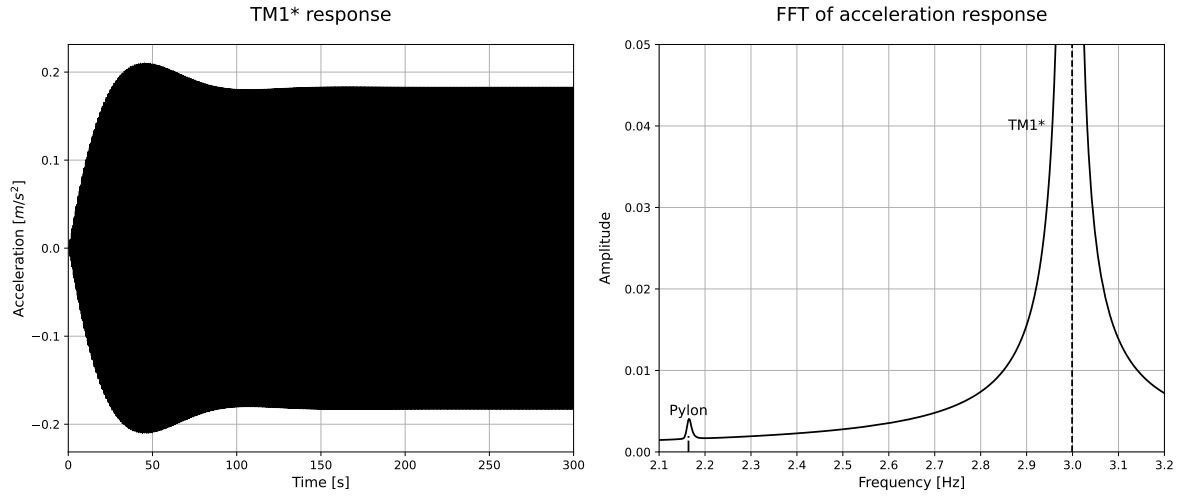


Figure 6.8: Transient response of the first torsional* mode shape for TC3, assuming $\xi_{s,min} = 0.2\%$

Observed is that all modes exhibit acceleration responses below the specified comfort limit of $a_{vert} = 1.0 \text{ m/s}^2$. The first and second vertical modes show comparable maximum accelerations, which can be attributed to the same load magnitude, as the reduction coefficient remains equal. While the first vertical mode technically falls outside the critical frequency range $f_{crit} = 1.25 \sim 4.60 \text{ Hz}$, the eigenfrequency obtained from FEM may differ from the actual frequency after the bridge is installed. Therefore, a reduced loading magnitude was maintained for the frequency ranges, as expressed in equation 6.2. This approach enhances the robustness of the optimization, accounting for potential deviations in eigenfrequency estimation.

$$\psi(f_n) = 0.25 \quad \text{if } 0.75 \leq f_n \leq 1.25 \text{ Hz} \quad \text{or} \quad 4.60 \leq f_n \leq 5.10 \text{ Hz} \quad (6.2)$$

where:

$\psi(f_n)$ = reduction coefficient determined by the critical frequency
 f_n = critical frequency obtained from modal analysis in FEM

7

Discussion

The research demonstrates that geometric optimization of pedestrian bridge design is a viable approach to mitigate human-induced vibrations and reduce dependence on external damping. However, the findings must be interpreted with caution, considering the study's limitations. This chapter examines the research outcomes, their significance, and the constraints that may influence its applicability. Additionally, it discusses key assumptions made to simplify the analysis or approximate specific structural behaviour. Recommendations for further research are outlined in chapter 8 - Conclusions and recommendations.

7.1 Interpretation

The research shows that geometric optimisation is a feasible solution for reducing human-induced vibrations and external damping reliance. This section explains the interpretation used to arrive at this outcome and the key findings which resulted from this.

Assessment method

The method employed to evaluate the maximum deck accelerations is direct time integration, implemented using the Newmark-Beta method. The parameters $\gamma = 1/2$ and $\beta = 1/4$ are selected for the numerical scheme, with a time step of $\Delta t = 30/f_{\max}$, where f_{\max} represents the maximum frequency within the critical range. This specific parameter selection eliminates amplitude and period decay, ensuring the stability and accuracy of the results. The Newmark-Beta method is preferred over alternative assessment methods due to its capability to account for modal coupling and (geometric) non-linear effects, which could be critical in dynamic systems with complex interactions. By incorporating these effects, the method provides reliable acceleration responses.

All eigenfrequencies within the identified critical range are subjected to direct time integration. The process conforms to the crowd-loading pattern, applied according to each mode shape. Consequently, performing this analysis over multiple iterations requires a systematic and robust workflow to identify the relevant eigenfrequencies and carry out the integration for all modes. While being thorough and precise, it is computationally intensive due to the significant time required for handling structural systems with various modes of vibration.

MOPSO

Multi-Objective Particle Swarm Optimization (MOPSO) has been applied to obtain the Pareto front, focusing on dominant solutions that minimize the total mass and maximum deck acceleration. The velocity components for the algorithm were set as $\alpha = 0.9$ (inertial), $\beta = 1.5$ (memory), and $\gamma = 0.5$ (social). To achieve these results, ten particles were generated and iterated over twelve cycles, producing a Pareto front representing the best possible design solutions within these parameters. It is worth discussing that the algorithm's performance could have been influenced by different tuning parameters or alternative utilisation. For instance, an initial run with fewer particles and a stronger emphasis on exploratory behaviour over fewer iterations could serve to define preliminary bounds. Subsequently, a second run of the MOPSO algorithm could be conducted with a larger particle initialization and more iterations, refining the search within these defined bounds for improved convergence.

Design variables

The design variables selected for the geometric optimisation of the case study include the pylon height, girder dimensions, number of hangers, and coupler plate length. Analysis of the results revealed that the coupler plate length and the number of hangers have negligible influence on the dynamic properties of the structure. Despite this, all design variables were included in the MOPSO process, leading to increased computational time. Conducting a sensitivity analysis of the individual design variables before the optimization process could provide valuable insight into their impact, allowing less influential variables to be excluded and significantly reducing computation time. Furthermore, the introduction of new design variables could lead to a better design, minimizing total mass and maximum deck acceleration.

Determination of eigenfrequencies

Modal analysis in finite element (FE) models was utilized to determine the eigenfrequencies of the structure, which plays a critical role in the magnitude of the applied loading. Accurate determination of eigenfrequencies is essential because they directly influence the dynamic response of the structure to human-induced vibrations. The transition periods defined by the reduction coefficient ψ are relatively small, resulting in significant changes in load magnitude over narrow frequency ranges. To validate the FE model experimentally, sledgehammer tests were conducted to obtain the mode shapes and eigenfrequencies of the structure. When comparing the eigenfrequencies derived from the FE model with those obtained from testing, discrepancies were no greater than 7.7%. This level of agreement indicates that the FE model provides a reasonably accurate representation of the dynamic properties. However, even small inaccuracies in the eigenfrequency determination can have significant implications. An underestimation of the eigenfrequencies could result in an incorrect prediction of the dynamic response, potentially leading to excessive vibrations and a reduction in pedestrian comfort. Caution should be made when assessing eigenfrequencies in FEM due to their potential shortcomings.

7.2 Implications

The findings of this study regard implications for pedestrian bridge design and optimization in the context of human-induced vibrations. The ability to mitigate excessive vibrations through geometric optimization without relying on external damping, suggests a more prominent role in vibration control strategies. This challenges conventional design approaches, where tuned mass dampers (TMDs) are often considered a necessity, despite their limitations due to eigenfrequency shifts under varying pedestrian loads.

From a methodological perspective, this research highlights the advantages of using direct time integration for accurately capturing geometric and material non-linearities and ensuring modal coupling effects are considered. While this method introduces higher computational costs, it provides a more reliable assessment of pedestrian-induced vibrations compared to simplified approaches.

Practically, these insights emphasize the importance of integrating optimization techniques in the preliminary design stages of pedestrian bridges. By considering crowd-loading effects within the optimization framework, designers can develop more resilient and cost-effective solutions that reduce reliance on external damping. This approach leads to faster construction timelines, lower maintenance costs, and improved long-term performance, making optimized bridge designs more feasible.

7.3 Limitations

This research is subject to several limitations, primarily stemming from the constrained time frame, the chosen optimization approach, and the necessary assumptions made to reduce computational demands. The restricted duration has imposed constraints on the depth of parametric studies, sensitivity analyses, and validation efforts, potentially limiting the generalizability of the findings.

Additionally, the optimization approach, while effective in balancing multiple objectives, may not explore the full design space due to trade-offs in computational efficiency. First of all, the design situations are limited to a single scenario, assumed to be representable and accurate. Secondly, does the clustering algorithm impose limitations on its accuracy for higher mode shapes. Thirdly, moving loads induced by pedestrians and joggers are excluded from the optimisation even though these seem to have profound effects. Fourthly, the assumption of a singular structural damping value leads to a conservative assessment of the dynamic response of the case study. Lastly, intentional excitation by a group of vandals deliberately trying to damage the structure should be considered, which is excluded from the optimisation.

Design situations

Design situation three is considered to evaluate comfort under human-induced vibrations, see table 7.1. It represents dense pedestrian traffic under medium comfort conditions. The selection of a single design situation aims to balance computational efficiency with representative loading and comfort limits.

Situation	Type	Class	Characteristics	Context
3	Traffic	TC3	0.5 persons / m ²	Still unrestricted walking, overtaking inhibited
	Comfort	CL2	0.50 m/s ² ~ 1.00 m/s ² ² 0.10 m/s ² ~ 0.30 m/s ² ³	Medium comfort

Table 7.1: Design situation three considered for assessment of the case study, see 3.8 - Design situations.

In the optimization, close to a linear response between the magnitude of crowd loading and maximum acceleration is observed. This justifies the reduction to one specific situation for which the most dominant is observed. However, if certain mode shapes show coupling effects or non-linear behaviour, all design situations should be assessed individually. This leads to a substantial increase in computational time.

Clustering algorithm

A DBSCAN [53] clustering algorithm is applied to determine the direction of crowd-loading based on the mode shape. Nodal displacement data is transferred from SOFiSTiK to Python, where clustering is performed. The resulting clusters are then returned to SOFiSTiK to apply the corresponding loads. However, it has been observed that the algorithm's precision decreases for higher mode shapes, particularly torsional ones, due to the lower magnitude of nodal displacements. This can lead to incorrect clustering, potentially producing non-viable results that might dominate certain design solutions. The case study shows that the dominant responses are correctly identified as the first and second vertical mode shapes. However, extending the optimisation process to different footbridge designs will ask for caution. To address cases where no clear mode shapes dominate the response, improved algorithm tuning or a more efficient clustering approach may be necessary. Figure 7.1 shows the algorithm clustering for mode shapes of a certain design solution. Observed can be that for the second torsional*, a derivation of the pure second torsional mode shape, three negative and two positive clusters are identified, even though they should both contain four. This could lead to incorrect load application and insufficient results from direct time integration.

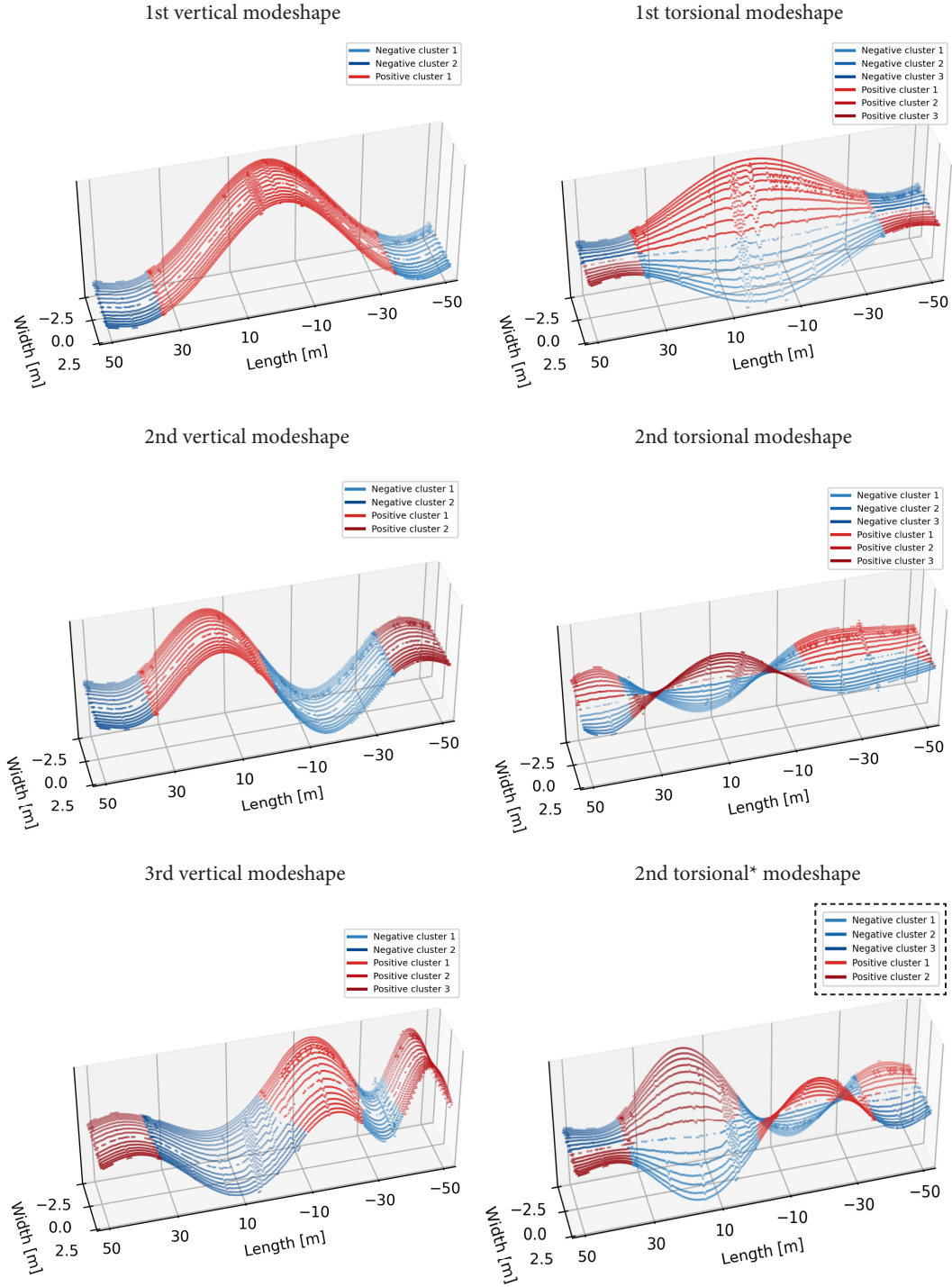


Figure 7.1: DBSCAN algorithm identifying clustering of the mode shapes, where the second torsional* mode shape shows incorrect clustering

Moving loads

The exclusion of moving loads within the optimization process is to be discussed. This decision is based on the assumption that crowd-induced loading will have a more profound impact on the structural response. However, this assumption introduces potential uncertainties, as moving loads such as individual pedestrians or joggers can contribute to localized dynamic effects that differ from the distributed nature of crowd loading. While pedestrian comfort has been assessed based on existing literature, the omission of moving load effects means that certain scenarios, particularly those involving resonant build-up due to periodic footfalls, may not be fully captured. In some cases, these effects could lead to acceleration responses that exceed those predicted by crowd-loading alone.

Structural damping

Minimum structural damping has been applied using Rayleigh damping for all critical modes. The damping value is selected to ensure a robust Pareto front, particularly in the absence of modal damping values from testing or similar structures. Notably, the conversion of unit load results from sledgehammer testing into values comparable to direct time integration highlights a strong correlation between the average structural damping and the first mode shape. Higher modes tend to be overestimated for both structural damping values. This observation suggests that incorporating the average damping value in the optimization process yields better results. Such an approach will enhance the prediction of acceleration responses while ensuring that optimized solutions remain reliable. This trade-off underscores the importance of carefully considering the damping assumptions when interpreting the results and refining the optimization methodology. The optimisation should enable engineers to account for different structural damping values or implementation of modal damping values from comparable structures.

Intentional excitation

In addition to crowd loading and moving loads, codes and guidelines state that deliberate dynamic loading caused by intentional, coordinated jumping by vandals should be considered. Figure 7.2 illustrates the stress response of the main girder of the original design when subjected to five vandals exerting a force with a dynamic load factor of 1.6 at midspan, as specified in prEN1991-2-2021 [23]. The analysis considers only the first vertical mode shape, as the resulting stress response remains well below the yield limit of $f_y = 355 \text{ N/mm}^2$. The assumption is made that higher modes will likewise remain below this stress level. Alternative methodologies are presented in the literature, namely NEN-EN1991-2-2019 [54] and the S etra guideline [4], which use different structural damping values and load models. The newly proposed Eurocode prEN1991-2-2021 assumes increased structural damping for ULS verification, which is confirmed by the literature. However, it states that large bridges require pedestrian load class TC4 or TC5. Since no classification of large bridges is given, the implementation of this load model is up for debate. For conventional bridges, two to five vandals are to be considered, for which the upper limit of five vandals was selected for this verification.

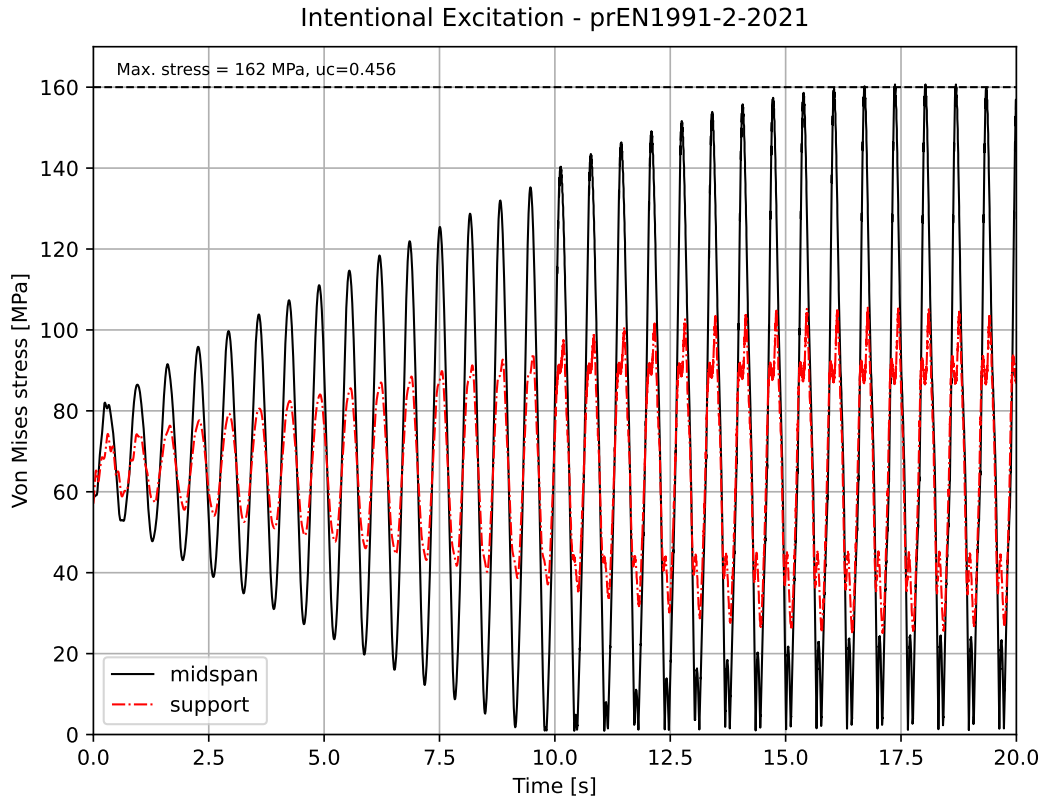


Figure 7.2: Stress response of the main girder at the support and midspan for intentional excitation according to prEN1991-2-2021 subjected to five vandals located at midspan with DLF=1.6

Conclusion and recommendations

8.1 Conclusion

This research seeks to optimize bridge geometry to reduce human-induced vibrations and lessen the dependence on external damping. The main research question is formulated as follows:

”What is the impact of parametric optimisation for footbridge geometry on reducing human-induced vibrations, and how does it affect the necessity of external damping systems?”

To effectively address the main research question, two sub-research questions have been formulated and addressed. Answering the sub-questions will build a logical progression of insights, leading to a conclusive response of the main research question.

Sub-question 1: ”What methods are most effective for assessing pedestrian-induced vibrations in footbridges?”.

- The single degree of freedom (SDOF) method can be utilised in the preliminary stages of design to gain an order of magnitude for the acceleration response. It shows a large scatter compared to the response obtained by direct time integration and measurements. FEM software needs to ensure the modal mass and load can be retrieved to evaluate the maximum response. In this research, only vertical modes of vibration could be obtained due to incorrect scaling of the modal load by the proportionality factor α_k since the response is a combination of torsional and lateral contributions. The scaling refers to mass matrix diagonalization, used by FE software to reduce the computational time for solving algebraic equations. Scaling the modal load by a proportionality factor α_k allows for the correct response evaluation for pure modes.
- Direct time integration proves to be a powerful approach for achieving high accuracy in human-induced vibration analysis. By accounting for geometric and material non-linearities, this method captures the full transient response of the structure, inherently incorporating modal coupling effects without requiring the superposition of modal responses. The Newmark-Beta method, with parameter selection $\gamma = 1/2$, $\beta = 1/4$, ensures numerical stability, preventing frequency and amplitude decay as the $\Delta t/T$ ratio increases. This approach provides valuable insight into the transient and steady-state response of the structure. A case study of a pedestrian bridge has been optimised according to the procedure outlined in this research. When compared to experimental validation through sledgehammer tests, similar accelerations are observed for the first vertical mode shape for an average structural damping value of $\xi_{s,avr} = 0.4\%$. However, higher mode shapes tend to be overestimated under the same damping assumption.
- Higher harmonics, signified by a multiple integer of the step frequency of pedestrians can impose contributions to the acceleration response. However, the literature mentions these effects are only observed up to the second harmonic, never leading to significant vibration of footbridges. This is confirmed by the Four Footfall Harmonics (4FFH) method, which considers the acceleration response induced by loading up to the fourth harmonic of pedestrians. Results show that only the first two harmonics show a notable response, for which major contributions stem from the first harmonic. This confirms that methods which include up to the second harmonic are most suited for assessment, ensuring accuracy and restricted computational time.

Sub-question 2: "How can geometric parameter studies be performed to optimise footbridge performance, tailored to reducing the need for external damping devices?"

- Employing a robust optimisation process with limited computational time, careful consideration of design variables and sound fitness evaluation tailored to the engineer's needs will provide a fruitful geometric optimisation of pedestrian bridges. In the context of the case study, it is observed that external damping through tuned mass dampers (TMDs) becomes unnecessary and the total mass of the structure is reduced by 12.9%. Furthermore, the form-finding procedure for self-anchored suspension bridges gives a reduction of 90 % of the bending moments in the initial stress state of the optimised solution compared to the original design.
- Reducing the number of design situations considered for all critical modes through direct time integration will significantly decrease computational time. A design situation is characterised by a traffic class denoting the dynamic loading in terms of pedestrian density and a comfort class specifying the allowable maximum acceleration. Analysing the transient responses of critical modes before the optimisation can offer valuable insights into the dynamic behaviour. As for the case study, a linear relationship between loading magnitude and acceleration response is established. This signifies the use of one design situation which represents dense traffic under medium comfort.
- Deploying a stochastic, meta-heuristic population-based genetic optimization algorithm stimulates a diverse exploration of the design space. It enables an efficient trade-off between multiple conflicting objectives, in this research signified by total mass and maximum deck accelerations. The optimisation progressively evolves optimal footbridge configurations that adhere to structural constraints and dynamic performance criteria. Different optimisation strategies can be utilised, such as coarse to fine exploration of the design space via velocity vector modification or multiple runs of the algorithm whilst being subjected to bound restrictions of the design variables.
- Design variable freedom can be seen as a double-edged sword in the optimization process. It allows for a comprehensive exploration of the design space, providing the flexibility to identify the most suitable solutions along the Pareto front, which represent the best trade-offs between competing objectives. This freedom facilitates the discovery of optimal configurations that might otherwise be overlooked if the design space were more restricted. However, the increased number of design variables introduces higher computational costs. The larger the design space, the more time and resources are required to evaluate potential solutions, leading to a trade-off between exploration and efficiency, where balancing these aspects is deemed crucial.
- An optimised solution among the Pareto front gives further implications for detailed design, such as the design of connections. The optimisation process allows for seamless integration of the optimised solution back to the FE software, enabling engineers to perform detailed analysis with refined elements. An example would be conversion from beam to shell elements and performing a geometric and material non-linear imperfection buckling analysis for localised buckling of connection details.

Main research question

To answer the main research question, the conclusions obtained in the sub-research questions are combined and extended. Parametric optimisation of footbridge geometry has profound effects on the dynamic characteristics, reducing human-induced vibrations and external damping reliance. Modification of design variables of the case study shows that maximum deck accelerations of the critical modes drop below the stated comfort limits whilst reducing the total mass of the structure. The modification of eigenfrequencies reduces crowd-loading since step- and eigenfrequency no longer coincide. External damping by tuned mass damper design is excluded, leading to faster construction timelines, lower maintenance costs, and improved long-term performance.

In practical terms, the optimisation process emphasizes the importance of integrating optimization techniques in the early design stages of pedestrian bridges. Designers can develop more resilient and cost-effective solutions that reduce reliance on external damping. In the context of the case study: form-finding the optimal stress in the main cable system; performing strength verification of the girders and cables; utilising a clustering algorithm for dynamic crowd-load appliance; assessing the dynamic response by direct time integration and applying an optimisation algorithm for multiple iterations has shown optimised results. By applying Multi-Objective Particle Swarm Optimisation (MOPSO), the computational time has been reduced from twenty days to calculate the entire design space, to two days providing a satisfactory Pareto front.

Design variable freedom will harness the full potential of the optimisation, since the ability to explore a wide range of parameters and designs is wishful. However, it comes with implications for the computational time and the means of verifying strength and the dynamic response. The optimisation process requires a robust integration scheme which performs a detailed analysis of the dynamic response.

8.2 Recommendations

The research is paired with recommendations for its intended use and its belonging limitations. This section aims to provide recommendations for how geometric optimisation of pedestrian bridges can be utilised in other designs whilst cohering to the verification outlined in the literature.

Moving load analysis

Incorporating the analysis of moving loads related to pedestrian and jogger comfort into the optimization process is strongly recommended. Results indicate that the loading magnitude of individual pedestrians and joggers is comparable to that of an equivalent number of perfectly synchronized pedestrians in a crowd-loading scenario. The resulting acceleration responses from moving loads closely match those induced by crowds, suggesting that moving load analysis could be critical for certain design configurations. Moving loads are primarily characterized by the time required to cross the footbridge, which depends on the users' walking velocity. A longer crossing time increases the potential for resonance build-up. However, critical velocity considerations are unnecessary, as they are typically relevant only for high-speed trains or similar loads.

Early stages of design

The optimisation process utilised in this research will prove to be most fruitful in early stages of the design. Ensuring greater variable freedom leads to a greater optimised Pareto front generation, thus a better-optimised solution. The writer identifies that incorporation in early stages is not easily implemented, requiring a robust optimisation process with seamless integration of all components to arrive at a stable iteration scheme. Furthermore, a detailed analysis performed to reduce human-induced vibrations is not always wishful in engineering practice.

Recommended is the generalisation of geometric parameter-driven design, for which new design solutions are easily generated. Utilising eigenfrequency identification and clustering algorithms to correctly apply crowd-loading responses needed for comfort evaluation. One could think of the development of an optimisation loop in which standardised engineering practice for geometric parameter-driven design by use of Rhino Grasshopper [55] is incorporated in further aspects of the optimisation process.

Design variable selection

Prioritizing variables that have a strong correlation with dynamic performance can enhance the convergence rate of the optimization process. Key parameters such as stiffness distribution, mass distribution, and damping characteristics should be carefully evaluated, as they play a fundamental role in shaping the structure's vibrational response. A comprehensive sensitivity analysis of these variables is recommended before initiating the optimization to ensure their direct impact on the structural dynamics. This preliminary step helps refine the selection of variables, eliminating those with negligible effects, thereby significantly reducing computational effort and improving the efficiency of the optimization algorithm. Additionally, constraints related to feasibility, constructibility, and material limitations should be integrated into the selection process to balance performance improvements with practical implementation.

Bibliography

1. Baus, U. & Schlaich, M. *Footbridges* ISBN: 978-3-7643-8222-3 (Birkhäuser, July 2007).
2. Tucci, P. L. The Pons Sublicius: A Reinvestigation. *Memoirs of the American Academy in Rome* **56/57**. Publisher: [American Academy in Rome, University of Michigan Press], 177–212. ISSN: 0065-6801. <https://www.jstor.org/stable/24616441> (2024) (2011).
3. Kumar, P. & Kumar, A. *human induced vibration in structures* Oct. 2013.
4. Maury, C. P. B. Sétra Footbridges Assessment of vibrational behaviour of footbridges under pedestrian loading.
5. European Commission. Joint Research Centre. & European Convention for Constructional Steelworks (ECCS). *EUR23984 / JRC-document - Design of lightweight footbridges for human induced vibrations: background document in support to the implementation, harmonization and further development of the Eurocodes*. eng (2009).
6. *Advanced load models for synchronous pedestrian excitation and optimised design guidelines for steel footbridges (SYNPEX)* eng (ed Butz, C.) ISBN: 978-92-79-08303-7 (Off. for Official Publ. of the European Communities, Luxembourg, 2008).
7. *BS 5400 - Steel, concrete and composite bridges - Part 2. Specification for loads* British Standard (1978).
8. *Steel Design Guide 11 - Vibrations of Steel-Framed Structural Systems Due to Human Activity - Second Edition* AISC (2016).
9. *NEN-EN 1990+A1+A1/C2 nl - Eurocode: Basis of structural design* NEN (2019).
10. *prEN 1990 - Eurocode: Basis of structural and geotechnical design* NEN (2021).
11. Willford, M. R. & Young, P. *A design guide for footfall induced vibration of structures: [a tool for designers to engineer the footfall vibration characteristics of buildings or bridges]* eng. OCLC: 305136373. 2006.
12. *ISO 10137 - Bases for design of structures - Serviceability of buildings and walkways against vibrations* ISO (2012).
13. *ISO 2631 - Mechanical vibration and shock - Evaluation of human exposure to whole-body vibration* ISO (1997).
14. Živanović, S., Pavic, A. & Reynolds, P. Vibration serviceability of footbridges under human-induced excitation: a literature review. *Journal of Sound and Vibration* **279**, 1–74. ISSN: 0022-460X. <https://www.sciencedirect.com/science/article/pii/S0022460X04001646> (2005).
15. Matsumoto, Y., Shiojiri, H. & Nishioka, T. *Dynamic design of footbridges* Aug. 1978. (2023).
16. Synold, G. *Einfluß von Brückenschwingungen auf das Wohlbefinden von Menschen, Diplomarbeit am Institut für Tragwerksentwurf und -konstruktion* Universität Stuttgart (1995).
17. Živanović, S., Pavic, A. & Reynolds, P. Vibration serviceability of footbridges under human-induced excitation: a literature review. *Journal of Sound and Vibration* **279**, 1–74. ISSN: 0022-460X. <https://www.sciencedirect.com/science/article/pii/S0022460X04001646> (2005).
18. Ellis, B. Serviceability evaluation of floor vibration induced by walking loads, 30–36 (Nov. 2001).
19. Dallard, P. The London Millennium Footbridge. en. **79** (2001).
20. Caetano, E., Cunha, A. & Moutinho, C. Implementation of Passive Devices for Vibration Control at Coimbra Footbridge. eng. Accepted: 2022-09-11T04:03:15Z. <https://repositorio-aberto.up.pt/handle/10216/102322> (2024) (2007).
21. Oeding, D. Verkehrsbelastung und Dimensionierung von Gehwegen und anderen Anlagen des Fußgängerverkehrs. *Dissertation* (1963).
22. Feldmann, C. H. *bibinitperiod M. in Footbridge Vibration Design* Num Pages: 18 (CRC Press, 2009). ISBN: 978-0-429-18293-8.
23. *prEN 1991-2 - Eurocode 1: Actions on structures - Part 2: Traffic loads on bridges* NEN (2021).
24. Lanczos, C. An Iteration Method for the Solution of the Eigenvalue I Problem. of Linear Differential and Integral Operators. *Journal of Reasearch of the National Bureau of Standards Publication* **45, No. 4** (Oct. 1950).

25. Duan, L. & Chen, W.-F. *Bridge Engineering Handbook* Second Edition (CRC Press, 2014).
26. Bachmann, H. & Ammann, W. *Vibrations in Structures: Induced by Man and Machines* en. Google-Books-ID: T7PhFZVfMC4C (IABSE, 1987).
27. *Mechanical Vibrations: Theory and Application to Structural Dynamics, 3rd Edition* / Wiley en-ae. <https://www.wiley.com/en-ae/Mechanical+Vibrations%3A+Theory+and+Application+to+Structural+Dynamics%2C+3rd+Edition-p-9781118900208> (2023).
28. Chopra, A. K. *Dynamics of Structures* Fourth Edition (Prentice Hall, 2011).
29. J.M.J., S., A.W.C.M., V. & E.C., K. *Structural Dynamics CT4140 Part 1 - Structural Vibrations* Jan. 2005.
30. Wheeler, J. Induced vibrations in footbridges. *Proceedings of the 10th Australian Road Research Board (ARRB) Conference* **10** (3) (1980).
31. Andriacchi, T. Walking speed as a basis for normal and abnormal gait measurements. *Journal of Biomechanics* **10** (1977).
32. *NEN-EN 1990+A1+A1/C2/NB nl - Nationale Annex for NEN-EN 1990+A1:2006+A1:2006/C2:2019 Eurocode: Basis of structural design* NEN (2019).
33. Mary, A. FEA Finite Element Procedures by K-J-Bathe. en. https://www.academia.edu/8326777/FEA_Finite_Element_Procedures_by_K_J_Bathe (2024).
34. Bathe, K. J. & Wilson, E. L. Stability and accuracy analysis of direct integration methods. en. *Earthquake Engineering & Structural Dynamics* **1**, 283–291. ISSN: 0098-8847, 1096-9845. <https://onlinelibrary.wiley.com/doi/10.1002/eqe.4290010308> (2024) (Jan. 1972).
35. Chatzi, D. E., Abbiati, D. G. & Agathos, D. K. The Finite Element Method for the Analysis of Non-Linear and Dynamic Systems: Non-Linear Dynamics Part I. en.
36. Zoltowski, K., Banas, A., Binczyk, M. & Kalitowski, P. Control of the bridge span vibration with high coefficient passive damper. Theoretical consideration and application. *Engineering Structures* **254**, 113781. ISSN: 0141-0296. <https://www.sciencedirect.com/science/article/pii/S0141029621018551> (2024) (Mar. 2022).
37. Alhasan, A., Vafaei, M., Ali, S. & Jian, T. Viscoelastic damper for vibration mitigation of footbridges. *IOP Conference Series: Earth and Environmental Science* **1205**, 012051 (June 2023).
38. Garcia-Troncoso, N., Ruiz-Teran, A. & Stafford, P. J. Attenuation of pedestrian-induced vibrations in girder footbridges using tuned-mass dampers. *Advances in Bridge Engineering* **1**, 14. ISSN: 2662-5407. <https://doi.org/10.1186/s43251-020-00013-8> (2024) (Nov. 2020).
39. Frahm, H. US989958A. <https://patents.google.com/patent/US989958A/en> (2024)(1911).
40. Hartog, J. P. D. *Mechanical Vibrations* en. Google-Books-ID: mtDCAgAAQBAJ. ISBN: 978-0-486-13185-6 (Courier Corporation, Feb. 2013).
41. Connor, J. *Introduction to Structural Motion Control* in (Aug. 2002). <https://www.semanticscholar.org/paper/Introduction-to-Structural-Motion-Control-Connor/d0137c09f23ba626bbca3142236411b17adb3ca0> (2024).
42. Ghosh, A. & Basu, B. A closed-form optimal tuning criterion for TMD in damped structures. en. *Structural Control and Health Monitoring* **14**. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.176>, 681–692. ISSN: 1545-2263. <https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.176> (2024) (2007).
43. Int., S. W. *Fietsbrug Voldijk Tilburg - Fabrication Drawings, Production Stage* Dec. 2012.
44. Jung, M.-R., Shin, S.-U., Attard, M. M. & Kim, M.-Y. Deflection Theory for Self-Anchored Suspension Bridges under Live Load. EN. *Journal of Bridge Engineering* **20**. Publisher: American Society of Civil Engineers, 04014093. ISSN: 1943-5592. <https://ascelibrary.org/doi/10.1061/%28ASCE%29BE.1943-5592.0000687> (2024) (July 2015).
45. Kim, H.-K., Lee, M.-J. & Chang, S.-P. Non-linear shape-finding analysis of a self-anchored suspension bridge. *Engineering Structures* **24**, 1547–1559. ISSN: 0141-0296. <https://www.sciencedirect.com/science/article/pii/S0141029602000974> (2024) (Dec. 2002).
46. Kim, K.-S. & Lee, H. S. Analysis of target configurations under dead loads for cable-supported bridges. *Computers & Structures* **79**, 2681–2692. ISSN: 0045-7949. <https://www.sciencedirect.com/science/article/pii/S0045794901001201> (2024) (Nov. 2001).
47. Jung, M.-R., Min, D.-J. & Kim, M.-Y. Nonlinear analysis methods based on the unstrained element length for determining initial shaping of suspension bridges under dead loads. *Computers & Structures* **128**, 272–285. ISSN: 0045-7949. <https://www.sciencedirect.com/science/article/pii/S0045794913002083> (2024) (Nov. 2013).

48. SCIA Engineer nl. <https://www.scia.net/nl/scia-engineer> (2025).
49. Mediaworks. *GSA: Structural Analysis and Design Software* en. <https://www.oasys-software.com/products/gsa/> (2025).
50. *SOFiSTiK FEM, BIM and CAD Software for Structural Engineers* <https://www.sofistik.com/en/> (2024).
51. Doebling, S. W., Farrar, C. R. & Cornwell, P. J. *DIAMOND: A graphical interface toolbox for comparative modal analysis and damage identification* English. Tech. rep. LA-UR-97-38; CONF-970736-2 (Los Alamos National Lab. (LANL), Los Alamos, NM (United States), June 1997). <https://www.osti.gov/biblio/491555> (2024).
52. R.R.A. Martins, J. & Ning, A. *Engineering Design Optimisation* ISBN: 978-1-108-83341-7 (Cambridge University Press, 2021).
53. Deng, D. *DBSCAN clustering algorithm based on density* in *2020 7th international forum on electrical engineering and automation (IFEEA)* (2020), 949–953.
54. *NEN-EN 1991-2+C1/NB nl - National Annex for NEN-EN 1991-2+C1: Eurocode 1: Actions on structures - Part 2: Traffic loads on bridges* NEN (2019).
55. Robert McNeel & Associates. *Rhinoceros and grasshopper* manual (2024). <https://www.rhino3d.com/>.
56. *NEN-EN 1991-2+C1 nl - Eurocode 1: Actions on structures - Part 2: Traffic loads on bridges* NEN (2015).
57. *NEN-EN 1991-1-4+A1+C2/NB nl - National Annex for NEN-EN 1991-1-4+A1+C2: Eurocode 1: Actions on structures- Part 1-4: General actions - Windload* NEN (2011).

A

Appendix A - Python source code

Within the optimisation process, four Python scripts have been utilised to apply the MOPSO algorithm and generate the Pareto front. These scripts and their respective part in the optimisation process, see figure 5.11 in Chapter 5.5 - Optimisation procedure case study, are explained below.

- **Parametric script of the case study - Step 2**
Parametric script of the case study according to the specified design variables, including modal analysis and load case generation.
- **ULS verification - Step 5**
Parametric script for the ULS verification of the main girders and cable system.
- **Mode shape clustering - Step 7**
The parametric script for the mode shape clustering according to the DBSCAN algorithm for crowd-induced loading.
- **MOPSO algorithm - Steps 1, 9 & 10**
Parametric script for the multi-objective particle swarm optimisation of the case study and the generated Pareto front with its export to a csv-file.

A.1 Parametric script of the case study - Step 2

The parametric script of the case study according to the specified design variables, including modal analysis and load case generation is presented in the Python script below.

```
1  """Python script of the parametric case study"""
2
3  import numpy as np
4  from os.path import isfile
5  import matplotlib.pyplot as plt
6  from mpl_toolkits.mplot3d import Axes3D
7  from mpl_toolkits.mplot3d.art3d import Poly3DCollection
8  from matplotlib import cm
9  from matplotlib.gridspec import GridSpec
10 from PIL import Image, ImageOps
11
12 def par_model(hang_ss, hang_ms, hang_ms_p, con_ms, hp, fz_p, c_d, tw, tf, wl, wu, hw, wf,
13              filename):
14     # Variables
15     if ((hp + 0.85) - fz_p) < 6.78:
16         fz_p = (hp + 0.85) - 6.78
17
18     rein_r = 0.6 # reinforcement ratio
19     A_rl = ((rein_r / 100) * wl ** 2 / 100)
20     A_ru = (2 * rein_r / 100) * wu ** 2 / 100
21
22     w = 5.10 # width deck [m]
23
24     del_hw = (hw - 1100) / 2
25     del_wf = (wf - 390) / 2
26
27     hc = 200 # height coupler plate
28     bc = 20 # thickness coupler plate
29
30     alph = np.deg2rad(13.8) # angle pylons [degrees] 13.8
31
32     cables = 1
```

```

33 cab_type = ["CZ", "CE"]
34
35 # Cable variables
36 alph_cs = np.deg2rad(15)
37 girders_cs = 2 * (2 * (hw / np.cos(alph_cs)) * tw + 2 * (wf * tf))
38 red_cs = 2 * (2 * (1100 / np.cos(alph_cs)) * 10 + 2 * (390 * 20)) - girders_cs
39 red_sw = red_cs * 1e-6 * 78.5
40
41 wd = 13.65 - red_sw #13.65 self weight to be resisted [kN/m]
42
43 # Curvature variables
44 v0 = 5.51 # at ends bridge [m]
45 v1 = 6.78 # at midspan bridge [m]
46 l = 103.2 # total length bridge [m]
47
48 # Locations pylons (pos. foundation)
49 zp = hp + 0.85
50 yp = hp * np.tan(alph) + 2.07
51
52
53 # plotting the bridge
54 plt_model = []
55 plt_model_deck = []
56 plt_model_deck_test = []
57
58 plt_g_ss = []
59 plt_g_ms = []
60 plt_c_ss = []
61 plt_c_ms = []
62 plt_tr_ss = []
63 plt_tr_ms = []
64 plt_pyl = []
65 plt_p_ss = []
66 plt_p_ms = []
67 plt_h_ss = []
68 plt_h_ms = []
69
70 # loading deck
71 sln = []
72 sar_ss = []
73 sar_ms = []
74 cab = []
75
76 # wind loads
77 sln_w_xx = []
78 sln_w_xx_p = []
79 sln_w_yy = []
80 sln_w_zz = []
81 sln_w_p = []
82
83 # add moments pylons
84 pnt_pyl = []
85
86
87 #region A) Girder elements side span
88 div_ss = hang_ss + 1
89
90 x1_g_ss = -51.5100000
91 x2_g_ss = -34.5100000
92 l_ss = abs(x1_g_ss - x2_g_ss)
93 y_g_ss = - (w / 2)
94 z1_g_ss = 5.51000000
95 z2_g_ss = 6.19000000
96 elnr_g_ss = 700
97
98 xh_ss_values = [i * (abs(x1_g_ss) - abs(x2_g_ss)) / div_ss + x1_g_ss for i in range(1,
99 div_ss)]
100
101 zg_ss_values = [
102     (4 * (v0 - v1) / (l ** 2)) * (i * (abs(x1_g_ss) - abs(x2_g_ss)) / (div_ss)) ** 2 +
103     (-4 * (v0 - v1) / l) * (i * (abs(x1_g_ss) - abs(x2_g_ss)) / (div_ss)) + v0
104     for i in range(1, div_ss)
105 ]
106
107 def print_lines_g_ss(start_num, sign_x, sign_y, sno):
108     output = ""
109     for i in range(div_ss):
110         if i < 1:

```

```

112         elnr = start_num + i
113         output += f"\nSLN{start_num+1:05d}GRP{5:05d}STYP{1:05d}B{1:05d}SNO{1:05d}TITL{1:05d}Line\"
114
115         x1 = sign_x * xh_ss_values[i - 1] if i > 0 else sign_x * x1_g_ss
116         y1 = sign_y * abs(y_g_ss)
117         z1 = abs(zg_ss_values[i - 1]) if i > 0 else abs(z1_g_ss)
118
119         x2 = sign_x * xh_ss_values[i] if i < hang_ss else sign_x * x2_g_ss
120         y2 = sign_y * abs(y_g_ss)
121         z2 = abs(zg_ss_values[i]) if i < hang_ss else abs(z2_g_ss)
122
123         output += f"\nSLNB{1:05d}X1{1:05d}{round(x1,3)}{1:05d}{round(y1,3)}{1:05d}{round(z1,3)}{1:05d}X2{1:05d}{
124             round(x2,3)}{1:05d}{round(y2,3)}{1:05d}{round(z2,3)}\"
125         plt_model.append([(x1,y1,z1),(x2,y2,z2)])
126         sln.append(elnr)
127
128         sln_w_yy.append(elnr)
129         sln_w_zz.append(elnr)
130
131         if sign_y == -1:
132             sln_w_xx.append(elnr)
133
134     else:
135         elnr = start_num + i
136         output += f"\nSLN{start_num+1:05d}GRP{4:05d}STYP{1:05d}B{1:05d}SNO{1:05d}TITL{1:05d}Line\"
137
138         x1 = sign_x * xh_ss_values[i - 1] if i > 0 else sign_x * x1_g_ss
139         y1 = sign_y * abs(y_g_ss)
140         z1 = abs(zg_ss_values[i - 1]) if i > 0 else abs(z1_g_ss)
141
142         x2 = sign_x * xh_ss_values[i] if i < hang_ss else sign_x * x2_g_ss
143         y2 = sign_y * abs(y_g_ss)
144         z2 = abs(zg_ss_values[i]) if i < hang_ss else abs(z2_g_ss)
145
146         output += f"\nSLNB{1:05d}X1{1:05d}{round(x1,3)}{1:05d}{round(y1,3)}{1:05d}{round(z1,3)}{1:05d}X2{1:05d}{
147             round(x2,3)}{1:05d}{round(y2,3)}{1:05d}{round(z2,3)}\"
148         plt_model.append([(x1,y1,z1),(x2,y2,z2)])
149         sln.append(elnr)
150
151         sln_w_yy.append(elnr)
152         sln_w_zz.append(elnr)
153
154         if sign_y == -1:
155             sln_w_xx.append(elnr)
156
157     return output
158
159 result_g_ss = (
160     print_lines_g_ss(elnr_g_ss, -1, -1, 2) + # Negative xh_ms, negative y_g_ms
161     print_lines_g_ss(elnr_g_ss + div_ss, -1, 1, 1) + # Negative xh_ms, positive y_g_ms
162     print_lines_g_ss(elnr_g_ss + 2 * div_ss, 1, -1, 1) + # Positive xh_ms, negative
163     y_g_ms
164     print_lines_g_ss(elnr_g_ss + 3 * div_ss, 1, 1, 2) # Positive xh_ms, positive y_g_ms
165 )
166
167 #endregion
168
169 #region B) Girder elements main span
170 div_ms = hang_ms + 1
171 div_ms_p = hang_ms_p + 1
172 l_ms = 1 - 2 * l_ss # length main span [m]
173 l_ms_1 = 29.571
174
175 x1_g_ms = -34.51
176 x2_g_ms = -4.93
177 x3_g_ms = 34.51
178 y_g_ms = - (w / 2)
179 z1_g_ms = 6.19
180 z2_g_ms = 6.78
181 elnr_g_ms = 800
182
183 if con_ms == 1:
184     xh_ms_values = [i * (abs(x1_g_ms) - abs(x2_g_ms)) / div_ms + x1_g_ms for i in range
185                     (1, div_ms)]
186
187     zg_ms_values = [
188         (4 * (v0 - v1) / (1 ** 2)) * (i * (abs(x1_g_ms) - abs(x2_g_ms)) / (div_ms) + l_ss
189         ) ** 2 +
190         (-4 * (v0 - v1) / 1) * (i * (abs(x1_g_ms) - abs(x2_g_ms)) / (div_ms) + l_ss) + v0

```

```

187         for i in range(1, div_ms)
188     ]
189     div = div_ms
190     ind = 1
191 elif con_ms == 0:
192     delt_x_ms = l_ms / (div_ms_p)
193     xh_ms_values = [(i * delt_x_ms) - (l_ms / 2) for i in range(1, div_ms_p)]
194     zg_ms_values = [
195         (4 * (v0 - v1) / (1 ** 2)) * ((i * delt_x_ms) + l_ss) ** 2 +
196         (-4 * (v0 - v1) / 1) * ((i * delt_x_ms) + l_ss) + v0
197         for i in range(1, div_ms_p)
198     ]
199     div = div_ms_p
200     ind = 0
201
202 def print_lines_g_ms(start_num, sign_x, sign_y, sno, div, ind):
203     output = "" # Initialize an empty string to store the appended lines
204     for i in range(div):
205         elnr = start_num + i
206         output += f"\nSLN{u*10}{elnr}_GRP4_STYP'B'_SNO_{sno}_TITL_\"Line\""
207
208         if ind == 1:
209             x1 = sign_x * xh_ms_values[i - 1] if i > 0 else sign_x * x1_g_ms
210             y1 = sign_y * abs(y_g_ms)
211             z1 = abs(zg_ms_values[i - 1]) if i > 0 else abs(z1_g_ms)
212
213             x2 = sign_x * xh_ms_values[i] if i < (div - 1) else sign_x * x2_g_ms
214             y2 = sign_y * abs(y_g_ms)
215             z2 = abs(zg_ms_values[i]) if i < (div - 1) else abs(z2_g_ms)
216
217         if ind == 0:
218             x1 = xh_ms_values[i - 1] if i > 0 else x1_g_ms
219             y1 = sign_y * abs(y_g_ms)
220             z1 = abs(zg_ms_values[i - 1]) if i > 0 else z1_g_ms
221
222             x2 = xh_ms_values[i] if i < (div - 1) else x3_g_ms
223             y2 = sign_y * abs(y_g_ms)
224             z2 = abs(zg_ms_values[i]) if i < (div - 1) else z1_g_ms
225
226         output += f"\nSLNB_X1_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_X2_{round(
227             (x2,3)}_{round(y2,3)}_{round(z2,3)}"
228         plt_model.append([(x1,y1,z1),(x2,y2,z2)])
229         sln.append(elnr)
230
231         sln_w_yy.append(elnr)
232         sln_w_zz.append(elnr)
233
234         if sign_y == -1:
235             sln_w_xx.append(elnr)
236
237     return output # Return the appended lines
238
239 def print_lines_g_ms_mid(start_num, sign_y, sno):
240     output = ""
241     elnr = start_num
242     output += f"\nSLN{u*10}{elnr}_GRP4_STYP'B'_SNO_{sno}_TITL_\"Line\""
243
244     x1 = x2_g_ms
245     y1 = sign_y * abs(y_g_ms)
246     z1 = z2_g_ms
247
248     x2 = -1*x2_g_ms
249     y2 = sign_y * abs(y_g_ms)
250     z2 = z2_g_ms
251
252     output += f"\nSLNB_X1_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_X2_{round(x2,
253         3)}_{round(y2,3)}_{round(z2,3)}"
254     plt_model.append([(x1,y1,z1),(x2,y2,z2)])
255     sln.append(elnr)
256
257     sln_w_yy.append(elnr)
258     sln_w_zz.append(elnr)
259     sln_w_xx_p.append(elnr)
260
261     return output # Return the appended lines
262
263 # Collect results from multiple calls
264 result_g_ms = ""
265
266 if con_ms == 1:

```

```

265     result_g_ms = (
266         print_lines_g_ms(elnr_g_ms, -1, -1, 2, div, ind) + # Negative xh_ms, negative
            y_g_ms
267         print_lines_g_ms(elnr_g_ms + div_ms, -1, 1, 1, div, ind) + # Negative xh_ms,
            positive y_g_ms
268         print_lines_g_ms(elnr_g_ms + 2 * div_ms, 1, -1, 1, div, ind) + # Positive xh_ms,
            negative y_g_ms
269         print_lines_g_ms(elnr_g_ms + 3 * div_ms, 1, 1, 2, div, ind) + # Positive xh_ms,
            positive y_g_ms
270         print_lines_g_ms_mid(880, -1, 1) +
271         print_lines_g_ms_mid(881, 1, 2)
272     )
273
274     elif con_ms == 0:
275         result_g_ms = (
276             print_lines_g_ms(elnr_g_ms, 1, -1, 1, div, ind) + # Negative xh_ms, negative
                y_g_ms
277             print_lines_g_ms(elnr_g_ms + div, 1, 1, 2, div, ind) # Negative xh_ms, positive
                y_g_ms
278         )
279
280     #endregion
281
282     # Cable values (SA theory)
283     sno_mc = 6
284
285     v0_mid = (4 * (v0 - v1) / (1 ** 2)) * (1 / 2) ** 2 + (-4 * (v0 - v1) / 1) * (1 / 2) #
        curvature mid span [m]
286
287     v0_bar = (4 * (v0 - v1) / (1 ** 2)) * (l_ss) ** 2 + (-4 * (v0 - v1) / 1) * (l_ss) #
        curvature at pylons [m]
288
289     vy = 2.990 + c_d # position cable at midspan [m]
290
291     fz = zp - v1 # drape in vertical direction [m]
292     fy = yp - vy # drape in horizontal direction [m]
293
294     Hw = (wd * l_ms ** 2) / (8 * (fz + v0_mid - v0_bar))
295
296     Hw_p = (wd * l_ms ** 2) / (8 * (fz_p + v0_mid - v0_bar))
297
298     #region C) Cable elements side span
299     elnr_c_ss = 900
300     grp_ss = np.arange(21, 21 + div_ss + 1, 1)
301
302     # constants linear function for sidespan
303     a1_c_ss_lin = ((hp + 0.85) - v0) / l_ss
304     b1_c_ss_lin = v0
305
306     # constants curvature bridge
307     a_curv = 4 * (v0 - v1) / (1 ** 2)
308     b_curv = -4 * (v0 - v1) / 1
309
310     # x-values cables
311     delt_x_ss = (abs(x1_g_ss) - abs(x2_g_ss)) / (div_ss)
312
313     # y-values cables
314     ay_ss = (fy * fact1) / (l_ss ** 2)
315     yc_ss_values = [ay_ss * (i * delt_x_ss) ** 2 + vy
316                     for i in range(1, div_ss)
317     ]
318
319     # z-values cables
320     zc_ss_values = [a1_c_ss_lin * (i * delt_x_ss) + b1_c_ss_lin -
321                     ((wd / (2 * Hw)) * (i * delt_x_ss) * (l_ss - (i * delt_x_ss))) -
322                     (a_curv * (i * delt_x_ss) ** 2 + b_curv * (i * delt_x_ss)) +
323                     (v0_bar * (i * delt_x_ss) / l_ss)
324                     for i in range(1, div_ss)
325     ]
326
327     def print_lines_c_ss(start_num, sign_x, sign_y, grp, sno):
328         output = "" # Initialize an empty string to store the appended lines
329         for i in range(div_ss):
330             elnr = start_num + i
331             output += f"\nSLN_{elnr}_GRP_{grp[i]}_STYP_{cab_type[cables]}_SNO_{sno}_
                TITL_\"Line\"
332
333             x1 = sign_x * xh_ss_values[i - 1] if i > 0 else sign_x * x1_g_ss
334             y1 = sign_y * abs(yc_ss_values[i - 1]) if i > 0 else sign_y * vy
335             z1 = abs(zc_ss_values[i - 1]) if i > 0 else z1_g_ss

```

```

336
337     x2 = sign_x * xh_ss_values[i] if i < hang_ss else sign_x * x2_g_ss
338     y2 = sign_y * abs(yc_ss_values[i]) if i < hang_ss else sign_y * yp
339     z2 = abs(zc_ss_values[i]) if i < hang_ss else zp
340
341     output += f"\nSLNB_{X1_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_{X2_{round(
342         (x2,3)}_{round(y2,3)}_{round(z2,3)}"
343     plt_model.append([(x1,y1,z1),(x2,y2,z2)])
344
345     return output # Return the appended lines
346
347 # Collect results from multiple calls
348 result_c_ss = (
349     print_lines_c_ss(elnr_c_ss, -1, -1, grp_ss, sno_mc) + # Negative xh_ms, negative
350     y_g_ms
351     print_lines_c_ss(elnr_c_ss + div_ss, -1, 1, grp_ss, sno_mc) + # Negative xh_ms,
352     positive y_g_ms
353     print_lines_c_ss(elnr_c_ss + 2 * div_ss, 1, -1, grp_ss, sno_mc) + # Positive xh_ms,
354     negative y_g_ms
355     print_lines_c_ss(elnr_c_ss + 3 * div_ss, 1, 1, grp_ss, sno_mc) # Positive xh_ms,
356     positive y_g_ms
357 )
358 #endregion
359
360 #region D) Cable elements main span
361 elnr_c_ms = 1000
362
363 # constants curvature bridge
364 a_curv = 4 * (v0 - v1) / (l ** 2)
365 b_curv = -4 * (v0 - v1) / l
366
367 if con_ms == 1:
368     grp_ms = np.arange(21+div_ss, 21+div_ss+div_ms+1, 1)
369
370     delt_x_ms = (abs(x1_g_ms) - abs(x2_g_ms)) / (div_ms)
371
372     xc_ms_values = [(i * delt_x_ms) - (l_ms / 2) for i in range(1, div_ms)]
373
374     ay_ms = (fy * fact2) / (l_ms_1 ** 2)
375     yc_ms_values = [round(ay_ms * (l_ms_1 - (i * delt_x_ms)) ** 2 + vy, 3) for i in range
376         (1, div_ms)]
377
378     zc_ms_values = [zp - ((wd / (2 * Hw)) * (i * delt_x_ms) * (l_ms - (i * delt_x_ms)) -
379         (a_curv * (i * delt_x_ms + l_ss) ** 2 + b_curv * (i * delt_x_ms + l_ss)) +
380         v0_bar) for i in range(1, div_ms)]
381
382     div = div_ms
383     ind = 1
384
385 elif con_ms == 0:
386     grp_ms = np.arange(21+div_ss, 21+div_ss+div_ms_p, 1)
387
388     delt_x_ms = l_ms / (div_ms_p)
389
390     xc_ms_values = [(i * delt_x_ms) - (l_ms / 2) for i in range(1, div_ms_p)]
391
392     ay_ms = (fy * fact2) / ((l_ms / 2) ** 2)
393     yc_ms_values = [round(ay_ms * ((l_ms / 2) - (i * delt_x_ms)) ** 2 + vy, 3) for i in
394         range(1, div_ms_p)]
395
396     zc_ms_values = [zp - ((wd / (2 * Hw_p)) * (i * delt_x_ms) * (l_ms - (i * delt_x_ms))
397         -
398         (a_curv * (i * delt_x_ms + l_ss) ** 2 + b_curv * (i * delt_x_ms + l_ss)) +
399         v0_bar) for i in range(1, div_ms_p)]
400
401     div = div_ms_p
402     ind = 0
403
404 def print_lines_c_ms(start_num, sign_x, sign_y, grp, sno, div, ind):
405     output = "" # Initialize an empty string to store the appended lines
406     for i in range(div):
407         elnr = start_num + i
408         output += f"\nSLN_{elnr}_{GRP_{grp[i]}}_{STYP_{cab_type[cables]}}_{SNO_{sno}}_{
409             TITL_{Line}"
410
411     if ind == 1:
412         x1 = sign_x * abs(xh_ms_values[i - 1]) if i > 0 else sign_x * abs(x1_g_ms)
413         y1 = sign_y * abs(yc_ms_values[i - 1]) if i > 0 else sign_y * yp

```



```

407         z1 = abs(zc_ms_values[i - 1]) if i > 0 else zp
408
409         x2 = sign_x * abs(xh_ms_values[i]) if i < (div - 1) else sign_x * abs(
410             x2_g_ms)
411         y2 = sign_y * abs(yc_ms_values[i]) if i < (div - 1) else sign_y * vy
412         z2 = abs(zc_ms_values[i]) if i < (div - 1) else z2_g_ms
413
414     elif ind == 0:
415         x1 = xc_ms_values[i - 1] if i > 0 else x1_g_ms
416         y1 = sign_y * abs(yc_ms_values[i - 1]) if i > 0 else sign_y * yp
417         z1 = abs(zc_ms_values[i - 1]) if i > 0 else zp
418
419         x2 = xc_ms_values[i] if i < (div - 1) else x3_g_ms
420         y2 = sign_y * abs(yc_ms_values[i]) if i < (div - 1) else sign_y * yp
421         z2 = abs(zc_ms_values[i]) if i < (div - 1) else zp
422
423     output += f"\nSLNB_{X1}_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_{X2}_{round(
424         (x2,3)}_{round(y2,3)}_{round(z2,3)}"
425     plt_model.append([(x1,y1,z1), (x2,y2,z2)])
426
427     return output # Return the appended lines
428
429 def print_lines_c_ms_mid(start_num, sign_y, sno):
430     output = ""
431     elnr = start_num
432     output += f"\nSLN_{elnr}_{GRP}_{30}_{STYP}_{cab_type[cables]}_{SNO}_{sno}_{TITL}\nLine
433         \"
434
435     x1 = x2_g_ms
436     y1 = sign_y * abs(vy)
437     z1 = z2_g_ms
438
439     x2 = -1*x2_g_ms
440     y2 = sign_y * abs(vy)
441     z2 = z2_g_ms
442
443     output += f"\nSLNB_{X1}_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_{X2}_{round(x2,
444         3)}_{round(y2,3)}_{round(z2,3)}"
445     plt_model.append([(x1,y1,z1), (x2,y2,z2)])
446
447     return output # Return the appended lines
448
449 # Collect results from multiple calls
450 if con_ms == 1:
451     result_c_ms = (
452         print_lines_c_ms(elnr_c_ms, -1, -1, grp_ms, sno_mc, div, ind) + # Negative xh_ms
453         , negative y_g_ms
454         print_lines_c_ms(elnr_c_ms + div, -1, 1, grp_ms, sno_mc, div, ind) + # Negative
455         xh_ms, positive y_g_ms
456         print_lines_c_ms(elnr_c_ms + 2 * div, 1, -1, grp_ms, sno_mc, div, ind) + #
457         Negative xh_ms, positive y_g_ms
458         print_lines_c_ms(elnr_c_ms + 3 * div, 1, 1, grp_ms, sno_mc, div, ind) +
459         print_lines_c_ms_mid(1080, -1, sno_mc) + # Negative xh_ms, positive y_g_ms
460         print_lines_c_ms_mid(1081, 1, sno_mc) # Negative xh_ms, positive y_g_ms
461     )
462 elif con_ms == 0:
463     result_c_ms = (
464         print_lines_c_ms(elnr_c_ms, 1, -1, grp_ms, sno_mc, div, ind) + # Negative xh_ms,
465         negative y_g_ms
466         print_lines_c_ms(elnr_c_ms + div, 1, 1, grp_ms, sno_mc, div, ind) # Negative
467         xh_ms, positive y_g_ms
468     )
469 #endregion
470
471 #region E) Through elements side span
472 el_nr_tr = 1200
473 grp_tr = np.arange(6, 8, 1)
474 sno_tr = np.arange(3, 5, 1)
475 sp_tr_ss = 0.599 # spacing throughs
476
477 x_tr_g_ss_values = [x1_g_ss] + xh_ss_values + [x2_g_ss]
478 z_tr_g_ss_values = [z1_g_ss] + zg_ss_values + [z2_g_ss]
479
480 def print_lines_tr_ss(start_num, sign_x, grp, sno, div_ss):
481     x_tr_ss_values = []
482     z_tr_ss_values = []
483     x_tr_ss_values_end = []
484     z_tr_ss_values_end = []

```

```

478
479
480 # Generate x_tr_ss_values and z_tr_ss_values based on provided x_tr_g_ss_values and
      z_tr_g_ss_values
481 for i in range(len(x_tr_g_ss_values) - 1):
482     nr_tr_ss_values = int(np.sqrt( (x_tr_g_ss_values[i + 1] - x_tr_g_ss_values[i]) **
      2 +
483                                     (z_tr_g_ss_values[i + 1] - z_tr_g_ss_values[i]) ** 2)
      / sp_tr_ss)
484
485     del_x_tr_ss_values = (abs(x_tr_g_ss_values[i]) - abs(x_tr_g_ss_values[i + 1])) /
      nr_tr_ss_values
486     del_z_tr_ss_values = (z_tr_g_ss_values[i + 1] - z_tr_g_ss_values[i]) /
      nr_tr_ss_values
487
488     for j in range(nr_tr_ss_values):
489         x_tr_ss_values.append(x_tr_g_ss_values[i] + j * del_x_tr_ss_values)
490         z_tr_ss_values.append(z_tr_g_ss_values[i] + j * del_z_tr_ss_values)
491
492     x_tr_ss_values_end.append(del_x_tr_ss_values + abs(x_tr_g_ss_values[-1]))
493     z_tr_ss_values_end.append(z_tr_ss_values[-1])
494
495
496 # Generate lines based on the created x_tr_ss_values and z_tr_ss_values lists
497
498 output = "" # Initialize an empty string to store the appended lines
499
500 for i in range(len(x_tr_ss_values) - 1):
501     elnr = start_num + i
502
503     output += f"\nSLN{elnr}_GRP{grp[0]}_STYP{'N'}_SNO{sno[0]}_TITL\"Line\""
504
505     x1 = sign_x * abs(x_tr_ss_values[i])
506     y1 = -y_g_ss # y1 set to vy as requested
507     z1 = z_tr_ss_values[i]
508
509     x2 = sign_x * abs(x_tr_ss_values[i])
510     y2 = y_g_ss # y2 set to vy as requested
511     z2 = z_tr_ss_values[i]
512
513     output += f"\nSLNB{X1}{round(x1,3)}{round(y1,3)}{round(z1,3)}X2{round(
      (x2,3)}{round(y2,3)}{round(z2,3)}"
514     plt_model.append([(x1, y1, z1), (x2, y2, z2)])
515
516 for j in range(2):
517
518     output += f"\nSLN{elnr+j+1}_GRP{grp[0]}_STYP{'N'}_SNO{sno[0]}_TITL\"
      Line\""
519
520     x1 = sign_x * x_tr_ss_values_end[0]
521     y1 = -y_g_ss # y1 set to vy as requested
522     z1 = z_tr_ss_values_end[0]
523
524     x2 = sign_x * x_tr_ss_values_end[0]
525     y2 = y_g_ss # y2 set to vy as requested
526     z2 = z_tr_ss_values_end[0]
527
528     output += f"\nSLNB{X1}{round(x1,3)}{round(y1,3)}{round(z1,3)}X2{round(
      (x2,3)}{round(y2,3)}{round(z2,3)}"
529     plt_model.append([(x1, y1, z1), (x2, y2, z2)])
530
531
532 return output # Return the appended lines
533
534
535 result_tr_ss = (
536     print_lines_tr_ss(el_nr_tr, -1, grp_tr, sno_tr, div_ss) + # Negative xh_ms, negative
      y_g_ms
537     print_lines_tr_ss(el_nr_tr + 100, 1, grp_tr, sno_tr, div_ss) # Negative xh_ms,
      positive y_g_ms
538 )
539 #endregion
540
541
542 #region F) Through elements main span
543 el_nr_tr_ms = 1600
544 sp_tr_ms = 0.576 # spacing throughs
545
546 if con_ms == 1:
547     x_tr_g_ms_values = [x1_g_ms] + xh_ms_values + [x2_g_ms]

```

```

548     z_tr_g_ms_values = [z1_g_ms] + zg_ms_values + [z2_g_ms]
549     x_tr_g_ms_values_mid = [x2_g_ms, abs(x2_g_ms)]
550     z_tr_g_ms_values_mid = [z2_g_ms, z2_g_ms]
551
552     if con_ms == 0:
553         x_tr_g_ms_values = [x1_g_ms] + xh_ms_values + [x3_g_ms]
554         z_tr_g_ms_values = [z1_g_ms] + zg_ms_values + [z1_g_ms]
555
556
557     def print_lines_tr_ms(start_num, sign_x, grp, sno, x_tr_g_ms_values, z_tr_g_ms_values,
558                          div):
559
560         x_tr_ms_values = []
561         z_tr_ms_values = []
562
563         # Generate x_tr_ss_values and z_tr_ss_values based on provided x_tr_g_ms_values and
564         # z_tr_g_ms_values
565         for i in range(len(x_tr_g_ms_values) - 1):
566             nr_tr_ms_values = int(np.sqrt((x_tr_g_ms_values[i] - x_tr_g_ms_values[i + 1]) **
567                                           2 +
568                                           (z_tr_g_ms_values[i] - z_tr_g_ms_values[i + 1]) ** 2)
569                                   / sp_tr_ms)
570
571             del_x_tr_ms_values = abs((x_tr_g_ms_values[i] - x_tr_g_ms_values[i + 1])) /
572                                   nr_tr_ms_values
573
574             del_z_tr_ms_values = abs((z_tr_g_ms_values[i] - z_tr_g_ms_values[i + 1])) /
575                                   nr_tr_ms_values)
576
577             for j in range(nr_tr_ms_values + 1):
578                 x_tr_ms_values.append(x_tr_g_ms_values[i] + j * del_x_tr_ms_values)
579                 if con_ms == 1:
580                     z_tr_ms_values.append(z_tr_g_ms_values[i] + j * del_z_tr_ms_values)
581                 elif con_ms == 0 and x_tr_g_ms_values[i + 1] <= 0:
582                     z_tr_ms_values.append(z_tr_g_ms_values[i] + j * del_z_tr_ms_values)
583                 else:
584                     z_tr_ms_values.append(z_tr_g_ms_values[i] - j * del_z_tr_ms_values)
585
586         # Generate lines based on the created x_tr_ss_values and z_tr_ss_values lists
587         # Initialize an empty string to store the appended lines
588
589         output = ""
590
591         if con_ms == 0:
592             output += f"\nSLN{0000000}{el_nr_tr_ms+750}GRP{grp[0]}STYP'N'SNO{sno[0]}TITL\"
593             output += f"\nSLNB{X1}{round(abs(x1_g_ms),3)}{round(y_g_ms,3)}{round(z1_g_ms,3)}X2{round(abs(x1_g_ms),3)}{round(abs(y_g_ms),3)}{round(z1_g_ms,3)}"
594
595         for i in range(len(x_tr_ms_values) - 1):
596
597             elnr = start_num + i
598
599             # Check if the index of x_tr_ss_values equals any x_tr_ss_values[j] / len(div_ss)
600             if con_ms == 0:
601                 for j in range(div):
602                     output += f"\nSLN{0000000}{elnr}GRP{grp[0]}STYP'N'SNO{sno[0]}TITL\"
603                     output += f"\nSLN{0000000}{elnr}GRP{grp[0]}STYP'N'SNO{sno[0]}TITL\"
604                     break
605                 # if i == int((len(x_tr_ms_values) / div)) * (j + 1) or i == (int((len(x_tr_ms_values) / div)) * (j + 1) + 1) or i == (int((len(x_tr_ms_values) / div)) * (j + 1) + -1) or i == 0 or i == 1:
606                 # output += f"\nSLN{0000000}{elnr}GRP{grp[1]}STYP'N'SNO{sno[1]}TITL \"Line\"
607                 # break # Exit the loop once a match is found
608                 # else:
609                 # output += f"\nSLN{0000000}{elnr}GRP{grp[0]}STYP'N'SNO{sno[0]}TITL \"Line\"
610                 # break
611
612             elif con_ms == 1:
613                 for j in range(div):
614                     output += f"\nSLN{0000000}{elnr}GRP{grp[0]}STYP'N'SNO{sno[0]}TITL\"
615                     output += f"\nSLN{0000000}{elnr}GRP{grp[0]}STYP'N'SNO{sno[0]}TITL\"
616                     break

```

```

612         # if i == int((len(x_tr_ms_values) / (div / 2 + 1))) * (j + 1) or i == (
        #         int((len(x_tr_ms_values) / (div / 2 + 1))) * (j + 1) + 1) or i == (
        #         int((len(x_tr_ms_values) / (div / 2 + 1))) * (j + 1) + -1) or i == 0
        #         or i == 1:
613         #         output += f"\nSLN          {elnr} GRP {grp[1]} STYP 'N' SNO {sno[1]}
        #         TITL \"Line\""
614         #         break # Exit the loop once a match is found
615         # else:
616         #         output += f"\nSLN          {elnr} GRP {grp[0]} STYP 'N' SNO {sno[0]}
        #         TITL \"Line\""
617         #         break
618
619         x1 = sign_x * x_tr_ms_values[i]
620         y1 = -y_g_ss
621         z1 = z_tr_ms_values[i]
622
623
624         x2 = sign_x * x_tr_ms_values[i]
625         y2 = y_g_ss
626         z2 = z_tr_ms_values[i]
627
628         output += f"\nSLNB_{X1_{round(x1,3)}}_{round(y1,3)}_{round(z1,3)}_{X2_{round
        (x2,3)}}_{round(y2,3)}_{round(z2,3)}"
629         plt_model.append([(x1, y1, z1), (x2, y2, z2)])
630         plt_tr_ms.append([(x1, y1, z1), (x2, y2, z2)])
631
632         return output # Return the appended lines
633
634
635 def print_lines_tr_ms_mid(elnr, grp, sno, x_tr_g_ms_values_mid, z_tr_g_ms_values_mid):
636     output = ""
637     output += f"\nSLN_{elnr}GRP_{grp[0]}STYP'N'SNO_{sno[0]}TITL\"Line\""
638
639     x1 = x_tr_g_ms_values_mid[0]
640     y1 = -y_g_ss
641     z1 = z_tr_g_ms_values_mid[0]
642
643
644     x2 = x_tr_g_ms_values_mid[0]
645     y2 = y_g_ss
646     z2 = z_tr_g_ms_values_mid[0]
647
648     output += f"\nSLNB_{X1_{round(x1,3)}}_{round(y1,3)}_{round(z1,3)}_{X2_{round(x2,
        3)}}_{round(y2,3)}_{round(z2,3)}"
649     plt_model.append([(x1, y1, z1), (x2, y2, z2)])
650
651
652     output += f"\nSLN_{elnr+1}GRP_{grp[0]}STYP'N'SNO_{sno[0]}TITL\"Line\""
653
654     x1 = x_tr_g_ms_values_mid[1]
655     y1 = -y_g_ss
656     z1 = z_tr_g_ms_values_mid[1]
657
658
659     x2 = x_tr_g_ms_values_mid[1]
660     y2 = y_g_ss
661     z2 = z_tr_g_ms_values_mid[1]
662
663     output += f"\nSLNB_{X1_{round(x1,3)}}_{round(y1,3)}_{round(z1,3)}_{X2_{round(x2,
        3)}}_{round(y2,3)}_{round(z2,3)}"
664
665     plt_model.append([(x1, y1, z1), (x2, y2, z2)])
666     return output
667
668
669 if con_ms == 1:
670     result_tr_ms = (
671         print_lines_tr_ms(el_nr_tr_ms, -1, grp_tr, sno_tr, x_tr_g_ms_values,
        z_tr_g_ms_values, div_ms) + # Negative xh_ms, negative y_g_ms
672         print_lines_tr_ms(el_nr_tr_ms + 200, 1, grp_tr, sno_tr, x_tr_g_ms_values,
        z_tr_g_ms_values, div_ms) +
673         print_lines_tr_ms(el_nr_tr_ms + 400, 1, grp_tr, sno_tr, x_tr_g_ms_values_mid,
        z_tr_g_ms_values_mid, div_ms) + # Negative xh_ms, positive y_g_ms
674         print_lines_tr_ms_mid(el_nr_tr_ms + 450, grp_tr, sno_tr, x_tr_g_ms_values_mid,
        z_tr_g_ms_values_mid)
675     )
676 elif con_ms == 0:
677     result_tr_ms = (
678         print_lines_tr_ms(el_nr_tr_ms, 1, grp_tr, sno_tr, x_tr_g_ms_values,
        z_tr_g_ms_values, div_ms_p) # Negative xh_ms, positive y_g_ms

```

```

679 )
680 #endregion
681
682
683 #region G) Pylon elements
684 elnr_pyl = 2600
685 sno_pyl = '8.9'
686 grp_pyl = []
687
688 def print_lines_pyl(start_num, sign_x, sign_y, sno):
689     output = "" # Initialize an empty string to store the appended lines
690
691     elnr = start_num
692     output += f"\nSLN{elnr}GRP{start_num-2300}STYP'N'SNO'{sno}'TITL\"Line
693         \"
694
695     x1 = sign_x * x1_g_ms
696     y1 = sign_y * 2.07
697     z1 = 0.85
698
699     x2 = sign_x * x1_g_ms
700     y2 = sign_y*yp
701     z2 = zp
702
703     output += f"\nSLNB{X1}{round(x1,3)}{round(y1,3)}{round(z1,3)}X2{round(x2,
704         3)}{round(y2,3)}{round(z2,3)}"
705     plt_model.append([(x1,y1,z1),(x2,y2,z2)])
706     plt_pyl.append([(x1,y1,z1),(x2,y2,z2)])
707
708     sln_w_p.append(start_num)
709     pnt_pyl.append([x2, y2, z2])
710     grp_pyl.append(start_num - 2300)
711
712     return output # Return the appended lines
713
714 # Collect results from multiple calls
715 result_pyl = (
716     print_lines_pyl(elnr_pyl + 1, -1, -1, sno_pyl) + # Negative xh_ms, negative y_g_ms
717     print_lines_pyl(elnr_pyl + 2, -1, 1, sno_pyl) + # Negative xh_ms, positive y_g_ms
718     print_lines_pyl(elnr_pyl + 3, 1, -1, sno_pyl) + # Positive xh_ms, negative y_g_ms
719     print_lines_pyl(elnr_pyl + 4, 1, 1, sno_pyl) # Positive xh_ms, positive y_g_ms
720 )
721
722 #endregion
723
724
725 #region H) Plate coupling elements side span
726 sno_p = 20
727 elnr_p_ss = 2700
728
729 def print_lines_p_ss(start_num, sign_x, sign_y, sno):
730     output = "" # Initialize an empty string to store the appended lines
731     for i in range(hang_ss):
732         elnr = start_num + i
733
734         x1 = sign_x * xh_ss_values[i]
735         y1 = sign_y * abs(vy)
736         z1 = zg_ss_values[i]
737
738         x2 = sign_x * xh_ss_values[i]
739         y2 = sign_y * abs(y_g_ss)
740         z2 = zg_ss_values[i]
741
742         output += f"\nSPT{elnr}X{round(x1,3)}{round(y1,3)}{round(z1,3)}"
743         output += f"\nSPT{elnr+50}X{round(x2,3)}{round(y2,3)}{round(z2,3)}"
744         output += f"\nSPTP{TYPE_PPM}REF{elnr}"
745
746         plt_model.append([(x1,y1,z1),(x2,y2,z2)])
747         plt_p_ss.append([(x1,y1,z1),(x2,y2,z2)])
748
749     return output # Return the appended lines
750
751 # Collect results from multiple calls
752 result_p_ss = (
753     print_lines_p_ss(elnr_p_ss, -1, -1, sno_p) + # Negative xh_ms, negative y_g_ms
754     print_lines_p_ss(elnr_p_ss + div_ss, -1, 1, sno_p) + # Negative xh_ms, positive
755     y_g_ms

```

```

756     print_lines_p_ss(elnr_p_ss + 2 * div_ss, 1, -1, sno_mc) + # Positive xh_ms, negative
757     y_g_ms
758     print_lines_p_ss(elnr_p_ss + 3 * div_ss, 1, 1, sno_mc) # Positive xh_ms, positive
759     y_g_ms
760 )
761
762 #endregion
763
764 #region I) Plate coupling elements main span
765 sno_p = 20
766 elnr_p_ms = 2800
767
768 if con_ms == 1:
769     ind = hang_ms
770     x_p_ms = [x2_g_ms, abs(x2_g_ms)]
771     z_p_ms = [z2_g_ms, abs(z2_g_ms)]
772
773 elif con_ms == 0:
774     ind = hang_ms_p
775
776 def print_lines_p_ms(start_num, sign_x, sign_y, sno, ind, xh_ms_values, zg_ms_values):
777     output = "" # Initialize an empty string to store the appended lines
778     for i in range(ind):
779         elnr = start_num + i
780
781         if con_ms == 1:
782             x1 = sign_x * xh_ms_values[i]
783             y1 = sign_y * abs(vy)
784             z1 = zg_ms_values[i]
785
786             x2 = sign_x * xh_ms_values[i]
787             y2 = sign_y * abs(y_g_ms)
788             z2 = zg_ms_values[i]
789
790         elif con_ms == 0:
791             x1 = sign_x * xh_ms_values[i]
792             y1 = sign_y * abs(vy)
793             z1 = zg_ms_values[i]
794
795             x2 = sign_x * xh_ms_values[i]
796             y2 = sign_y * abs(y_g_ms)
797             z2 = zg_ms_values[i]
798
799         output += f"\n\SPT_{elnr}_{X_{round(x1,3)}}_{round(y1,3)}_{round(z1,3)}"
800         output += f"\n\SPT_{elnr+50}_{X_{round(x2,3)}}_{round(y2,3)}_{round(z2,3)}"
801         output += f"\n\SPT_{TYPE_PPM_REF_{elnr}}"
802
803         plt_model.append([(x1,y1,z1),(x2,y2,z2)])
804         plt_p_ms.append([(x1,y1,z1),(x2,y2,z2)])
805     return output # Return the appended lines
806
807
808 # Collect results from multiple calls
809 if con_ms == 1:
810     result_p_ms = (
811         print_lines_p_ms(elnr_p_ms, -1, -1, sno_p, ind, xh_ms_values, zg_ms_values) + #
812         Negative xh_ms, negative y_g_ms
813         print_lines_p_ms(elnr_p_ms + 20, -1, 1, sno_p, ind, xh_ms_values, zg_ms_values) +
814         # Negative xh_ms, positive y_g_ms
815         print_lines_p_ms(elnr_p_ms + 40, 1, -1, sno_p, ind, xh_ms_values, zg_ms_values) +
816         print_lines_p_ms(elnr_p_ms + 60, 1, 1, sno_p, ind, xh_ms_values, zg_ms_values) +
817         print_lines_p_ms(elnr_p_ms + 80, 1, -1, sno_p, 2, x_p_ms, z_p_ms) +
818         print_lines_p_ms(elnr_p_ms + 100, 1, 1, sno_p, 2, x_p_ms, z_p_ms) # Positive
819         xh_ms, positive y_g_ms
820     )
821 if con_ms == 0:
822     result_p_ms = (
823         print_lines_p_ms(elnr_p_ms, 1, -1, sno_p, ind, xh_ms_values, zg_ms_values) +
824         print_lines_p_ms(elnr_p_ms + 20, 1, 1, sno_p, ind, xh_ms_values, zg_ms_values)
825     )
826
827 #endregion
828
829 #region J) Vertical hangers side span
830 elnr_h_ss = 3000
831 grp_h_ss = np.arange(41, 41 + hang_ss, 1)

```

```

831 sno_h_ss = 7
832
833
834 def print_lines_h_ss(start_num, sign_x, sign_y, grp, sno):
835     output = "" # Initialize an empty string to store the appended lines
836     for i in range(hang_ss):
837         elnr = start_num + i
838         output += f"\nSLN{elnr}_GRP{grp[i]}_STYP{'{cab_type[cables]}'_SNO{sno}_TITL\"Line\""}
839
840         x1 = sign_x * xh_ss_values[i]
841         y1 = sign_y * abs(vy)
842         z1 = zg_ss_values[i]
843
844         x2 = sign_x * xh_ss_values[i]
845         y2 = sign_y * yc_ss_values[i]
846         z2 = zc_ss_values[i]
847
848         output += f"\nSLNB{X1_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_X2_{round(x2,3)}_{round(y2,3)}_{round(z2,3)}"
849         plt_model.append([(x1,y1,z1),(x2,y2,z2)])
850
851     return output # Return the appended lines
852
853
854 # Collect results from multiple calls
855 result_h_ss = (
856     print_lines_h_ss(elnr_h_ss, -1, -1, grp_h_ss, sno_mc) + # Negative xh_ms, negative
857     print_lines_h_ss(elnr_h_ss + div_ss, -1, 1, grp_h_ss, sno_mc) + # Negative xh_ms,
858     print_lines_h_ss(elnr_h_ss + 2 * div_ss, 1, -1, grp_h_ss, sno_mc) + # Positive xh_ms,
859     print_lines_h_ss(elnr_h_ss + 3 * div_ss, 1, 1, grp_h_ss, sno_mc) # Positive xh_ms,
860     positive y_g_ms
861 )
862 #endregion
863
864 #region K) Vertical hangers main span
865 elnr_h_ms = 3200
866 sno_h_ms = 7
867
868
869 if con_ms == 1:
870     grp_h_ms = np.arange(41 + hang_ss, 41 + hang_ss + hang_ms, 1)
871     ind = hang_ms
872 else:
873     grp_h_ms = np.arange(41 + hang_ss, 41 + hang_ss + hang_ms_p, 1)
874     ind = hang_ms_p
875
876 def print_lines_h_ms(start_num, sign_x, sign_y, grp, sno, ind):
877     output = "" # Initialize an empty string to store the appended lines
878
879     for i in range(ind):
880         elnr = start_num + i
881         output += f"\nSLN{elnr}_GRP{grp[i]}_STYP{'{cab_type[cables]}'_SNO{sno}_TITL\"Line\""}
882
883         if con_ms == 1:
884             x1 = sign_x * xh_ms_values[i]
885             y1 = sign_y * abs(vy)
886             z1 = zg_ms_values[i]
887
888             x2 = sign_x * xh_ms_values[i]
889             y2 = sign_y * yc_ms_values[i]
890             z2 = zc_ms_values[i]
891
892         elif con_ms == 0:
893             x1 = xh_ms_values[i]
894             y1 = sign_y * abs(vy)
895             z1 = zg_ms_values[i]
896
897             x2 = xh_ms_values[i]
898             y2 = sign_y * yc_ms_values[i]
899             z2 = zc_ms_values[i]
900
901         output += f"\nSLNB{X1_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_X2_{round(x2,3)}_{round(y2,3)}_{round(z2,3)}"
902         plt_model.append([(x1,y1,z1),(x2,y2,z2)])

```

```

903     return output # Return the appended lines
904
905 # Collect results from multiple calls
906 if con_ms == 1:
907     result_h_ms = (
908         print_lines_h_ms(elnr_h_ms, -1, -1, grp_h_ms, sno_h_ms, ind) + # Negative xh_ms,
909             negative y_g_ms
910         print_lines_h_ms(elnr_h_ms + 50, -1, 1, grp_h_ms, sno_h_ms, ind) + # Negative
911             xh_ms, positive y_g_ms
912         print_lines_h_ms(elnr_h_ms + 100, 1, -1, grp_h_ms, sno_h_ms, ind) + # Positive
913             xh_ms, negative y_g_ms
914         print_lines_h_ms(elnr_h_ms + 150, 1, 1, grp_h_ms, sno_h_ms, ind) # Positive
915             xh_ms, positive y_g_ms
916     )
917 elif con_ms == 0:
918     result_h_ms = (
919         print_lines_h_ms(elnr_h_ms, 1, -1, grp_h_ms, sno_h_ms, ind) + # Negative xh_ms,
920             negative y_g_ms
921         print_lines_h_ms(elnr_h_ms + 50, 1, 1, grp_h_ms, sno_h_ms, ind) # Negative xh_ms,
922             positive y_g_ms
923     )
924 #endregion
925
926 #region L) Deck plate elements side span
927 elnr_d_pl_ss1 = np.arange(1, div_ss + 1, 1)
928 elnr_d_pl_ss2 = np.arange(div_ss + 1, div_ss * 2 + 1, 1)
929 grp_d_pl_ss1 = np.arange(10, 10 + div_ss + 1, 1)
930 grp_d_pl_ss2 = np.arange(10 + div_ss, 10 + div_ss * 2 + 1, 1)
931
932 def print_lines_d_pl_ss(start_num, sign_x, grp):
933     output = "" # Initialize an empty string to store the appended lines
934     for i in range(div_ss):
935         elnr = start_num
936         output += f"\nSAR_{elnr[i]}_GRP_{grp[i]}_MNO_1_NRA_7_T_10_CB_0_CT_0_MCTL_REGM_
937             TITL_\"Area\"_nSARB_OUT"
938
939         x1 = sign_x * abs(xh_ss_values[i - 1]) if i > 0 else sign_x * abs(x1_g_ss)
940         y1 = - abs(y_g_ss)
941         z1 = zg_ss_values[i - 1] if i > 0 else z1_g_ss
942
943         x2 = sign_x * abs(xh_ss_values[i]) if i < (div_ss - 1) else sign_x * abs(x2_g_ss)
944         y2 = - abs(y_g_ss)
945         z2 = zg_ss_values[i] if i < (div_ss - 1) else z2_g_ss
946
947         x3 = sign_x * abs(xh_ss_values[i]) if i < (div_ss - 1) else sign_x * abs(x2_g_ss)
948         y3 = abs(y_g_ss)
949         z3 = zg_ss_values[i] if i < (div_ss - 1) else z2_g_ss
950
951         x4 = sign_x * abs(xh_ss_values[i - 1]) if i > 0 else sign_x * abs(x1_g_ss)
952         y4 = abs(y_g_ss)
953         z4 = zg_ss_values[i - 1] if i > 0 else z1_g_ss
954
955         output += (f"\nSLNB_{X1_{round(x1,3)}}_{round(y1,3)}}_{round(z1,3)}}_X2_{
956             round(x2,3)}}_{round(y2,3)}}_{round(z2,3)}}_nSARB_OUT"
957         f"\nSLNB_{X1_{round(x2,3)}}_{round(y2,3)}}_{round(z2,3)}}_X2_{round(x3,3)}}_{round
958             (y3,3)}}_{round(z3,3)}}_nSARB_OUT"
959         f"\nSLNB_{X1_{round(x3,3)}}_{round(y3,3)}}_{round(z3,3)}}_X2_{round(x4,3)}}_{round
960             (y4,3)}}_{round(z4,3)}}_nSARB_OUT"
961         f"\nSLNB_{X1_{round(x4,3)}}_{round(y4,3)}}_{round(z4,3)}}_X2_{round(x1,3)}}_{round
962             (y1,3)}}_{round(z1,3)}}")
963
964         plt_model_deck.append([(x1,y1,z1),(x2,y2,z2),(x3,y3,z3),(x4,y4,z4)])
965         sar_ss.append(elnr[i])
966
967     return output # Return the appended lines
968
969 result_d_pl_ss = (
970     print_lines_d_pl_ss(elnr_d_pl_ss1, -1, grp_d_pl_ss1 + 100) +
971     print_lines_d_pl_ss(elnr_d_pl_ss2, 1, grp_d_pl_ss2 + 100)
972 )
973
974 #endregion

```



```

972 #region M) Deck plate elements main span
973
974 if con_ms == 1:
975     div = div_ms
976     x_d_pl_ms = [x2_g_ms, abs(x2_g_ms)]
977     z_d_pl_ms = [z2_g_ms, abs(z2_g_ms)]
978
979     grp_d_pl_ms1 = np.arange(10 + div_ss * 2, 10 + div_ss * 2 + div_ms + 1, 1)
980     grp_d_pl_ms2 = np.arange(10 + div_ss * 2 + div_ms, 10 + div_ss * 2 + div_ms * 2, 1)
981
982     elnr_d_pl_ms1 = np.arange(div_ss * 2 + 1, div_ss * 2 + div_ms + 1, 1)
983     elnr_d_pl_ms2 = np.arange(div_ss * 2 + div_ms + 1, div_ss * 2 + div_ms * 2 + 1, 1)
984
985
986 elif con_ms == 0:
987     div = hang_ms_p + 1
988     grp_d_pl_ms = np.arange(10 + div_ss * 2, 10 + div_ss * 2 + div_ms_p + 1, 1)
989     elnr_d_pl_ms = np.arange(div_ss * 2 + 1, div_ss * 2 + div_ms_p + 1, 1)
990
991 def print_lines_d_pl_ms(start_num, sign_x, grp, div, xh_ms_values, zg_ms_values):
992     output = "" # Initialize an empty string to store the appended lines
993     for i in range(div):
994         elnr = start_num
995         output += f"\nSAR_{elnr[i]}_GRP_{grp[i]}_MNO_1_NRA_7_T_10_CB_0_CT_0_MCTL_REGM_
996             TITL_\"Area_\"_nSARB_OUT"
997
998         if con_ms == 1:
999             x1 = sign_x * abs(xh_ms_values[i - 1]) if i > 0 else sign_x * abs(x1_g_ms)
1000             y1 = - abs(y_g_ms)
1001             z1 = zg_ms_values[i - 1] if i > 0 else z1_g_ms
1002
1003             x2 = sign_x * abs(xh_ms_values[i]) if i < (div_ms - 1) else sign_x * abs(
1004                 x2_g_ms)
1005             y2 = - abs(y_g_ms)
1006             z2 = zg_ms_values[i] if i < (div_ms - 1) else z2_g_ms
1007
1008             x3 = sign_x * abs(xh_ms_values[i]) if i < (div_ms - 1) else sign_x * abs(
1009                 x2_g_ms)
1010             y3 = abs(y_g_ms)
1011             z3 = zg_ms_values[i] if i < (div_ms - 1) else z2_g_ms
1012
1013             x4 = sign_x * abs(xh_ms_values[i - 1]) if i > 0 else sign_x * abs(x1_g_ms)
1014             y4 = abs(y_g_ms)
1015             z4 = zg_ms_values[i - 1] if i > 0 else z1_g_ms
1016
1017         elif con_ms == 0:
1018             x1 = xh_ms_values[i - 1] if i > 0 else x1_g_ms
1019             y1 = - abs(y_g_ms)
1020             z1 = zg_ms_values[i - 1] if i > 0 else z1_g_ms
1021
1022             x2 = xh_ms_values[i] if i < (div_ms_p - 1) else x3_g_ms
1023             y2 = - abs(y_g_ms)
1024             z2 = zg_ms_values[i] if i < (div_ms_p - 1) else z1_g_ms
1025
1026             x3 = xh_ms_values[i] if i < (div_ms_p - 1) else x3_g_ms
1027             y3 = abs(y_g_ms)
1028             z3 = zg_ms_values[i] if i < (div_ms_p - 1) else z1_g_ms
1029
1030             x4 = xh_ms_values[i - 1] if i > 0 else x1_g_ms
1031             y4 = abs(y_g_ms)
1032             z4 = zg_ms_values[i - 1] if i > 0 else z1_g_ms
1033
1034         output += (f"\nSLNB_{X1_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_X2_{
1035             round(x2,3)}_{round(y2,3)}_{round(z2,3)}}_nSARB_OUT"
1036             f"\nSLNB_{X1_{round(x2,3)}_{round(y2,3)}_{round(z2,3)}_X2_{round(x3,3)}_{round(
1037                 y3,3)}_{round(z3,3)}}_nSARB_OUT"
1038             f"\nSLNB_{X1_{round(x3,3)}_{round(y3,3)}_{round(z3,3)}_X2_{round(x4,3)}_{round(
1039                 y4,3)}_{round(z4,3)}}_nSARB_OUT"
1040             f"\nSLNB_{X1_{round(x4,3)}_{round(y4,3)}_{round(z4,3)}_X2_{round(x1,3)}_{round(
1041                 y1,3)}_{round(z1,3)}}")
1042
1043         plt_model_deck.append([(x1,y1,z1),(x2,y2,z2),(x3,y3,z3),(x4,y4,z4)])
1044         sar_ms.append(elnr[i])
1045
1046     return output # Return the appended lines
1047
1048 def print_lines_d_pl_ms_mid(start_num, sign_x, grp, div, xh_ms_values, zg_ms_values):
1049     output = "" # Initialize an empty string to store the appended lines

```

```

1045     for i in range(1):
1046         elnr = start_num + i
1047         output += f"\nSAR_{elnr}_{GRP}_{grp}_{MNO}_{1}_{NRA}_{7}_{T}_{10}_{CB}_{0}_{CT}_{0}_{MCTL}_{REGM}_{TITL}_{"
1048             Area"\nSARB_OUT"
1049
1050         x1 = xh_ms_values[0]
1051         y1 = - abs(y_g_ms)
1052         z1 = zg_ms_values[0]
1053
1054         x2 = xh_ms_values[1]
1055         y2 = - abs(y_g_ms)
1056         z2 = zg_ms_values[1]
1057
1058         x3 = xh_ms_values[1]
1059         y3 = abs(y_g_ms)
1060         z3 = zg_ms_values[1]
1061
1062         x4 = xh_ms_values[0]
1063         y4 = abs(y_g_ms)
1064         z4 = zg_ms_values[0]
1065
1066         output += (f"\nSLNB_{X1}_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_{X2}_{"
1067             round(x2,3)}_{round(y2,3)}_{round(z2,3)}_{round(z3,3)}_{round(z4,3)}_{round(x3,3)}_{round(y3,3)}_{round(z3,3)}_{round(z4,3)}_{round(x4,3)}_{round(y4,3)}_{round(z4,3)}_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}")
1068
1069         plt_model_deck.append([(x1,y1,z1),(x2,y2,z2),(x3,y3,z3),(x4,y4,z4)])
1070         sar_ms.append(elnr_d_pl_ms2[-1] + 1)
1071
1072     return output # Return the appended lines
1073
1074
1075 if con_ms == 1:
1076     result_d_pl_ms = (
1077         print_lines_d_pl_ms(elnr_d_pl_ms1, -1, grp_d_pl_ms1 + 100, div, xh_ms_values,
1078             zg_ms_values) +
1079         print_lines_d_pl_ms(elnr_d_pl_ms2, 1, grp_d_pl_ms2 + 100, div, xh_ms_values,
1080             zg_ms_values) +
1081         print_lines_d_pl_ms_mid(elnr_d_pl_ms2[-1] + 1, 1, grp_d_pl_ms2[-1] + 1 + 100,
1082             div, x_d_pl_ms, z_d_pl_ms)
1083     )
1084 elif con_ms == 0:
1085     result_d_pl_ms = (
1086         print_lines_d_pl_ms(elnr_d_pl_ms, 1, grp_d_pl_ms + 100, div, xh_ms_values,
1087             zg_ms_values)
1088     )
1089 #endregion
1090
1091 #region N) Coupling elements pylons
1092 elnr_c_pyl = 3400
1093
1094 def print_lines_c_pyl(start_num, sign_x, sign_y):
1095     output = "" # Initialize an empty string to store the appended lines
1096
1097     elnr = start_num
1098     output += f"\nSLN_{elnr}_{GRP}_{8}_{STYP}_{'N'}_{SNO}_{5}_{FIXA}_{'MYMZ'}_{FIXE}_{'MYMZ'}_{TITL}_{"
1099         Line\"
1100
1101     x1 = sign_x * abs(x2_g_ss)
1102     y1 = sign_y * abs(y_g_ss)
1103     z1 = z2_g_ss
1104
1105     x2 = sign_x * abs(x2_g_ss)
1106     y2 = sign_y * abs(y_g_ss)
1107     z2 = 5.840
1108
1109     output += f"\nSLNB_{X1}_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_{X2}_{round(x2,3)}_{round(y2,3)}_{round(z2,3)}"
1110     plt_model.append([(x1,y1,z1),(x2,y2,z2)])
1111
1112     output += f"\nSLN_{elnr+1}_{GRP}_{8}_{STYP}_{'N'}_{SNO}_{10}_{TITL}_{Line\"
1113
1114     x1 = sign_x * abs(x2_g_ss)
1115     y1 = sign_y * (abs(np.tan(alpha) * (5.84-0.85)) + 2.07)
1116     z1 = 5.840

```

```

1114
1115     x2 = sign_x * abs(x2_g_ss)
1116     y2 = sign_y * abs(y_g_ss)
1117     z2 = 5.840
1118
1119     output += f"\nSLNB_{X1_{round(x1,3)}}_{round(y1,3)}_{round(z1,3)}_{X2_{round(x2,
1120         3)}}_{round(y2,3)}_{round(z2,3)}"
1121     plt_model.append([(x1,y1,z1),(x2,y2,z2)])
1122
1123     output += f"\nSLN_{elnr+2}_GRP_8_STYP_'N'_SNO_5_FIXA_'MYMZ'_FIXE_'MYMZ'_TITL_
1124         \"Line\""
1125
1126     x1 = sign_x * abs(x2_g_ss)
1127     y1 = sign_y * abs(y_g_ss)
1128     z1 = z2_g_ss
1129
1130     x2 = sign_x * abs(x2_g_ss)
1131     y2 = sign_y * (abs(np.tan(alpha) * (5.84-0.85)) + 2.07)
1132     z2 = 5.840
1133
1134     output += f"\nSLNB_{X1_{round(x1,3)}}_{round(y1,3)}_{round(z1,3)}_{X2_{round(x2,
1135         3)}}_{round(y2,3)}_{round(z2,3)}"
1136     plt_model.append([(x1,y1,z1),(x2,y2,z2)])
1137
1138     return output # Return the appended lines
1139
1140 # Collect results from multiple calls
1141 result_c_pyl = (
1142     print_lines_c_pyl(elnr_c_pyl, -1, -1) + # Negative xh_ms, negative y_g_ms
1143     print_lines_c_pyl(elnr_c_pyl + 10, -1, 1) + # Negative xh_ms, positive y_g_ms
1144     print_lines_c_pyl(elnr_c_pyl + 20, 1, -1) + # Positive xh_ms, negative y_g_ms
1145     print_lines_c_pyl(elnr_c_pyl + 30, 1, 1) # Positive xh_ms, positive y_g_ms
1146 )
1147
1148 #endregion
1149
1150 #region 0) Cable pylons
1151 elnr_cab_pyl = 3500
1152
1153 def print_lines_cab_pyl(start_num, sign_x):
1154     elnr = start_num
1155
1156     output = "" # Initialize an empty string to store the appended lines
1157     output += f"\nSLN_{elnr}_GRP_61_STYP_'CZ'_SNO_13_TITL_\"Line\""
1158
1159     x1 = sign_x * abs(x2_g_ss)
1160     y1 = abs(y_g_ss)
1161     z1 = 5.840
1162
1163     x2 = sign_x * abs(x2_g_ss)
1164     y2 = - abs(y_g_ss)
1165     z2 = 5.840
1166
1167     output += f"\nSLNB_{X1_{round(x1,3)}}_{round(y1,3)}_{round(z1,3)}_{X2_{round(x2,
1168         3)}}_{round(y2,3)}_{round(z2,3)}"
1169     plt_model.append([(x1,y1,z1),(x2,y2,z2)])
1170
1171     return output
1172
1173 result_cab_pyl = (
1174     print_lines_cab_pyl(elnr_cab_pyl, -1) + # Negative xh_ms, negative y_g_ms
1175     print_lines_cab_pyl(elnr_cab_pyl + 1, 1) # Negative xh_ms, positive y_g_ms
1176 )
1177
1178 #endregion
1179
1180 #region P) Define rollers main cable
1181 pointnr_rol_cab = 100
1182
1183 def print_lines_rol_cab(pointnr, sign_x, sign_y):
1184     output = "" # Initialize an empty string to store the appended lines
1185
1186     x1 = sign_x * abs(x1_g_ss)
1187     y1 = sign_y * abs(vy)
1188     z1 = z1_g_ss

```

```

1189         output += f"\nSPT_{pointnr}_X_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_FIX_XP
1190             _TITL_"Point\""
1191         return output # Return the appended lines
1192
1193     result_rol_cab = (
1194         print_lines_rol_cab(pointnr_rol_cab, -1, -1) +
1195         print_lines_rol_cab(pointnr_rol_cab + 1, -1, 1) +
1196         print_lines_rol_cab(pointnr_rol_cab + 2, 1, -1) +
1197         print_lines_rol_cab(pointnr_rol_cab + 3, 1, 1)
1198     )
1199
1200     #endregion
1201
1202     #region Q) Define points main girder
1203     pointnr_con_mg = 200
1204
1205     def print_lines_con_mg(pointnr, sign_x, sign_y):
1206         output = "" # Initialize an empty string to store the appended lines
1207
1208         x1 = sign_x * abs(x1_g_ss)
1209         y1 = sign_y * abs(y_g_ss)
1210         z1 = z1_g_ss
1211
1212         output += f"\nSPT_{pointnr}_X_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_TITL_
1213             _\"Point\""
1214         # output += f"\nSPTP TYPE PPMM REF {pointnr - 100}"
1215         return output # Return the appended lines
1216
1217     result_con_mg = (
1218         print_lines_con_mg(pointnr_con_mg, -1, -1) +
1219         print_lines_con_mg(pointnr_con_mg + 1, -1, 1) +
1220         print_lines_con_mg(pointnr_con_mg + 2, 1, -1) +
1221         print_lines_con_mg(pointnr_con_mg + 3, 1, 1)
1222     )
1223
1224     #endregion
1225
1226     #region R) LOAD - SW DL and static mass!
1227
1228     def print_sw_and_dl(sar, sln):
1229         output = ""
1230         +PROG SOFILOAD
1231         HEAD EXPORT FROM DATABASE
1232         UNIT TYPE 5
1233         LC 1 'G' 1 FACD 1 TITL "Selfweight"
1234
1235         LC 2 'G' 1 TITL "Additional deadload"
1236         ""
1237         for i in range(len(sar)):
1238             output += f"\nAREA_{SAR}_{sar[i]}_TITL_'Epoxy_Finish'_WIDE_{0}TYPE_{PG}0.100000"
1239
1240         for j in range(len(sln)):
1241             output += f"\nLINE_{Sln}_{sln[j]}_TITL_'Edge+cable'_WIDE_{0}TYPE_{PG}0.4"
1242
1243         output += "\n_{LC}_{3}'G'_FACT_{1.0}TITL_'Static_load_pedestrians_{TC3}!"
1244
1245         for k in range(len(sar)):
1246             output += f"\nAREA_{SAR}_{sar[k]}_TITL_'Pedestrian_weight'_WIDE_{0}TYPE_{PG}0.35"
1247
1248         return output
1249
1250     sar = sar_ss + sar_ms
1251
1252     result_sw_and_dl = (
1253         print_sw_and_dl(sar, sln)
1254     )
1255
1256     #endregion
1257
1258     #region S) Fixed connection main girder
1259     pointnr_con_mg2 = 150
1260
1261     def print_lines_con_mg2(pointnr, sign_x, sign_y):
1262         output = "" # Initialize an empty string to store the appended lines
1263
1264         x1 = sign_x * abs(x1_g_ss)
1265         y1 = sign_y * abs(y_g_ss)
1266         z1 = z1_g_ss

```

```

1267
1268         output += f"\nSPT_{pointnr}_X_{round(x1,3)}_{round(y1,3)}_{round(z1,3)}_TITL_
1269         \"Point\""
1270
1271         output += f"\nSPTP_TYPE_PPMREF_{pointnr}_50"
1272
1273         return output # Return the appended lines
1274
1275 result_con_mg2 = (
1276     print_lines_con_mg2(pointnr_con_mg2, -1, -1) +
1277     print_lines_con_mg2(pointnr_con_mg2 + 1, -1, 1) +
1278     print_lines_con_mg2(pointnr_con_mg2 + 2, 1, -1) +
1279     print_lines_con_mg2(pointnr_con_mg2 + 3, 1, 1)
1280 )
1281
1282 #endregion
1283
1284 #region S2) Point moments pylon (eccentricity)
1285 pointnr_pyl = 2600
1286
1287 def print_points_pylon(pointnr):
1288     output = "" # Initialize an empty string to store the appended lines
1289
1290     for i in range(len(pnt_pyl)):
1291         output += f"\nSPT_{pointnr+i}_X_{pnt_pyl[i][0]}_{pnt_pyl[i][1]}_{pnt_pyl[i]
1292         ][2]}_TITL_\"Point\""
1293
1294     return output # Return the appended lines
1295
1296 result_points_pylon = (
1297     print_points_pylon(pointnr_pyl)
1298 )
1299 #endregion
1300
1301 #region T) Cross section and material generation
1302
1303 cs_generation = f"""
1304 +PROG AQUA urs:1
1305
1306 HEAD Voldijkbrug_definitief_V1
1307 UNIT 5
1308 NORM 'NEN' 'en199x-200x' COUN 31 CAT 'A2' UNIT 5
1309 STEE 1 S '355' TITL "=Structural Steel - S 355 (EN 199"
1310 SSLA EPS SERV SIG 1[-] TYPE EXT
1311 SSLA EPS ULTI SIG 1[-] TYPE EXT
1312 HMAT 1 TYPE FOUR NSP 0 A 1
1313 MATE 2 E 205000 MUE 0.300000 G 73077 GAM 78.50000 FY 460 FT 610 TITL "Steel - Macalloy
1314 460 Cables"
1315 CONC 3 C '45' TYPR B TITL "=Concrete Pylons - C45/55 (EN 19"
1316 SSLA EPS SERV SIG 1.500000[-] TYPE LIM
1317 SSLA EPS ULTI SIG 1.500000[-] TYPE LIM
1318 SSLA EPS CALC SIG -1.500000[-] TYPE LIM
1319 HMAT 3 TYPE FOUR NSP 0.030000 A 1
1320 CONC 4 C '30' TYPR B TITL "=Concrete Foundation - C 30/37 (E"
1321 SSLA EPS SERV SIG 1.500000[-] TYPE LIM
1322 SSLA EPS ULTI SIG 1.500000[-] TYPE LIM
1323 SSLA EPS CALC SIG -1.500000[-] TYPE LIM
1324 HMAT 4 TYPE FOUR NSP 0.030000 A 1
1325 CONC 5 C '30' GAM 0 TYPR B RHO 0 TITL "=Concrete Zero Weight - C 30/37 ("
1326 SSLA EPS SERV SIG 1.500000[-] TYPE LIM
1327 SSLA EPS ULTI SIG 1.500000[-] TYPE LIM
1328 SSLA EPS CALC SIG -1.500000[-] TYPE LIM
1329 HMAT 5 TYPE FOUR NSP 0.030000 A 1
1330 CONC 6 C '30N' TYPR B TITL "=Concrete Pylons - C 30/37 N (EN"
1331 SSLA EPS SERV SIG 1.500000[-] TYPE LIM
1332 SSLA EPS ULTI SIG 1.500000[-] TYPE LIM
1333 SSLA EPS CALC SIG -1.500000[-] TYPE LIM
1334 HMAT 6 TYPE FOUR NSP 0.030000 A 1
1335 STEE 7 B '500B' TMAX 32 TITL "=Reinforcement Steel - B 500 B (E"
1336 SSLA EPS SERV SIG 1.150000[-] TYPE EXT
1337 SSLA EPS ULTI SIG -1.150000[-] TYPE EXT
1338 SSLA EPS CALC SIG -1.150000[-] TYPE EXT
1339 HMAT 7 TYPE FOUR NSP 0 A 1
1340 STEE 8 S '355' TITL "=Structural Steel - S 355 (EN 199"
1341 SSLA EPS SERV SIG 1[-] TYPE EXT
1342 SSLA EPS ULTI SIG 1[-] TYPE EXT
1343 HMAT 8 TYPE FOUR NSP 0 A 1
1344 STEE 9 S '460Q' TITL "=Steel - Macalloy 460 Cables"

```

```

1344 SSLA EPS SERV SIG 1[-] TYPE EXT
1345 SSLA EPS ULTI SIG 1[-] TYPE EXT
1346 HMAT 9 TYPE FOUR NSP 0 A 1
1347 CONC 10 C '45N' TYPR B TITL "Concrete Piles - C45/55 N (EN 19"
1348 SSLA EPS SERV SIG 1.500000[-] TYPE LIM
1349 SSLA EPS ULTI SIG 1.500000[-] TYPE LIM
1350 SSLA EPS CALC SIG -1.500000[-] TYPE LIM
1351 HMAT 10 TYPE FOUR NSP 0.030000 A 1
1352 CONC 11 C '30' TYPR B TITL "Concrete Foundation - C 30/37 (E"
1353 SSLA EPS SERV SIG 1.500000[-] TYPE LIM
1354 SSLA EPS ULTI SIG 1.500000[-] TYPE LIM
1355 SSLA EPS CALC SIG -1.500000[-] TYPE LIM
1356 HMAT 11 TYPE FOUR NSP 0.030000 A 1
1357 CONC 12 C '30' GAM 0 TYPR B RHO 0 TITL "Concrete Zero Weight - C 30/37 ("
1358 SSLA EPS SERV SIG 1.500000[-] TYPE LIM
1359 SSLA EPS ULTI SIG 1.500000[-] TYPE LIM
1360 SSLA EPS CALC SIG -1.500000[-] TYPE LIM
1361 HMAT 12 TYPE FOUR NSP 0.030000 A 1
1362 CONC 13 C '30N' TYPR B TITL "Concrete Pylons - C 30/37 N (EN"
1363 SSLA EPS SERV SIG 1.500000[-] TYPE LIM
1364 SSLA EPS ULTI SIG 1.500000[-] TYPE LIM
1365 SSLA EPS CALC SIG -1.500000[-] TYPE LIM
1366 HMAT 13 TYPE FOUR NSP 0.030000 A 1
1367 STEE 14 B '500B' TMAX 32 TITL "Reinforcement Steel - B 500 B (E"
1368 SSLA EPS SERV SIG 1.150000[-] TYPE EXT
1369 SSLA EPS ULTI SIG -1.150000[-] TYPE EXT
1370 SSLA EPS CALC SIG -1.150000[-] TYPE EXT
1371 HMAT 14 TYPE FOUR NSP 0 A 1
1372 $ -----
1373 $ SMAT definition
1374 $ -----
1375 SMAT 1 LTYP STD MTYP PLAS ALPH 0 TITL "compression only"
1376 $ Reaction type -----
1377 $ Curve (P-level)
1378 SFLA U -1000[mm] F -1.000000E+06[kN] S POL TYPE P
1379 SFLA U 0[mm] F 0[kN] S POL
1380 SFLA U 1000[mm] F 0[kN] S POL
1381 $ -----
1382 $ SMAT definition
1383 $ -----
1384 SMAT 2 LTYP IHNG MTYP PLAS ALPH 0
1385 $ Reaction type -----
1386 $ Curve (P-level)
1387 SFLA U -1000[mm] F -1.000000E+09[kN] S POL TYPE N
1388 SFLA U 0[mm] F 0[kN] S POL
1389 SFLA U 1000[mm] F 0[kN] S POL
1390 CTRL
1391 CTRL RFCS 1
1392 CTRL FACE -1
1393 CTRL REFD 0
1394 CTRL STYP FEM
1395 CTRL SCUT 16
1396 CTRL PLAS 1
1397 SECT 1 MNO 8 MRF 0 FSYM NONE BTYP BEAM TITL "Girder - Right"
1398 SV IT 100[o/o]
1399
1400 $ Excentric girder results
1401 PLAT 'f11' YB -342.5000-{del_wf} ZB 220+{del_hw} YE 47.50000+{del_wf} ZE 220+{del_hw} T {
1402 tf} MNO 8
1403 PLAT 'web1' YB -52.50000-{del_wf} ZB -860-{del_hw} YE -342.5000-{del_wf} ZE 220+{del_hw}
1404 T {tw} MNO 8
1405 PLAT 'web2' YB 347.5000+{del_wf} ZB -860-{del_hw} YE 47.50000+{del_wf} ZE 220+{del_hw} T
1406 {tw} MNO 8
1407 PLAT 'f12' YB -52.50000-{del_wf} ZB -860-{del_hw} YE 347.5000+{del_wf} ZE -860-{del_hw} T
1408 {tf} MNO 8
1409
1410 CTRL
1411 CTRL RFCS 1
1412 CTRL FACE -1
1413 CTRL REFD 0
1414 CTRL STYP FEM
1415 CTRL SCUT 16
1416 CTRL PLAS 1
1417 SECT 2 MNO 8 MRF 0 FSYM NONE BTYP BEAM TITL "Girder - Left"
1418 SV IT 100[o/o]
1419
1420 $ Excentric girder results
1421 PLAT 'f11' YB 342.5000+{del_wf} ZB 220+{del_hw} YE -47.50000-{del_wf} ZE 220+{del_hw} T {
1422 tf} MNO 8

```

```

1418 PLAT 'web1' YB 52.50000+{del_wf} ZB -860-{del_hw} YE 342.5000+{del_wf} ZE 220+{del_hw} T
      {tw} MNO 8
1419 PLAT 'web2' YB -347.5000-{del_wf} ZB -860-{del_hw} YE -47.50000-{del_wf} ZE 220+{del_hw}
      T {tw} MNO 8
1420 PLAT 'f12' YB 52.50000+{del_wf} ZB -860-{del_hw} YE -347.5000-{del_wf} ZE -860-{del_hw} T
      {tf} MNO 8
1421
1422 CTRL
1423 CTRL RFCS 1
1424 CTRL FACE -1
1425 CTRL REFD 0
1426 CTRL STYP FEM
1427 CTRL SCUT 16
1428 CTRL PLAS 1
1429 SECT 3 MNO 8 MRF 0 FSYM NONE BTYP BEAM TITL "Trough - Type 1"
1430 SV IT 100[o/o]
1431 PLAT 'T3' YB -152.3896 ZB -117 YE -25 ZE 233 T 6 MNO 8
1432 PLAT '0800' YB -25 ZB 233 YE 25 ZE 233 T 6 MNO 8
1433 PLAT '0900' YB 25 ZB 233 YE 152.3896 ZE -117 T 6 MNO 8
1434 CTRL
1435 CTRL RFCS 1
1436 CTRL FACE -1
1437 CTRL REFD 0
1438 CTRL STYP FEM
1439 CTRL SCUT 16
1440 CTRL PLAS 1
1441 SECT 4 MNO 8 MRF 0 FSYM NONE BTYP BEAM TITL "Trough - Type 2"
1442 SV IT 100[o/o]
1443 PLAT 'T4' YB -149.1893 ZB -117 YE -100 ZE 233 T 12 MNO 8
1444 PLAT '0200' YB -100 ZB 233 YE 100 ZE 233 T 12 MNO 8
1445 PLAT '0300' YB 100 ZB 233 YE 149.1893 ZE -117 T 12 MNO 8
1446 CTRL
1447 CTRL PLAS 0
1448 SECT 5 MNO 8 MRF 0 BTYP BEAM BCY 'A' BCZ 'A' TITL "Compr. Sup. - CHS 101.6 x 4"
1449 PROF TYPE 'CHS' Z1 101.6000 Z3 4 MNO 8 ALPH 0 YM 0 ZM 0 DTYP S REF C
1450 CTRL
1451 CTRL RFCS 0
1452 SCIT 6 D 72 MNO 9 MRF 0 IT 100[o/o] AY 100[o/o] AZ 100[o/o] TITL "Primary Cable - D 72"
1453 CTRL
1454 CTRL RFCS 0
1455 SCIT 7 D 36 MNO 9 MRF 0 IT 100[o/o] AY 100[o/o] AZ 100[o/o] TITL "Secondary Cable - D 36"
1456
1457 SREC 8 H {wl}[mm] B {wl}[mm] SO 45[mm] SU 45[mm] MNO 3 MRF 7 ASO {A_rl}[cm2] ASU {A_rl}[
      cm2] RTYP cu DASO 20 DASU 20 TITL "Pylon - Part 1"
1458
1459 SREC 9 H {wu}[mm] B {wu}[mm] SO 45[mm] SU 45[mm] MNO 3 MRF 7 ASO {A_rl}[cm2] ASU {A_rl}[
      cm2] RTYP cu DASO 20 DASU 20 TITL "Pylon - Part 2"
1460
1461 CTRL
1462 CTRL PLAS 0
1463 SECT 10 MNO 8 MRF 0 BTYP BEAM BCY 'B' BCZ 'C' TITL "Cant. Sup. - HE 300 A"
1464 PROF TYPE 'HEA' Z1 300 MNO 8 ALPH 0 YM 0 ZM 0 DTYP S REF C
1465 CTRL
1466 CTRL RFCS 1
1467 CTRL FACE -1
1468 CTRL REFD 0
1469 CTRL STYP FEM
1470 CTRL SCUT 17
1471 CTRL PLAS 1
1472 SECT 11 MNO 11 MRF 14 FSYM NONE BTYP BEAM TITL "Foundation Beam Pylon"
1473 SV IT 100[o/o]
1474 LAY 1 TYPE MIN MRF 14
1475 LAY 2 TYPE MIN MRF 14
1476 LAY 3 TYPE MIN MRF 14
1477 LAY 4 TYPE MIN MRF 14
1478 POLY TYPE 0 MNO 11
1479 VERT '0100' Y 500 Z -850 EXP 1
1480 VERT '0101' Y 500 Z 850 EXP 1
1481 VERT '0102' Y -500 Z 850 EXP 1
1482 VERT '0103' Y -500 Z -850 EXP 1
1483 VERT '0100' Y 500 Z -850 EXP 1
1484 CUT 'ZS' ZB 'S' NS 0 MS 0 MNO 11 MRF 14 LAY 1 TYPE WEB VYFK 0 INCL 90
1485 CTRL
1486 CTRL RFCS 0
1487 SCIT 12 D 56 MNO 9 MRF 0 IT 100[o/o] AY 100[o/o] AZ 100[o/o] TITL "End Cable - D 56"
1488 CTRL
1489 CTRL RFCS 0
1490 SCIT 13 D 48 MNO 9 MRF 0 IT 100[o/o] AY 100[o/o] AZ 100[o/o] TITL "Pylon Cable - D 48"
1491 CTRL
1492 CTRL RFCS 1

```

```

1493 CTRL FACE -1
1494 CTRL REFD 0
1495 CTRL STYP FEM
1496 CTRL SCUT 17
1497 CTRL PLAS 1
1498 SECT 14 MNO 11 MRF 14 FSYM NONE BTYP BEAM TITL "Pier - Pylon"
1499 SV IT 100[o/o]
1500 LAY 1 TYPE MIN MRF 14
1501 LAY 2 TYPE MIN MRF 14
1502 LAY 3 TYPE MIN MRF 14
1503 LAY 4 TYPE MIN MRF 14
1504 POLY TYPE 0 MNO 11
1505 VERT '0100' Y 500 Z -300 EXP 1
1506 VERT '0101' Y 500 Z 300 EXP 1
1507 VERT '0102' Y -500 Z 300 EXP 1
1508 VERT '0103' Y -500 Z -300 EXP 1
1509 VERT '0100' Y 500 Z -300 EXP 1
1510 CUT 'ZS' ZB 'S' NS 0 MS 0 MNO 11 MRF 14 LAY 1 TYPE WEB VYFK 0 INCL 90
1511 CTRL
1512 CTRL RFCS 1
1513 CTRL FACE -1
1514 CTRL REFD 0
1515 CTRL STYP FEM
1516 CTRL SCUT 17
1517 CTRL PLAS 1
1518 SECT 15 MNO 11 MRF 14 FSYM NONE BTYP BEAM TITL "Pier - Abutment"
1519 SV IT 100[o/o]
1520 LAY 1 TYPE MIN MRF 14
1521 LAY 2 TYPE MIN MRF 14
1522 LAY 3 TYPE MIN MRF 14
1523 LAY 4 TYPE MIN MRF 14
1524 POLY TYPE 0 MNO 11
1525 VERT '0100' Y 625 Z -400 EXP 1
1526 VERT '0101' Y 625 Z 400 EXP 1
1527 VERT '0102' Y -625 Z 400 EXP 1
1528 VERT '0103' Y -625 Z -400 EXP 1
1529 VERT '0100' Y 625 Z -400 EXP 1
1530 CUT 'ZS' ZB 'S' NS 0 MS 0 MNO 11 MRF 14 LAY 1 TYPE WEB VYFK 0 INCL 90
1531 CTRL
1532 CTRL RFCS 1
1533 CTRL FACE -1
1534 CTRL REFD 0
1535 CTRL STYP FEM
1536 CTRL SCUT 17
1537 CTRL PLAS 1
1538 SECT 16 MNO 11 MRF 14 FSYM NONE BTYP BEAM TITL "Compression Element"
1539 SV IT 100[o/o]
1540 LAY 1 TYPE MIN MRF 14
1541 LAY 2 TYPE MIN MRF 14
1542 LAY 3 TYPE MIN MRF 14
1543 LAY 4 TYPE MIN MRF 14
1544 POLY TYPE 0 MNO 11
1545 VERT '0100' Y 200 Z -200 EXP 1
1546 VERT '0101' Y 200 Z 200 EXP 1
1547 VERT '0102' Y -200 Z 200 EXP 1
1548 VERT '0103' Y -200 Z -200 EXP 1
1549 VERT '0100' Y 200 Z -200 EXP 1
1550 CUT 'ZS' ZB 'S' NS 0 MS 0 MNO 11 MRF 14 LAY 1 TYPE WEB VYFK 0 INCL 90
1551 CTRL
1552 CTRL RFCS 1
1553 CTRL FACE -1
1554 CTRL REFD 0
1555 CTRL STYP FEM
1556 CTRL SCUT 17
1557 CTRL PLAS 1
1558 SECT 17 MNO 10 MRF 14 FSYM NONE BTYP BEAM TITL "Foundation Piles"
1559 SV IT 100[o/o]
1560 LAY 1 TYPE MIN MRF 14
1561 LAY 2 TYPE MIN MRF 14
1562 LAY 3 TYPE MIN MRF 14
1563 LAY 4 TYPE MIN MRF 14
1564 POLY TYPE 0 MNO 10
1565 VERT '0100' Y 200 Z -200 EXP 1
1566 VERT '0101' Y 200 Z 200 EXP 1
1567 VERT '0102' Y -200 Z 200 EXP 1
1568 VERT '0103' Y -200 Z -200 EXP 1
1569 VERT '0100' Y 200 Z -200 EXP 1
1570 CUT 'ZS' ZB 'S' NS 0 MS 0 MNO 10 MRF 14 LAY 1 TYPE WEB VYFK 0 INCL 90
1571 CTRL
1572 CTRL RFCS 1

```



```

1573 CTRL FACE -1
1574 CTRL REFD 0
1575 CTRL STYP FEM
1576 CTRL SCUT 17
1577 CTRL PLAS 1
1578 SECT 18 MNO 12 MRF 0 FSYM NONE BTYP BEAM TITL "Pier - Pylon (Coupler)"
1579 SV IT 100[o/o]
1580 POLY TYPE 0 MNO 12
1581 VERT '0100' Y 500 Z -300 EXP 1
1582 VERT '0101' Y 500 Z 300 EXP 1
1583 VERT '0102' Y -500 Z 300 EXP 1
1584 VERT '0103' Y -500 Z -300 EXP 1
1585 VERT '0100' Y 500 Z -300 EXP 1
1586 CUT 'ZS' ZB 'S' NS 0 MS 0 MNO 12 LAY 1 TYPE WEB VYFK 0 INCL 90
1587 CTRL
1588 CTRL RFCS 1
1589 CTRL FACE -1
1590 CTRL REFD 0
1591 CTRL STYP FEM
1592 CTRL SCUT 17
1593 CTRL PLAS 1
1594 SECT 19 MNO 12 MRF 0 FSYM NONE BTYP BEAM TITL "Pier - Abutment (Coupler)"
1595 SV IT 100[o/o]
1596 POLY TYPE 0 MNO 12
1597 VERT '0100' Y 500 Z -400 EXP 1
1598 VERT '0101' Y 500 Z 400 EXP 1
1599 VERT '0102' Y -500 Z 400 EXP 1
1600 VERT '0103' Y -500 Z -400 EXP 1
1601 VERT '0100' Y 500 Z -400 EXP 1
1602 CUT 'ZS' ZB 'S' NS 0 MS 0 MNO 12 LAY 1 TYPE WEB VYFK 0 INCL 90
1603 CTRL
1604 SREC 20 H {hc} B {bc} MNO 8 MRF 0 REF C IT 100[o/o] AY 100[o/o] AZ 100[o/o] BCYZ '0'
    SPT 0 TITL "Coupler Plate Hangers"
1605 CTRL
1606 SREC 21 H 30 B 380 MNO 8 MRF 0 REF C IT 100[o/o] AY 100[o/o] AZ 100[o/o] BCYZ '0' SPT
    0 TITL "Coupler Plate Hangers Horizontal"
1607 CTRL
1608 CTRL RFCS 1
1609 CTRL FACE -1
1610 CTRL REFD 0
1611 CTRL STYP FEM
1612 CTRL SCUT 17
1613 CTRL PLAS 1
1614 SECT 22 MNO 8 MRF 0 FSYM NONE BTYP BEAM TITL "Coupler Abutment"
1615 SV IT 100[o/o]
1616 POLY TYPE 0 MNO 8
1617 VERT '0100' Y -189.1276 Z -126.8099 EXP 1
1618 VERT '0101' Y 218.7109 Z 63.36833 EXP 1
1619 VERT '0102' Y 189.1276 Z 126.8099 EXP 1
1620 VERT '0103' Y -218.7109 Z -63.36833 EXP 1
1621 VERT '0100' Y -189.1276 Z -126.8099 EXP 1
1622 CTRL
1623 SREC 99 H 400 B 400 MNO 5 MRF 0 REF C IT 100[o/o] AY 100[o/o] AZ 100[o/o] BCYZ '0' SPT
    0 TITL "B/H = 400 / 400 mm"
1624
1625
1626 END
1627
1628 ""
1629
1630 #endregion
1631
1632
1633 #region U) Smoothen out CS
1634
1635 smoothen_cs =""
1636
1637 +prog aqua urs:7
1638 head
1639
1640 inte all
1641
1642 end
1643
1644 ""
1645
1646 #endregion
1647
1648
1649 #region V) Foundation code

```

```

1650
1651 def foundation(mesh):
1652
1653     foundation_code = f"""
1654
1655     +prog sofimshc
1656
1657     UNIT 5
1658     SYST SPAC GDIV 10000 GDIR NEGZ
1659     CTRL TOPO 0
1660     CTRL TOLG VAL 0.010000[m]
1661     CTRL DELN 0
1662     CTRL MESH 1
1663     CTRL EDRL 1
1664     CTRL HMIN VAL {mesh}[m]
1665
1666     SPT      21 X      51.5100      2.55500      4.24999 NX 0 0 -1 SX -1.83697E-16 -1 0 TITL "
1667         Point"
1668     SPT      25 X     -35.1100     -3.00000     -0.30000 FIX PY NX 0 0 -1 SX 1 0 0 TITL "Point"
1669     SPTS TYPE 'CZZ' CP 290000 AR 1
1670     SPT      26 X      33.9100     -3.00000     -0.30000 FIX PY NX 0 0 -1 SX 1 0 0 TITL "Point"
1671     SPTS TYPE 'CZZ' CP 290000 AR 1
1672     SPT      33 X     -35.1100      0.00000     -0.30000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1673     SPTS TYPE 'CZZ' CP 290000 AR 1
1674     SPT      34 X     -33.9100      0.00000     -0.30000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1675     SPTS TYPE 'CZZ' CP 290000 AR 1
1676     SPT      35 X     -33.9100      3.00000     -0.30000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1677     SPTS TYPE 'CZZ' CP 290000 AR 1
1678     SPT      36 X     -35.1100      3.00000     -0.30000 FIX PX NX 0 0 -1 SX 1 0 0 TITL "Point"
1679     SPTS TYPE 'CZZ' CP 290000 AR 1
1680     SPT      37 X     -33.9100     -3.00000     -0.30000 FIX PY NX 0 0 -1 SX 1 0 0 TITL "Point"
1681     SPTS TYPE 'CZZ' CP 290000 AR 1
1682     SPT      46 X      4.93000      0.000319      6.78000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1683     SPT      52 X     -52.2599      1.98995      3.45000 FIX PX NX 0 0 -1 SX 1 0 0 TITL "Point"
1684     SPTS TYPE 'CZZ' CP 178000 AR 1
1685     SPT      53 X     -50.7600      2.99000      3.45000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1686     SPTS TYPE 'CZZ' CP 178000 AR 1
1687     SPT      54 X     -50.7599      1.99000      3.45000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1688     SPTS TYPE 'CZZ' CP 178000 AR 1
1689     SPT      55 X      52.2599      1.98995      3.45000 FIX PX NX 0 0 -1 SX -1 0 0 TITL "Point"
1690     SPTS TYPE 'CZZ' CP 178000 AR 1
1691     SPT      56 X     -50.7600      0.98988      3.45001 NX 0 0 -1 SX 1 0 0 TITL "Point"
1692     SPTS TYPE 'CZZ' CP 178000 AR 1
1693     SPT      58 X     -50.7600     -1.01004      3.45001 NX 0 0 -1 SX 1 0 0 TITL "Point"
1694     SPTS TYPE 'CZZ' CP 178000 AR 1
1695     SPT      59 X      50.7600      2.99000      3.45000 NX 0 0 -1 SX -1 0 0 TITL "Point"
1696     SPTS TYPE 'CZZ' CP 178000 AR 1
1697     SPT      60 X     -50.7600     -2.01004      3.45001 NX 0 0 -1 SX 1 0 0 TITL "Point"
1698     SPTS TYPE 'CZZ' CP 178000 AR 1
1699     SPT      61 X     -50.7600     -3.01000      3.45002 FIX PY NX 0 0 -1 SX 1 0 0 TITL "Point"
1700     SPTS TYPE 'CZZ' CP 178000 AR 1
1701     SPT      62 X      50.7599      1.99000      3.45000 NX 0 0 -1 SX -1 0 0 TITL "Point"
1702     SPTS TYPE 'CZZ' CP 178000 AR 1
1703     SPT      63 X      50.7600      0.98988      3.45001 NX 0 0 -1 SX -1 0 0 TITL "Point"
1704     SPTS TYPE 'CZZ' CP 178000 AR 1
1705     SPT      65 X      50.7600     -1.01004      3.45001 NX 0 0 -1 SX -1 0 0 TITL "Point"
1706     SPTS TYPE 'CZZ' CP 178000 AR 1
1707     SPT      66 X      50.7600     -2.01004      3.45001 NX 0 0 -1 SX -1 0 0 TITL "Point"
1708     SPTS TYPE 'CZZ' CP 178000 AR 1
1709     SPT      67 X      50.7600     -3.01000      3.45002 FIX PY NX 0 0 -1 SX -1 0 0 TITL "Point"
1710     SPTS TYPE 'CZZ' CP 178000 AR 1
1711     SPT      72 X      33.9100      0.00000     -0.30000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1712     SPTS TYPE 'CZZ' CP 290000 AR 1
1713     SPT      73 X      35.1100      0.00000     -0.30000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1714     SPTS TYPE 'CZZ' CP 290000 AR 1
1715     SPT      74 X      35.1100      3.00000     -0.30000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1716     SPTS TYPE 'CZZ' CP 290000 AR 1
1717     SPT      75 X      33.9100      3.00000     -0.30000 FIX PX NX 0 0 -1 SX 1 0 0 TITL "Point"
1718     SPTS TYPE 'CZZ' CP 290000 AR 1
1719     SPT      76 X      35.1100     -3.00000     -0.30000 FIX PY NX 0 0 -1 SX 1 0 0 TITL "Point"
1720     SPTS TYPE 'CZZ' CP 290000 AR 1
1721     SPT      77 X     -52.2599     -0.01005      3.45000 NX 0 0 -1 SX 1 0 0 TITL "Point"
1722     SPTS TYPE 'CZZ' CP 178000 AR 1
1723     SPT      78 X     -52.2599     -2.01001      3.45000 FIX PX NX 0 0 -1 SX 1 0 0 TITL "Point"
1724     SPTS TYPE 'CZZ' CP 178000 AR 1
1725     SPT      79 X      52.2599     -0.01005      3.45000 NX 0 0 -1 SX -1 0 0 TITL "Point"
1726     SPTS TYPE 'CZZ' CP 178000 AR 1
1727     SPT      80 X      52.2599     -2.01001      3.45000 FIX PX NX 0 0 -1 SX -1 0 0 TITL "Point"
1728     SPTS TYPE 'CZZ' CP 178000 AR 1

```

```

1729      SLN          1 GRP 2 STYP 'N' SNO 16 DRX 1 3.63269E-12 -9.70173E-05 FIXE 'MYMZ' TITL "
1730      Line"
1731      SLNB X1      51.5098778 -2.55499292 4.24993531 X2 51.5100000 -2.55499292
1732      5.51000000
1733      SLN          2 GRP 2 STYP 'N' SNO 19 DRX -0.729563 -2.51634E-05 -0.683914 TITL "Line"
1734      SLNB X1     -50.7600039 -3.00999955 3.45001569 X2 -51.5098677 -3.00999955
1735      4.24993000
1736      SLN          3 GRP 2 STYP 'N' SNO 19 DRX 0.729563 -2.51634E-05 -0.683914 TITL "Line"
1737      SLNB X1      50.7600039 -3.00999955 3.45001569 X2 51.5098677 -3.00999955
1738      4.24993000
1739      SLN          4 GRP 2 STYP 'N' SNO 16 DRX 0.994643 0.103371 -7.65231E-06 FIXE 'MYMZ' TITL
1740      "Line"
1741      SLNB X1      51.5099904 2.55499900 4.24999493 X2 51.5100000 2.55500000
1742      5.50999872
1743
1744      SLN          17 GRP 2 STYP 'N' SNO 15 DRX 3.55255E-19 5.32907E-15 -1 TITL "Line"
1745      SLNB X1     -52.2599333 2.99000000 3.45000000 X2 -52.2599333 1.98995000
1746      3.45000000
1747      SLN          18 GRP 2 STYP 'N' SNO 15 DRX 3.55255E-19 5.32907E-15 -1 TITL "Line"
1748      SLNB X1     -52.2599333 1.98995000 3.45000000 X2 -52.2599333 0.99000000
1749      3.45000000
1750      SLN          19 GRP 2 STYP 'N' SNO 15 DRX 3.55255E-19 5.32907E-15 -1 TITL "Line"
1751      SLNB X1     -52.2599333 0.99000000 3.45000000 X2 -52.2599333 -0.01004955
1752      3.45000000
1753      SLN          20 GRP 2 STYP 'N' SNO 15 DRX -6.14988E-11 0 -1 TITL "Line"
1754      SLNB X1     -52.2599333 -0.01004955 3.45000000 X2 -52.2599333 -1.00998438
1755      3.45000000
1756      SLN          21 GRP 2 STYP 'N' SNO 15 DRX 9.84329E-10 -4.44076E-16 -1 TITL "Line"
1757      SLNB X1     -52.2599333 -1.00998438 3.45000000 X2 -52.2599333 -2.01001330
1758      3.45000000
1759      SLN          22 GRP 2 STYP 'N' SNO 15 DRX 9.84359E-10 -7.10224E-15 -1 TITL "Line"
1760      SLNB X1     -52.2599333 -2.01001330 3.45000000 X2 -52.2599333 -3.00999956
1761      3.45000000
1762      SLN          23 GRP 2 STYP 'N' SNO 15 DRX -3.07609E-11 -2.61475E-06 -1 TITL "Line"
1763      SLNB X1     -50.7600000 2.99000000 3.45000000 X2 -50.7599451 1.98999726
1764      3.45000261
1765      SLN          24 GRP 2 STYP 'N' SNO 19 DRX 0 0 -1 TITL "Line"
1766      SLNB X1     -50.7600000 2.99000000 3.45000000 X2 -52.2599333 2.99000000
1767      3.45000000
1768      SLN          25 GRP 2 STYP 'N' SNO 19 DRX 1.74318E-06 5.49193E-11 -1 TITL "Line"
1769      SLNB X1     -50.7599451 1.98999726 3.45000261 X2 -52.2599333 1.98995000
1770      3.45000000
1771      SLN          26 GRP 2 STYP 'N' SNO 19 DRX 3.48660E-06 -2.86251E-10 -1 TITL "Line"
1772      SLNB X1     -50.7599569 0.98987685 3.45000523 X2 -52.2599333 0.99000000
1773      3.45000000
1774      SLN          27 GRP 2 STYP 'N' SNO 19 DRX 5.22961E-06 1.74302E-10 -1 TITL "Line"
1775      SLNB X1     -50.7599686 -0.009999553 3.45000784 X2 -52.2599333 -0.01004955
1776      3.45000000
1777      SLN          28 GRP 2 STYP 'N' SNO 19 DRX 6.97294E-06 -7.82388E-10 -1 TITL "Line"
1778      SLNB X1     -50.7599804 -1.01003866 3.45001046 X2 -52.2599333 -1.00998438
1779      3.45000000
1780      SLN          29 GRP 2 STYP 'N' SNO 19 DRX 8.71624E-06 1.30186E-10 -1 TITL "Line"
1781      SLNB X1     -50.7599922 -2.01004284 3.45001307 X2 -52.2599333 -2.01001330
1782      3.45000000
1783      SLN          30 GRP 2 STYP 'N' SNO 19 DRX 1.04595E-05 3.09897E-14 -1 TITL "Line"
1784      SLNB X1     -50.7600039 -3.00999955 3.45001569 X2 -52.2599333 -3.00999956
1785      3.45000000
1786      SLN          31 GRP 2 STYP 'N' SNO 15 DRX 2.56281E-10 -2.61475E-06 -1 TITL "Line"
1787      SLNB X1     -50.7599451 1.98999726 3.45000261 X2 -50.7599569 0.98987685
1788      3.45000523
1789      SLN          32 GRP 2 STYP 'N' SNO 15 DRX 8.30525E-10 -2.61475E-06 -1 TITL "Line"
1790      SLNB X1     -50.7599569 0.98987685 3.45000523 X2 -50.7599686 -0.009999553
1791      3.45000784
1792      SLN          33 GRP 2 STYP 'N' SNO 15 DRX 1.33274E-10 -2.61475E-06 -1 TITL "Line"
1793      SLNB X1     -50.7599686 -0.009999553 3.45000784 X2 -50.7599804 -1.01003866
1794      3.45001046
1795      SLN          34 GRP 2 STYP 'N' SNO 15 DRX -9.22752E-11 -2.61475E-06 -1 TITL "Line"
1796      SLNB X1     -50.7599804 -1.01003866 3.45001046 X2 -50.7599922 -2.01004284
1797      3.45001307
1798      SLN          35 GRP 2 STYP 'N' SNO 15 DRX -3.07610E-11 -2.61475E-06 -1 TITL "Line"
1799      SLNB X1     -50.7599922 -2.01004284 3.45001307 X2 -50.7600039 -3.00999955
1800      3.45001569
1801      SLN          36 GRP 2 STYP 'N' SNO 19 DRX 0.729568 -1.79758E-11 -0.683909 TITL "Line"
1802      SLNB X1     -52.2599333 2.99000000 3.45000000 X2 -51.5100000 2.99000000
1803      4.25000000
1804      SLN          37 GRP 2 STYP 'N' SNO 19 DRX 0.729553 -4.30007E-10 -0.683925 TITL "Line"
1805      SLNB X1     -52.2599333 1.98995000 3.45000000 X2 -51.5099780 1.98992896
1806      4.24998833

```

```

1782     SLN      38 GRP 2 STYP 'N' SNO 19 DRX 0.729538 -2.10957E-11 -0.683941 TITL "Line"
1783     SLNB X1   -52.2599333 0.990000000 3.45000000 X2 -51.5099559 0.98995000
1784         4.24997667
1784     SLN      39 GRP 2 STYP 'N' SNO 19 DRX 0.729523 -3.95296E-10 -0.683957 TITL "Line"
1785     SLNB X1   -52.2599333 -0.01004955 3.45000000 X2 -51.5099339 -0.01002938
1786         4.24996500
1786     SLN      40 GRP 2 STYP 'N' SNO 19 DRX 0.729508 1.17355E-09 -0.683973 TITL "Line"
1787     SLNB X1   -52.2599333 -1.00998438 3.45000000 X2 -51.5099118 -1.01001795
1788         4.24995334
1788     SLN      41 GRP 2 STYP 'N' SNO 19 DRX 0.729492 1.54991E-09 -0.683989 TITL "Line"
1789     SLNB X1   -52.2599333 -2.01001330 3.45000000 X2 -51.5098898 -2.01004274
1790         4.24994167
1790     SLN      42 GRP 2 STYP 'N' SNO 19 DRX 0.729477 1.90275E-09 -0.684005 TITL "Line"
1791     SLNB X1   -52.2599333 -3.00999956 3.45000000 X2 -51.5098677 -3.00999955
1792         4.24993000
1792     SLN      43 GRP 2 STYP 'N' SNO 15 DRX -3.55255E-19 5.32907E-15 -1 TITL "Line"
1793     SLNB X1   52.2599333 2.990000000 3.45000000 X2 52.2599333 1.98995000
1794         3.45000000
1794     SLN      44 GRP 2 STYP 'N' SNO 19 DRX -0.729553 -2.88858E-05 -0.683924 TITL "Line"
1795     SLNB X1   -50.7599922 -2.01004284 3.45001307 X2 -51.5098898 -2.01004274
1796         4.24994167
1796     SLN      45 GRP 2 STYP 'N' SNO 19 DRX -0.729544 -2.88889E-05 -0.683934 TITL "Line"
1797     SLNB X1   -50.7599804 -1.01003866 3.45001046 X2 -51.5099118 -1.01001795
1798         4.24995334
1798     SLN      46 GRP 2 STYP 'N' SNO 19 DRX -0.729535 -2.88853E-05 -0.683944 TITL "Line"
1799     SLNB X1   -50.7599686 -0.009999553 3.45000784 X2 -51.5099339 -0.01002938
1800         4.24996500
1800     SLN      47 GRP 2 STYP 'N' SNO 19 DRX -0.729525 -2.88920E-05 -0.683954 TITL "Line"
1801     SLNB X1   -50.7599569 0.98987685 3.45000523 X2 -51.5099559 0.98995000
1802         4.24997667
1802     SLN      48 GRP 2 STYP 'N' SNO 19 DRX -0.729516 -2.88841E-05 -0.683964 TITL "Line"
1803     SLNB X1   -50.7599451 1.98999726 3.45000261 X2 -51.5099780 1.98992896
1804         4.24998833
1804     SLN      49 GRP 2 STYP 'N' SNO 19 DRX -0.729537 -2.88870E-05 -0.683941 TITL "Line"
1805     SLNB X1   -50.7600000 2.990000000 3.45000000 X2 -51.5100000 2.99000000
1806         4.25000000
1806     SLN      50 GRP 2 STYP 'N' SNO 15 DRX -3.46631E-10 1.16659E-05 -1 TITL "Line"
1807     SLNB X1   -51.5100000 2.990000000 4.25000000 X2 -51.5098677 -3.00999955
1808         4.24993000
1808     SLN      53 GRP 2 STYP 'N' SNO 16 DRX -1 8.16666E-09 -9.70148E-05 FIXA 'MYMZ' FIXE '
1809     MYMZ' TITL "Line"
1809     SLNB X1   -51.5098778 -2.55485504 4.24993531 X2 -51.5100000 -2.55500000
1810         5.51000000
1810     SLN      54 GRP 2 STYP 'N' SNO 16 DRX -1 1.23032E-10 -7.61128E-06 FIXA 'MYMZ' FIXE '
1811     MYMZ' TITL "Line"
1811     SLNB X1   -51.5099904 2.55500001 4.24999493 X2 -51.5100000 2.55500000
1812         5.51000000
1812
1813
1814
1815     SLN      77 GRP 2 STYP 'N' SNO 18 DRX -0.390616 0.605455 -0.693429 TITL "Line"
1816     SLNB X1   -33.9100000 -3.00000000 -0.30000000 X2 -34.5100000 -2.07000000
1817         0.85000000
1817     SLN      78 GRP 2 STYP 'N' SNO 14 DRX -1.45680E-30 -2.96059E-16 -1 TITL "Line"
1818     SLNB X1   -35.1100000 3.00000000 -0.30000000 X2 -35.1100000 0.0
1819         -0.30000000
1819     SLN      79 GRP 2 STYP 'N' SNO 14 DRX -3.88095E-30 1.77636E-15 -1 TITL "Line"
1820     SLNB X1   -35.1100000 0.0 -0.30000000 X2 -35.1100000 -3.00000000
1821         -0.30000000
1821     SLN      80 GRP 2 STYP 'N' SNO 14 DRX 3.80697E-31 2.07242E-15 -1 TITL "Line"
1822     SLNB X1   -33.9100000 3.00000000 -0.30000000 X2 -33.9100000 0.0
1823         -0.30000000
1823     SLN      81 GRP 2 STYP 'N' SNO 14 DRX 1.29365E-30 -5.92119E-16 -1 TITL "Line"
1824     SLNB X1   -33.9100000 0.0 -0.30000000 X2 -33.9100000 -3.00000000
1825         -0.30000000
1825     SLN      82 GRP 2 STYP 'N' SNO 18 DRX 0.390616 -0.605455 -0.693429 TITL "Line"
1826     SLNB X1   -35.1100000 3.00000000 -0.30000000 X2 -34.5100000 2.07000000
1827         0.85000000
1827     SLN      83 GRP 2 STYP 'N' SNO 18 DRX 0.886585 -1.13183E-14 -0.462566 TITL "Line"
1828     SLNB X1   -35.1100000 0.0 -0.30000000 X2 -34.5100000 0.0
1829         0.85000000
1829     SLN      84 GRP 2 STYP 'N' SNO 18 DRX 0.390616 0.605455 -0.693429 TITL "Line"
1830     SLNB X1   -35.1100000 -3.00000000 -0.30000000 X2 -34.5100000 -2.07000000
1831         0.85000000
1831     SLN      85 GRP 2 STYP 'N' SNO 11 DRX 1.57638E-31 8.58143E-16 -1 TITL "Line"
1832     SLNB X1   -34.5100000 2.07000000 0.85000000 X2 -34.5100000 -2.07000000
1833         0.85000000
1833     SLN      86 GRP 2 STYP 'N' SNO 18 DRX -0.886585 5.41250E-15 -0.462566 TITL "Line"
1834     SLNB X1   -33.9100000 0.0 -0.30000000 X2 -34.5100000 0.0
1835         0.85000000

```

```

1835     SLN      87 GRP 2 STYP 'N' SNO 18 DRX -0.390616 -0.605455 -0.693429 TITL "Line"
1836     SLNB X1 -33.9100000 3.00000000 -0.30000000 X2 -34.5100000 2.07000000
1837         0.85000000
1838
1839
1840     SLN     313 GRP 2 STYP 'N' SNO 18 DRX 3.05311E-15 -1.52493E-29 -1 TITL "Line"
1841     SLNB X1 -35.1100000 -3.00000000 -0.30000000 X2 -33.9100000 -3.00000000
1842         -0.30000000
1843     SLN     314 GRP 2 STYP 'N' SNO 18 DRX 3.05311E-15 -1.52493E-29 -1 TITL "Line"
1844     SLNB X1 -35.1100000 0.0 -0.30000000 X2 -33.9100000 0.0
1845         -0.30000000
1846     SLN     315 GRP 2 STYP 'N' SNO 18 DRX 3.05311E-15 -1.52493E-29 -1 TITL "Line"
1847     SLNB X1 -35.1100000 3.00000000 -0.30000000 X2 -33.9100000 3.00000000
1848         -0.30000000
1849     SLN     316 GRP 2 STYP 'N' SNO 19 DRX 0 0 -1 TITL "Line"
1850     SLNB X1 50.7600000 2.99000000 3.45000000 X2 52.2599333 2.99000000
1851         3.45000000
1852     SLN     317 GRP 2 STYP 'N' SNO 14 DRX -1.26111E-15 -2.96059E-16 -1 TITL "Line"
1853     SLNB X1 35.1100000 3.00000000 -0.30000000 X2 35.1100000 0.0
1854         -0.30000000
1855     SLN     318 GRP 2 STYP 'N' SNO 14 DRX -1.26111E-15 1.77636E-15 -1 TITL "Line"
1856     SLNB X1 35.1100000 0.0 -0.30000000 X2 35.1100000 -3.00000000
1857         -0.30000000
1858     SLN     319 GRP 2 STYP 'N' SNO 14 DRX -1.26111E-15 2.07242E-15 -1 TITL "Line"
1859     SLNB X1 33.9100000 3.00000000 -0.30000000 X2 33.9100000 0.0
1860         -0.30000000
1861     SLN     320 GRP 2 STYP 'N' SNO 14 DRX -1.26111E-15 -5.92119E-16 -1 TITL "Line"
1862     SLNB X1 33.9100000 0.0 -0.30000000 X2 33.9100000 -3.00000000
1863         -0.30000000
1864     SLN     321 GRP 2 STYP 'N' SNO 18 DRX -0.390616 -0.605455 -0.693429 TITL "Line"
1865     SLNB X1 35.1100000 3.00000000 -0.30000000 X2 34.5100000 2.07000000
1866         0.85000000
1867     SLN     322 GRP 2 STYP 'N' SNO 18 DRX -0.886585 -4.35056E-14 -0.462566 TITL "Line"
1868     SLNB X1 35.1100000 0.0 -0.30000000 X2 34.5100000 0.0
1869         0.85000000
1870     SLN     323 GRP 2 STYP 'N' SNO 18 DRX -0.390616 0.605455 -0.693429 TITL "Line"
1871     SLNB X1 35.1100000 -3.00000000 -0.30000000 X2 34.5100000 -2.07000000
1872         0.85000000
1873     SLN     324 GRP 2 STYP 'N' SNO 11 DRX -1.26111E-15 8.58143E-16 -1 TITL "Line"
1874     SLNB X1 34.5100000 2.07000000 0.85000000 X2 34.5100000 -2.07000000
1875         0.85000000
1876     SLN     327 GRP 2 STYP 'N' SNO 19 DRX -1.74318E-06 5.49193E-11 -1 TITL "Line"
1877     SLNB X1 50.7599451 1.98999726 3.45000261 X2 52.2599333 1.98995000
1878         3.45000000
1879     SLN     328 GRP 2 STYP 'N' SNO 19 DRX -3.48660E-06 -2.86251E-10 -1 TITL "Line"
1880     SLNB X1 50.7599569 0.98987685 3.45000523 X2 52.2599333 0.99000000
1881         3.45000000
1882     SLN     329 GRP 2 STYP 'N' SNO 19 DRX -5.22961E-06 1.74302E-10 -1 TITL "Line"
1883     SLNB X1 50.7599686 -0.009999553 3.45000784 X2 52.2599333 -0.01004955
1884         3.45000000
1885     SLN     330 GRP 2 STYP 'N' SNO 19 DRX -6.97294E-06 -7.82388E-10 -1 TITL "Line"
1886     SLNB X1 50.7599804 -1.01003866 3.45001046 X2 52.2599333 -1.00998438
1887         3.45000000
1888     SLN     331 GRP 2 STYP 'N' SNO 19 DRX -8.71624E-06 1.30186E-10 -1 TITL "Line"
1889     SLNB X1 50.7599922 -2.01004284 3.45001307 X2 52.2599333 -2.01001330
1890         3.45000000
1891     SLN     332 GRP 2 STYP 'N' SNO 19 DRX -1.04595E-05 3.09897E-14 -1 TITL "Line"
1892     SLNB X1 50.7600039 -3.00999955 3.45001569 X2 52.2599333 -3.00999956
1893         3.45000000
1894     SLN     333 GRP 2 STYP 'N' SNO 15 DRX -3.55255E-19 5.32907E-15 -1 TITL "Line"
1895     SLNB X1 52.2599333 1.98995000 3.45000000 X2 52.2599333 0.99000000
1896         3.45000000
1897     SLN     334 GRP 2 STYP 'N' SNO 15 DRX -3.55255E-19 5.32907E-15 -1 TITL "Line"
1898     SLNB X1 52.2599333 0.99000000 3.45000000 X2 52.2599333 -0.01004955
1899         3.45000000
1900     SLN     335 GRP 2 STYP 'N' SNO 18 DRX 0.390616 0.605455 -0.693429 TITL "Line"
1901     SLNB X1 33.9100000 -3.00000000 -0.30000000 X2 34.5100000 -2.07000000
1902         0.85000000
1903     SLN     336 GRP 2 STYP 'N' SNO 18 DRX 0.886585 3.75998E-14 -0.462566 TITL "Line"
1904     SLNB X1 33.9100000 0.0 -0.30000000 X2 34.5100000 0.0
1905         0.85000000
1906     SLN     337 GRP 2 STYP 'N' SNO 18 DRX 0.390616 -0.605455 -0.693429 TITL "Line"
1907     SLNB X1 33.9100000 3.00000000 -0.30000000 X2 34.5100000 2.07000000
1908         0.85000000
1909     SLN     338 GRP 2 STYP 'N' SNO 15 DRX 6.14988E-11 0 -1 TITL "Line"
1910     SLNB X1 52.2599333 -0.01004955 3.45000000 X2 52.2599333 -1.00998438
1911         3.45000000
1912     SLN     339 GRP 2 STYP 'N' SNO 15 DRX -9.84329E-10 -4.44076E-16 -1 TITL "Line"

```

```

1889      SLNB X1      52.2599333 -1.00998438  3.45000000 X2      52.2599333 -2.01001330
1890      3.45000000
1890      SLN      340 GRP 2 STYP 'N' SNO 15 DRX -9.84359E-10 -7.10224E-15 -1 TITL "Line"
1891      SLNB X1      52.2599333 -2.01001330  3.45000000 X2      52.2599333 -3.00999956
1891      3.45000000
1892      SLN      341 GRP 2 STYP 'N' SNO 15 DRX  3.07609E-11 -2.61475E-06 -1 TITL "Line"
1893      SLNB X1      50.7600000  2.99000000  3.45000000 X2      50.7599451  1.98999726
1893      3.45000261
1894      SLN      342 GRP 2 STYP 'N' SNO 15 DRX -2.56281E-10 -2.61475E-06 -1 TITL "Line"
1895      SLNB X1      50.7599451  1.98999726  3.45000261 X2      50.7599569  0.98987685
1895      3.45000523
1896      SLN      343 GRP 2 STYP 'N' SNO 15 DRX -8.30525E-10 -2.61475E-06 -1 TITL "Line"
1897      SLNB X1      50.7599569  0.98987685  3.45000523 X2      50.7599686 -0.009999553
1897      3.45000784
1898      SLN      344 GRP 2 STYP 'N' SNO 15 DRX -1.33274E-10 -2.61475E-06 -1 TITL "Line"
1899      SLNB X1      50.7599686 -0.009999553  3.45000784 X2      50.7599804 -1.01003866
1899      3.45001046
1900      SLN      345 GRP 2 STYP 'N' SNO 15 DRX  9.22752E-11 -2.61475E-06 -1 TITL "Line"
1901      SLNB X1      50.7599804 -1.01003866  3.45001046 X2      50.7599922 -2.01004284
1901      3.45001307
1902      SLN      346 GRP 2 STYP 'N' SNO 15 DRX  3.07610E-11 -2.61475E-06 -1 TITL "Line"
1903      SLNB X1      50.7599922 -2.01004284  3.45001307 X2      50.7600039 -3.00999955
1903      3.45001569
1904      SLN      347 GRP 2 STYP 'N' SNO 19 DRX -0.729568 -1.79758E-11 -0.683909 TITL "Line"
1905      SLNB X1      52.2599333  2.99000000  3.45000000 X2      51.5100000  2.99000000
1905      4.25000000
1906      SLN      348 GRP 2 STYP 'N' SNO 19 DRX -0.729553 -4.30007E-10 -0.683925 TITL "Line"
1907      SLNB X1      52.2599333  1.98995000  3.45000000 X2      51.5099780  1.98992896
1907      4.24998833
1908      SLN      349 GRP 2 STYP 'N' SNO 19 DRX -0.729538 -2.10957E-11 -0.683941 TITL "Line"
1909      SLNB X1      52.2599333  0.99000000  3.45000000 X2      51.5099559  0.98995000
1909      4.24997667
1910      SLN      350 GRP 2 STYP 'N' SNO 19 DRX -0.729523 -3.95296E-10 -0.683957 TITL "Line"
1911      SLNB X1      52.2599333 -0.01004955  3.45000000 X2      51.5099339 -0.01002938
1911      4.24996500
1912      SLN      351 GRP 2 STYP 'N' SNO 19 DRX -0.729508  1.17355E-09 -0.683973 TITL "Line"
1913      SLNB X1      52.2599333 -1.00998438  3.45000000 X2      51.5099118 -1.01001795
1913      4.24995334
1914      SLN      352 GRP 2 STYP 'N' SNO 19 DRX -0.729492  1.54991E-09 -0.683989 TITL "Line"
1915      SLNB X1      52.2599333 -2.01001330  3.45000000 X2      51.5098898 -2.01004274
1915      4.24994167
1916      SLN      353 GRP 2 STYP 'N' SNO 19 DRX -0.729477  1.90275E-09 -0.684005 TITL "Line"
1917      SLNB X1      52.2599333 -3.00999956  3.45000000 X2      51.5098677 -3.00999955
1917      4.24993000
1918      SLN      354 GRP 2 STYP 'N' SNO 19 DRX  0.729553 -2.88858E-05 -0.683924 TITL "Line"
1919      SLNB X1      50.7599922 -2.01004284  3.45001307 X2      51.5098898 -2.01004274
1919      4.24994167
1920      SLN      355 GRP 2 STYP 'N' SNO 19 DRX  0.729544 -2.88889E-05 -0.683934 TITL "Line"
1921      SLNB X1      50.7599804 -1.01003866  3.45001046 X2      51.5099118 -1.01001795
1921      4.24995334
1922      SLN      356 GRP 2 STYP 'N' SNO 19 DRX  0.729535 -2.88853E-05 -0.683944 TITL "Line"
1923      SLNB X1      50.7599686 -0.009999553  3.45000784 X2      51.5099339 -0.01002938
1923      4.24996500
1924      SLN      357 GRP 2 STYP 'N' SNO 19 DRX  0.729525 -2.88920E-05 -0.683954 TITL "Line"
1925      SLNB X1      50.7599569  0.98987685  3.45000523 X2      51.5099559  0.98995000
1925      4.24997667
1926      SLN      358 GRP 2 STYP 'N' SNO 19 DRX  0.729516 -2.88841E-05 -0.683964 TITL "Line"
1927      SLNB X1      50.7599451  1.98999726  3.45000261 X2      51.5099780  1.98992896
1927      4.24998833
1928      SLN      359 GRP 2 STYP 'N' SNO 19 DRX  0.729537 -2.88870E-05 -0.683941 TITL "Line"
1929      SLNB X1      50.7600000  2.99000000  3.45000000 X2      51.5100000  2.99000000
1929      4.25000000
1930      SLN      378 GRP 2 STYP 'N' SNO 15 DRX  3.46630E-10  1.16659E-05 -1 TITL "Line"
1931      SLNB X1      51.5100000  2.99000000  4.25000000 X2      51.5098677 -3.00999955
1931      4.24993000
1932
1933
1934      SLN      381 GRP 2 STYP 'N' SNO 18 DRX -4.31422E-15 -1.25860E-28 -1 TITL "Line"
1935      SLNB X1      35.1100000 -3.00000000 -0.30000000 X2      33.9100000 -3.00000000
1935      -0.30000000
1936      SLN      382 GRP 2 STYP 'N' SNO 18 DRX -4.31422E-15 -1.25860E-28 -1 TITL "Line"
1937      SLNB X1      35.1100000  0.0 -0.30000000 X2      33.9100000  0.0
1937      -0.30000000
1938      SLN      383 GRP 2 STYP 'N' SNO 18 DRX -4.31422E-15 -1.25860E-28 -1 TITL "Line"
1939      SLNB X1      35.1100000  3.00000000 -0.30000000 X2      33.9100000  3.00000000
1939      -0.30000000
1940
1941
1942      " " "
1943

```

```

1944         return foundation_code
1945
1946     #endregion
1947
1948
1949     #region Y) Apply cable stress
1950     grp_ss = np.arange(21, 21 + div_ss, 1)
1951     cab.extend(grp_ss)
1952
1953     # Group MS
1954     if con_ms == 1:
1955         grp_ms = np.arange(21 + div_ss, 21 + div_ss + div_ms, 1)
1956         cab.extend(grp_ms)
1957     else:
1958         grp_ms = np.arange(21 + div_ss, 21 + div_ss + div_ms_p, 1)
1959         cab.extend(grp_ms)
1960
1961
1962     # Append the final value
1963     if con_ms == 1:
1964         cab.append(30)
1965
1966     pointnr_con_mom = 250
1967
1968     def print_lines_con_mom(pointnr, sign_x, sign_y):
1969         output = "" # Initialize an empty string to store the appended lines
1970
1971         if con_ms == 1:
1972             output += f"\nNODE_{pointnr-100}_TYPE_MY_{-sign_x*_ (Hw_/2)*0.325}"
1973         elif con_ms == 0:
1974             output += f"\nNODE_{pointnr-100}_TYPE_MY_{-sign_x*_ (Hw_p/2)*0.325}"
1975
1976         return output
1977
1978     result_con_mom = (
1979         f"\n_{prog}_sofiload"+
1980         f"\nLC_{5}_Q'_{1}_TITL_\\"Moment\"" +
1981         print_lines_con_mom(pointnr_con_mom, -1, -1) +
1982         print_lines_con_mom(pointnr_con_mom + 1, -1, 1) +
1983         print_lines_con_mom(pointnr_con_mom + 2, 1, -1) +
1984         print_lines_con_mom(pointnr_con_mom + 3, 1, 1) +
1985         f"\nend"
1986     )
1987
1988
1989
1990
1991     # apply prestressing
1992     if con_ms == 1:
1993         H = Hw / 2
1994     else:
1995         H = Hw_p / 2
1996
1997
1998     start_plc = 20
1999     end_plc = 40
2000
2001     if hang_ms <= 2:
2002         apply_cable_stress = f""
2003         +prog ase
2004
2005         syst prob th3
2006
2007         grp -
2008
2009         ""
2010
2011         for j in range(len(cab)):
2012             apply_cable_stress += f"\ngrp_{cab[j]}_prex_{H}_hori'_fac_{1e-4}"
2013
2014             apply_cable_stress += f"\n_{nlc_{start_plc}_facd_{1.0}_nlcc_{5}_n_\nend"
2015
2016     elif hang_ms > 2:
2017
2018         apply_cable_stress = f""
2019         +prog ase
2020
2021         syst prob th3
2022
2023         grp -

```



```

2024     sto#f 1.05
2025
2026     """
2027
2028
2029     for j in range(len(cab)):
2030         apply_cable_stress += f"\ngrp_{cab[j]}_prex_{f*{H}_hori'_facs_1e-2"
2031
2032         apply_cable_stress += f"\n_{nlc_{start_plc}_facd_1.0_nlcc_5_n_nend"
2033
2034     elif hang_ms > 4:
2035
2036         apply_cable_stress = f""
2037         +prog ase
2038
2039         syst prob th3
2040
2041         grp -
2042
2043         sto#f 1.025
2044
2045         """
2046
2047         for j in range(len(cab)):
2048             apply_cable_stress += f"\ngrp_{cab[j]}_prex_{f*{H}_hori'_facs_1e-2"
2049
2050             apply_cable_stress += f"\n_{nlc_{start_plc}_facd_1.0_nlcc_5_n_nend"
2051
2052     adjust_cable_stress = f""
2053     +prog ase urs:6
2054
2055     syst prob th3 plc {start_plc}
2056
2057     grp - facs 1.0
2058
2059     lc {start_plc + 10} facd 1.0
2060     lcc 5
2061
2062     end
2063
2064     """
2065
2066
2067     #endregion
2068
2069
2070     #region AA) Static load cases
2071     q_wind = 0.5 * 1.25 * 24.5 ** 2 * 3.2 * ((hw / 1000) + 1.2) / 1000
2072     q_wind_pyl_1 = 0.5 * 1.25 * 24.5 ** 2 * 5.6 * (wl / 1000) / 1000
2073     q_wind_pyl_2 = 0.5 * 1.25 * 24.5 ** 2 * 5.6 * (wu / 1000) / 1000
2074
2075     def print_loadcases(sar_ss, sar_ms, sln_w_xx, sln_w_xx_p, sln_w_yy, sln_w_zz, sln_w_p,
2076                         pnt_pyl):
2077
2078         if con_ms == 1:
2079
2080             sln = sln_w_zz
2081
2082             output = f""
2083
2084             +PROG SOFILOAD
2085             HEAD EXPORT FROM DATABASE
2086             UNIT TYPE 5
2087
2088             LC 1 'G' 1 FACD 1 TITL "Selfweight"
2089
2090             LC 2 'G' 1 TITL "Additional deadload"
2091             """
2092
2093             for i in range(len(sar)):
2094
2095                 output += f"\nAREA_{sar[i]}_TITL'Epoxy_Finish'_WIDE_0_TYPE_PG_0.10"
2096
2097             for i in range(len(sln)):
2098                 output += f"\nLINE_{sln[i]}_TITL'Edge_cable'_WIDE_0_TYPE_PG_0.4"
2099
2100                 output += "\n_{nlc_{start_plc}_facd_1.0_nlcc_5_n_nend"
2101
2102             for i in range(len(sar)):

```



```

2103         output+= f"\nAREA_SAR_{sar[i]}_TITL_Pedestrian_weight_WIDE_0_TYPE_PG_0.35"
2104         "
2105     output += f"""\n \nLC    4 'Q' 1 TITL 'Moment'
2106     NODE 150 TYPE MYX 149.3186281074653
2107     NODE 151 TYPE MYX 149.3186281074653
2108     NODE 152 TYPE MYX -149.3186281074653
2109     NODE 153 TYPE MYX -149.3186281074653
2110     """
2111
2112     output += "\n\nLC_5'Q'_1TITL_'pedestrians_full'"
2113
2114     for i in range(len(sar_ss)):
2115         output += f"\nAREA_SAR_{sar_ss[i]}_TITL_Pedestrian_load_full_WIDE_0_TYPE_PG_4.55"
2116
2117     for j in range(len(sar_ms)):
2118         output += f"\nAREA_SAR_{sar_ms[j]}_TITL_Pedestrian_load_full_WIDE_0_TYPE_PG_3.25"
2119
2120     # Pedestrian load midspan
2121     output+= "\n\nLC_6'Q'_FACT_1.0TITL_Pedestrian_load_midspan'"
2122
2123     for k in range(len(sar_ms)):
2124         output += f"\nAREA_SAR_{sar_ms[k]}_TITL_Pedestrian_weight_WIDE_0_TYPE_PG_3.25"
2125
2126     # Pedestrian load sidespan
2127     output+= "\n\nLC_7'Q'_FACT_1.0TITL_Pedestrian_load_sidespan'"
2128
2129     for l in range(len(sar_ss)):
2130         output += f"\nAREA_SAR_{sar_ss[l]}_TITL_Pedestrian_weight_WIDE_0_TYPE_PG_4.55"
2131
2132     # Pedestrian load half midspan
2133     output+= "\n\nLC_8'Q'_FACT_1.0TITL_Pedestrian_load_midspan_half'"
2134
2135     for m in range(len(sar_ms) // 2):
2136         output += f"\nAREA_SAR_{sar_ms[m]}_TITL_Pedestrian_weight_WIDE_0_TYPE_PG_3.90"
2137
2138     output+= f"\nAREA_SAR_{sar_ms[-1]}_TITL_Pedestrian_weight_WIDE_0_TYPE_PG_3.90"
2139     "
2140     # Wind x-direction
2141     output+= "\n\nLC_9'W'_1TITL_Wind_x-direction'"
2142
2143     for n in range(len(sln_w_xx)):
2144         output += f"\nLINE_SLN_{sln_w_xx[n]}_TITL_Wind_x-direction_WIDE_0_TYPE_PYY_{q_wind}"
2145
2146     output += f"\nLINE_SLN_{sln_w_xx_p[0]}_TITL_Wind_x-direction_WIDE_0_TYPE_PYY_{q_wind}"
2147
2148     for o in range(len(grp_pyl)):
2149         output += f"\nBEAM_GRP_{grp_pyl[o]}_TYPE_PYY_PA_{q_wind_pyl_1}_PE_{q_wind_pyl_2}"
2150
2151     # Wind y-direction
2152     output += f"\n\nLC_10'W'_1TITL_Wind_y-direction'"
2153
2154     for p in range(len(sln_w_yy)):
2155         output += f"\nLINE_SLN_{sln_w_yy[p]}_TITL_Wind_girders_WIDE_0_TYPE_PXX_{q_wind*0.4}"
2156
2157     for q in range(len(grp_pyl)):
2158         output += f"\nBEAM_GRP_{grp_pyl[q]}_TYPE_PXX_PA_{q_wind_pyl_1}_PE_{q_wind_pyl_2}"
2159
2160     # Wind z-direction
2161     output+= "\n\nLC_11'W'_1TITL_Wind_z-direction'"
2162
2163     for p in range(len(sln_w_zz)):
2164         output += f"\nLINE_SLN_{sln_w_zz[p]}_TITL_Wind_girders_WIDE_0_TYPE_PZZ_2"
2165
2166     # Add. moment ULS
2167     output+= "\n\nLC_12'Q'_1TITL_Add_moment_ULS'"
2168
2169     output+= f"\nPOINT_NODE_2600_WIDE_0_TYPE_MXX_200"
2170     output+= f"\nPOINT_NODE_2602_WIDE_0_TYPE_MXX_200"

```

```

2171
2172 output+= f"\nP0IN_NODE_ '2601' _WIDE_0_ TYPE_MXX_200"
2173 output+= f"\nP0IN_NODE_ '2603' _WIDE_0_ TYPE_MXX_200"
2174
2175 # Add. moment ALS
2176 output+= " \n \n LC_13_ 'A' _1_ TITL_ 'Add. _moment_ ALS'"
2177
2178 output+= f"\nP0IN_NODE_ '2600' _WIDE_0_ TYPE_MXX_-120"
2179 output+= f"\nP0IN_NODE_ '2602' _WIDE_0_ TYPE_MXX_-120"
2180
2181 output+= f"\nP0IN_NODE_ '2601' _WIDE_0_ TYPE_MXX_120"
2182 output+= f"\nP0IN_NODE_ '2603' _WIDE_0_ TYPE_MXX_120"
2183
2184 # Ship impact loading
2185 output+= " \n \n LC_14_ 'A' _1_ TITL_ 'Ship_impact_loading'"
2186 output+= f"\nLINE_ AUTO_ TITL_ 'Accidental_1_ ' _PROJ_ N_0.500000_ TYPE_PYY_P1_1000_
2187 27.60957_-2.555000_6.378248_P2_1000_28.60957_-2.555000_6.378248"
2188
2189 # Imperfection ULS
2190 output += "" "\n
2191 LC 15 'Q' 1 TITL 'Imperfection ULS'
2192 BEAM grp 301,304 type UYS PA 0 PE 1/300
2193 BEAM grp 302,303 type UYS PA 0 PE -1/300
2194
2195 LC 16 'Q' 1 TITL 'Imperfection ALS'
2196 BEAM grp 302,304 type UZS PA 0 PE 1/300
2197 BEPL grp 301,303 type UZS P -1/300 A 50[%]
2198 ""
2199 if con_ms == 0:
2200 sln = sln_w_zz
2201
2202 output = f""
2203
2204 +PROG SOFILOAD
2205 HEAD EXPORT FROM DATABASE
2206 UNIT TYPE 5
2207
2208 LC 1 'G' 1 FACD 1 TITL "Selfweight"
2209
2210 LC 2 'G' 1 TITL "Additional deadload"
2211 ""
2212
2213 for i in range(len(sar)):
2214
2215 output += f"\nAREA_SAR_ '{sar[i]}' _TITL_ 'Epoxy_Finish' _WIDE_0_ TYPE_PG_0.10"
2216
2217 for i in range(len(sln)):
2218 output += f"\nLINE_SLN_ '{sln[i]}' _TITL_ 'Edge_+_cable' _WIDE_0_ TYPE_PG_0.4"
2219
2220 output += " \n \n LC_3_ 'G' _FACT_1.0_ TITL_ 'Static_load_pedestrians_(TC3)!' "
2221
2222 for i in range(len(sar)):
2223
2224 output+= f"\nAREA_SAR_ '{sar[i]}' _TITL_ 'Pedestrian_weight' _WIDE_0_ TYPE_PG_0.35
2225 "
2226
2227 output += f"" "\n \n LC 4 'Q' 1 TITL 'Moment'
2228 NODE 150 TYPE MYX 149.3186281074653
2229 NODE 151 TYPE MYX 149.3186281074653
2230 NODE 152 TYPE MYX -149.3186281074653
2231 NODE 153 TYPE MYX -149.3186281074653
2232 ""
2233
2234 output += " \n \n LC_5_ 'Q' _1_ TITL_ 'pedestrians_full_'"
2235
2236 for i in range(len(sar_ss)):
2237 output += f"\nAREA_SAR_ '{sar_ss[i]}' _TITL_ 'Pedestrian_load_full' _WIDE_0_ TYPE_
2238 PG_4.55"
2239
2240 for j in range(len(sar_ms)):
2241 output += f"\nAREA_SAR_ '{sar_ms[j]}' _TITL_ 'Pedestrian_load_full' _WIDE_0_ TYPE_
2242 PG_3.25"
2243
2244 # Pedestrian load midspan
2245 output+= " \n \n LC_6_ 'Q' _FACT_1.0_ TITL_ 'Pedestrian_load_midspan'"
2246
2247 for k in range(len(sar_ms)):
2248 output += f"\nAREA_SAR_ '{sar_ms[k]}' _TITL_ 'Pedestrian_load' _WIDE_0_ TYPE_PG_
2249 3.25"

```

```

2246 # Pedestrian load sidespan
2247 output+= "\n\LC0007\Q\FACT1.0\TITL\Pedestrian_load_sidespan"
2248
2249
2250 for l in range(len(sar_ss)):
2251     output += f"\nAREA\SAR_{sar_ss[l]}\TITL\Pedestrian_load\WIDE0\TYPE_PG\
2252         4.55"
2253
2254 # Pedestrian load half midspan
2255 output+= "\n\LC0008\Q\FACT1.0\TITL\Pedestrian_load_midspan_half"
2256
2257 for m in range(len(sar_ms) // 2):
2258     output += f"\nAREA\SAR_{sar_ms[m]}\TITL\Pedestrian_load\WIDE0\TYPE_PG\
2259         3.90"
2260
2261 # Wind x-direction
2262 output+= "\n\LC0009\W\1\TITL\Wind_x-direction"
2263
2264 for n in range(len(sln_w_xx)):
2265     output += f"\nLINE\SLN_{sln_w_xx[n]}\TITL\Wind_x-direction\WIDE0\TYPE_PYY\
2266         {q_wind}"
2267
2268 for o in range(len(grp_pyl)):
2269     output += f"\nBEAM\GRP_{grp_pyl[o]}\TYPE_PYY_PA_{q_wind_pyl_1}\PE_{
2270         q_wind_pyl_2}"
2271
2272 # Wind y-direction
2273 output += f"\n\LC0010\W\1\TITL\Wind_y-direction"
2274
2275 for p in range(len(sln_w_yy)):
2276     output += f"\nLINE\SLN_{sln_w_yy[p]}\TITL\Wind_girders\WIDE0\TYPE_PXX\{
2277         q_wind*0.4}"
2278
2279 for q in range(len(grp_pyl)):
2280     output += f"\nBEAM\GRP_{grp_pyl[q]}\TYPE_PXX_PA_{q_wind_pyl_1}\PE_{
2281         q_wind_pyl_2}"
2282
2283 # Wind z-direction
2284 output+= "\n\LC0011\W\1\TITL\Wind_z-direction"
2285
2286 for p in range(len(sln_w_zz)):
2287     output += f"\nLINE\SLN_{sln_w_zz[p]}\TITL\Wind_girders\WIDE0\TYPE_PZZ\2"
2288
2289 # Add. moment ULS
2290 output+= "\n\LC0012\Q\1\TITL\Add_moment_ULS"
2291
2292 output+= f"\nPOIN\NODE_2600\WIDE0\TYPE_MXX-200"
2293 output+= f"\nPOIN\NODE_2602\WIDE0\TYPE_MXX-200"
2294
2295 output+= f"\nPOIN\NODE_2601\WIDE0\TYPE_MXX200"
2296 output+= f"\nPOIN\NODE_2603\WIDE0\TYPE_MXX200"
2297
2298 # Add. moment ALS
2299 output+= "\n\LC0013\A\1\TITL\Add_moment_ALS"
2300
2301 output+= f"\nPOIN\NODE_2600\WIDE0\TYPE_MXX-120"
2302 output+= f"\nPOIN\NODE_2602\WIDE0\TYPE_MXX-120"
2303
2304 output+= f"\nPOIN\NODE_2601\WIDE0\TYPE_MXX120"
2305 output+= f"\nPOIN\NODE_2603\WIDE0\TYPE_MXX120"
2306
2307 # Ship impact loading
2308 output+= "\n\LC0014\A\1\TITL\Ship_impact_loading"
2309 output+= f"\nLINE\AUTO\TITL\Accidental\1\PROJ\N\0.500000\TYPE_PYY_P1\1000\
2310     27.60957-2.555000\6.378248\2\1000\28.60957-2.555000\6.378248"
2311
2312 # Imperfection ULS
2313 output += ""
2314 LC 15 'Q' 1 TITL 'Imperfection ULS'
2315 BEAM grp 301,304 type UYS PA 0 PE 1/300
2316 BEAM grp 302,303 type UYS PA 0 PE -1/300
2317
2318 LC 16 'Q' 1 TITL 'Imperfection ULS'
2319 BEAM grp 302,304 type UZS PA 0 PE 1/300
2320 BEPL grp 301,303 type UZS P -1/300 A 50[%]
2321 ""
2322
2323 return output
2324
2325 result_loadcases = (

```

```

2319     print_loadcases(sar_ss, sar_ms, sln_w_xx, sln_w_xx_p, sln_w_yy, sln_w_zz, sln_w_p,
2320                     pnt_pyl)
2321 )
2322 #endregion
2323
2324 #region BB) Girder load combinations
2325 grider_combinations = f"""
2326
2327 +prog ase
2328 head calculate all linear loadcases
2329 CTRL WARN 398
2330
2331 lc (1 15 1)
2332
2333 end
2334
2335 +prog ase urs:16
2336 head
2337
2338 syst prob th3 plc {start_plc}
2339
2340 lc 100 facd 1.2 type (D) titl 'ULS girder full'
2341     lcc 2 fact 1.20
2342     lcc 4 fact 1.00
2343     lcc 5 fact 1.35
2344     lcc 9 fact 0.45
2345     lcc 10 fact 0.18
2346     lcc 11 fact -0.45
2347
2348 end
2349
2350
2351 +prog ase urs:17
2352 head
2353
2354 syst prob th3 plc {start_plc}
2355
2356 lc 110 facd 1.2 type (D) titl 'ULS girder mid'
2357     lcc 2 fact 1.20
2358     lcc 4 fact 1.00
2359     lcc 6 fact 1.35
2360     lcc 9 fact 0.45
2361     lcc 10 fact 0.18
2362     lcc 11 fact -0.45
2363
2364 end
2365
2366 +prog ase urs:18
2367 head
2368
2369 syst prob th3 plc {start_plc}
2370
2371 lc 120 facd 1.2 type (D) titl 'ULS girder side'
2372     lcc 2 fact 1.20
2373     lcc 4 fact 1.00
2374     lcc 7 fact 1.35
2375     lcc 9 fact 0.45
2376     lcc 10 fact 0.18
2377     lcc 11 fact -0.45
2378
2379 end
2380
2381 +prog ase urs:19
2382 head
2383
2384 syst prob th3 plc {start_plc}
2385
2386 lc 130 facd 1.2 type (D) titl 'ULS girder half mid'
2387     lcc 2 fact 1.20
2388     lcc 4 fact 1.00
2389     lcc 8 fact 1.35
2390     lcc 9 fact 0.45
2391     lcc 10 fact 0.18
2392     lcc 11 fact -0.45
2393
2394 end
2395
2396 +prog ase urs:20
2397 head
2398
2399 syst prob th3 plc {start_plc}

```

```

2398 lc 140 facd 1.00 type (D) titl 'ALS girder'
2399     lcc 2 fact 1.00
2400     lcc 4 fact 1.00
2401     lcc 5 fact 0.40
2402     lcc 9 fact 0.30
2403     lcc 10 fact 0.12
2404     lcc 11 fact -0.30
2405     lcc 14 fact 1.0
2406 end
2407
2408 ""
2409
2410 #endregion
2411
2412
2413 #region CC) Von mises stress check girder + cables
2414 grp_cables = np.concatenate((cab, grp_h_ss, grp_h_ms))
2415
2416
2417 von_mises_stress_check = f"""
2418 +prog aqb urs:25.1
2419 ctrl msel stee
2420 echo full full
2421
2422 lc (100 140 10)
2423 beam grp 4 type beam
2424
2425 nstr ulti ksv ul chks +1.0
2426 comb gmax lcst 150
2427 stre e styp m0
2428
2429 end
2430
2431 +prog aqb
2432
2433 beam grp {'.'.join(map(str, grp_cables))} type cabl
2434
2435 lc (100 140 10)
2436
2437 nstr ulti ksv ul chks +1.0
2438 comb gmax lcst 160
2439 stre e styp m0
2440
2441 end
2442 ""
2443
2444 #endregion
2445
2446
2447 #region DD) Calculate eigenfrequencies
2448
2449 plc_ef = f"""
2450
2451 +prog ase
2452 head TH3 for cable shape
2453
2454 syst prob th3 plc {start_plc + 10}
2455
2456 lc {end_plc} facd 1.00 type -
2457     lcc 2 fact 1.0
2458     lcc 3 fact 1.0
2459
2460 end
2461 ""
2462
2463 eigenfrequencies = f"""
2464
2465 +prog ase
2466 head calculate eigenfrequencies
2467
2468 syst prob line plc {end_plc}
2469
2470 MASS 0
2471 MASS LC 2 SELE PZZ MX 100[%] MY 100[%] MZ 100[%]
2472 MASS LC 3 SELE PZZ MX 100[%] MY 100[%] MZ 100[%]
2473
2474 eige 40 etyp lanc lc 200
2475
2476 end
2477 ""

```

```

2478
2479 if isfile(rf"E:\Python-CDB\Runs\{filename}-part1.pdf"):
2480     pass
2481 else:
2482     eigenfrequencies += f""
2483     +sys nowait ursula "{filename}-part1.plb" -printto:"pdf" -page:2-3
2484     ""
2485
2486 #endregion
2487
2488
2489 #region EE) Pylon buckling load combinations
2490 pylon_buckling_combinations = f""
2491
2492 +prog ase
2493
2494 syst prob th3 plc {start_plc + 10} stor yes
2495
2496 end
2497
2498
2499 -prog ase urs
2500 CTRL WARN 293
2501 syst prob th3 iter 40
2502 rein mod beam rmod SAVE
2503
2504 grp - line
2505 grp (301 302 1) full
2506
2507 rein lcr 300
2508 DESI ULTI KSV ULD KSB ULD C1 3.50 S1 24 S2 50 SMOD YES tana 45 tanb 45
2509 NSTR S1 KSV ULD KSB ULD CHKC 30 CHKR 469
2510
2511 lc 300 facd 1.2 type (D) titl 'ULS pylon + imp.'
2512     lcc 2 fact 1.20
2513     lcc 4 fact 1.00
2514     lcc 5 fact 1.35
2515     lcc 9 fact 0.45
2516     lcc 10 fact 0.18
2517     lcc 11 fact -0.45
2518     lcc 12 fact 1.00
2519     lcc 15 fact 1.00
2520
2521 end
2522
2523 -prog aqb
2524 head lc 300
2525 echo full full
2526
2527 BEAM GRP (301 302 1)
2528
2529
2530 LC 300
2531 REIN BEAM LCR 300 RMOD SING
2532 DESI ULTI KSV ULD KSB ULD C1 3.50 S1 24 S2 50 SMOD YES tana 45 tanb 45
2533 NSTR S1 KSV ULD KSB ULD CHKC 30 CHKR 469
2534
2535 end
2536
2537 +prog ase
2538 CTRL WARN 293
2539 syst prob th3 plc 30 fmax 1.1 tol -2
2540
2541 grp - line
2542 grp (301 304 1) full
2543
2544 $rein lcr 310 rmod save
2545 $desi serv ksv sld ksb sld c1 2.1 s1 2.4 s2 5.0 smod yes tana 45 tanb 45
2546 nstr k1 ksv sld ksb sld crac yes cw 0.2 fmax 0.85 $chkc 45 chkr 500
2547
2548 lc 310 facd 1.00 type (D) titl 'SLS pylon + imp.'
2549     lcc 2 fact 1.00
2550     lcc 4 fact 1.00
2551     lcc 5 fact 1.00
2552     lcc 9 fact 0.30
2553     lcc 10 fact 0.12
2554     lcc 11 fact -0.30
2555     lcc 12 fact 0.67
2556 end
2557

```

```

2558 +prog aqb
2559 head lc 310
2560 echo full full
2561
2562 BEAM GRP (301 304 1)
2563
2564 LC 310
2565 rein beam lcr 310 rmod sing
2566 desi serv ksv sld ksb sld c1 3.5 s1 24 s2 50 smod yes tana 45 tanb 45
2567 nstr k1 ksv sld ksb sld chkc 45 chkr 500 crac yes cw 0.2
2568
2569 end
2570
2571 +prog ase
2572 CTRL WARN 293
2573 syst prob th3 plc 30 fmax 1.1 tol -2
2574 rein mod beam rmod save
2575
2576 grp - line
2577 grp (301 304 1) full
2578
2579 $rein lcr 320
2580 $DESI ULTI KSV ULD KSB ULD C1 2.1 S1 2.4 S2 5.0 SMOD YES tana 45 tanb 45
2581 NSTR K1 KSV ULD KSB ULD FMAX 0.85 $CHKC 30 CHKR 469
2582
2583 lc 320 facd 1.0 type (D) titl 'ALS pylon + imp.'
2584 lcc 2 fact 1.00
2585 lcc 4 fact 1.00
2586 lcc 5 fact 0.40
2587 lcc 9 fact 0.30
2588 lcc 10 fact 0.12
2589 lcc 11 fact -0.30
2590 lcc 13 fact 1.00
2591 lcc 14 fact 1.00
2592 $lcc 16 fact 1.00
2593
2594 end
2595
2596 +prog aqb
2597 head lc 320
2598 echo full full
2599
2600 BEAM GRP (301 304 1)
2601
2602 LC 320
2603 REIN BEAM LCR 320 RMOD SING
2604 DESI ULTI KSV ULD KSB ULD C1 3.50 S1 24 S2 50 SMOD YES tana 45 tanb 45
2605 NSTR K1 KSV ULD KSB ULD CHKC 30 CHKR 469
2606
2607 end
2608
2609 ""
2610 #endregion
2611
2612
2613
2614 # Part 1 - Stress check girders + cables & eigenfrequency calculation
2615 with open(rf"E:\Python-CDB\Runs\{filename}-part1.1.txt", "w") as file:
2616
2617     # Print cs_generation to the file
2618     file.write(f"{cs_generation}\n")
2619
2620     # SOFIMSHC run geometry
2621
2622     file.write(f"{foundation(mesh=0.30)}\n")
2623
2624     # SOFIMSHC for additional geometry
2625     file.write(f"{result_g_ss}\n")
2626     file.write(f"{result_g_ms}\n")
2627     file.write(f"{result_c_ss}\n")
2628     file.write(f"{result_c_ms}\n")
2629     file.write(f"{result_tr_ss}\n")
2630     file.write(f"{result_tr_ms}\n")
2631     file.write(f"{result_pyl}\n")
2632     file.write(f"{result_p_ss}\n")
2633     file.write(f"{result_p_ms}\n")
2634     file.write(f"{result_h_ss}\n")
2635     file.write(f"{result_h_ms}\n")
2636     file.write(f"{result_d_pl_ss}\n")
2637     file.write(f"{result_d_pl_ms}\n")

```

```

2638     file.write(f"{result_c_pyl}\n")
2639     file.write(f"{result_cab_pyl}\n")
2640     file.write(f"{result_rol_cab}\n")
2641     file.write(f"{result_con_mg2}\n")
2642     file.write(f"{result_points_pylon}\n")
2643     file.write(f"end\n")
2644
2645     # AQUA smoothen cross sections
2646     file.write(f"{smoothen_cs}\n")
2647
2648     # SOFILOAD apply moment and cable stress
2649     file.write(f"{result_con_mom}\n")
2650     file.write(f"{apply_cable_stress}\n")
2651     file.write(f"{adjust_cable_stress}\n")
2652
2653     # SOFILOAD apply static loadcases
2654     file.write(f"{result_loadcases}\n")
2655     file.write(f"end\n")
2656
2657     # ASE run girder
2658     file.write(f"{grider_combinations}\n")
2659
2660     # AQB von mises stress check
2661     file.write(f"{von_mises_stress_check}\n")
2662
2663     file.write(f"{plc_ef}\n")
2664
2665     with open(rf"E:\Python-CDB\Runs\{filename}-part1.2.txt", "w") as file:
2666         # Calculate eigenfrequencies
2667         file.write(f"{eigenfrequencies}\n")
2668
2669
2670     # Part 2 - Dynamic analysis
2671     with open(rf"E:\Python-CDB\Runs\{filename}-part2.1.txt", "w") as file:
2672
2673         file.write(f"{cs_generation}\n")
2674
2675         # SOFIMSHC run geometry
2676
2677         file.write(f"{foundation(mesh_□=□0.60)}\n")
2678
2679         # SOFIMSHC for additional geometry
2680         file.write(f"{result_g_ss}\n")
2681         file.write(f"{result_g_ms}\n")
2682         file.write(f"{result_c_ss}\n")
2683         file.write(f"{result_c_ms}\n")
2684         file.write(f"{result_tr_ss}\n")
2685         file.write(f"{result_tr_ms}\n")
2686         file.write(f"{result_pyl}\n")
2687         file.write(f"{result_p_ss}\n")
2688         file.write(f"{result_p_ms}\n")
2689         file.write(f"{result_h_ss}\n")
2690         file.write(f"{result_h_ms}\n")
2691         file.write(f"{result_d_pl_ss}\n")
2692         file.write(f"{result_d_pl_ms}\n")
2693         file.write(f"{result_c_pyl}\n")
2694         file.write(f"{result_cab_pyl}\n")
2695         file.write(f"{result_rol_cab}\n")
2696         file.write(f"{result_con_mg2}\n")
2697         file.write(f"{result_points_pylon}\n")
2698
2699
2700     with open(rf"E:\Python-CDB\Runs\{filename}-part2.3.txt", "w") as file:
2701         file.write(f"end\n")
2702
2703     # AQUA smoothen cross sections
2704     file.write(f"{smoothen_cs}\n")
2705
2706     # SOFILOAD apply moment and cable stress
2707     file.write(f"{result_con_mom}\n")
2708     file.write(f"{apply_cable_stress}\n")
2709     file.write(f"{adjust_cable_stress}\n")
2710
2711     # SOFILOAD apply static loadcases
2712     file.write(f"{result_loadcases}\n")
2713     file.write(f"end\n")
2714
2715
2716     # Part 3 - Pylon buckling
2717     with open(rf"E:\Python-CDB\Runs\{filename}-part3.txt", "w") as file:

```



```

2718
2719     file.write(f"{cs_generation}\n")
2720
2721     # SOFIMSHC run geometry
2722
2723     file.write(f"{foundation(mesh_□=□0.60)}\n")
2724
2725     # SOFIMSHC for additional geometry
2726     file.write(f"{result_g_ss}\n")
2727     file.write(f"{result_g_ms}\n")
2728     file.write(f"{result_c_ss}\n")
2729     file.write(f"{result_c_ms}\n")
2730     file.write(f"{result_tr_ss}\n")
2731     file.write(f"{result_tr_ms}\n")
2732     file.write(f"{result_pyl}\n")
2733     file.write(f"{result_p_ss}\n")
2734     file.write(f"{result_p_ms}\n")
2735     file.write(f"{result_h_ss}\n")
2736     file.write(f"{result_h_ms}\n")
2737     file.write(f"{result_d_pl_ss}\n")
2738     file.write(f"{result_d_pl_ms}\n")
2739     file.write(f"{result_c_pyl}\n")
2740     file.write(f"{result_cab_pyl}\n")
2741     file.write(f"{result_rol_cab}\n")
2742     file.write(f"{result_con_mg2}\n")
2743     file.write(f"{result_points_pylon}\n")
2744     file.write(f"end\n")
2745
2746     # AQUA smoothen cross sections
2747     file.write(f"{smoothen_cs}\n")
2748
2749     # SOFILOAD apply moment and cable stress
2750     file.write(f"{result_con_mom}\n")
2751     file.write(f"{apply_cable_stress}\n")
2752     file.write(f"{adjust_cable_stress}\n")
2753
2754     # SOFILOAD apply static loadcases
2755     file.write(f"{result_loadcases}\n")
2756     file.write(f"end\n")
2757
2758     # Perform pylon buckling check
2759     file.write(f"{pylon_buckling_combinations}\n")
2760
2761
2762
2763 def plot_3d_lines(coordinates, points, name=filename):
2764     fig = plt.figure(figsize=(20, 10))
2765
2766     # Adjust margins and spacing
2767     plt.subplots_adjust(hspace=0.3, wspace=0.3)
2768
2769     # First subplot: Sideview (elev=0, azim=90)
2770     ax1 = fig.add_subplot(121, projection='3d') # Top-left subplot
2771     for (x1, y1, z1), (x2, y2, z2) in coordinates:
2772         ax1.plot([x1, x2], [y1, y2], [z1, z2], color='b', linewidth=0.5)
2773         ax1.scatter([x1, x2], [y1, y2], [z1, z2], color='r', s=3)
2774     for (x1, y1, z1), (x2, y2, z2), (x3, y3, z3), (x4, y4, z4) in points:
2775         poly_points = [
2776             [x1, y1, z1],
2777             [x2, y2, z2],
2778             [x3, y3, z3],
2779             [x4, y4, z4]
2780         ]
2781         poly = Poly3DCollection([poly_points], color='grey', edgecolor='black', alpha
                                =0.7)
2782         # ax1.add_collection3d(poly) # Uncomment if polygons need to be plotted
2783     ax1.set_box_aspect([100, 20, 20])
2784     ax1.view_init(elev=10, azim=90)
2785
2786     # Second subplot: Perspective view (elev=20, azim=60)
2787     ax2 = fig.add_subplot(122, projection='3d') # Top-right subplot
2788     for (x1, y1, z1), (x2, y2, z2) in coordinates:
2789         ax2.plot([x1, x2], [y1, y2], [z1, z2], color='b', linewidth=0.5)
2790         ax2.scatter([x1, x2], [y1, y2], [z1, z2], color='r', s=3)
2791     for (x1, y1, z1), (x2, y2, z2), (x3, y3, z3), (x4, y4, z4) in points:
2792         poly_points = [
2793             [x1, y1, z1],
2794             [x2, y2, z2],
2795             [x3, y3, z3],
2796             [x4, y4, z4]

```

```

2797     ]
2798     poly = Poly3DCollection([poly_points], color='grey', edgecolor='black', alpha
2799                             =0.7)
2800     # ax2.add_collection3d(poly) # Uncomment if polygons need to be plotted
2801     ax2.set_box_aspect([100, 20, 20])
2802     ax2.view_init(elev=30, azim=50)
2803
2804     fig.suptitle(f"Iteration_{filename}", fontsize=16)
2805
2806     # Save the figure as a .png file
2807     plt.tight_layout()
2808     plt.subplots_adjust(left=0.1, right=0.9, top=0.9, bottom=0.1, wspace=0.3)
2809     plt.savefig(rf"E:\Python-CDB\Plots\{name}.png", dpi=300)
2810
2811     #endregion
2812
2813     plot_3d_lines(coordinates=plt_model, points=plt_model_deck)
2814
2815     return end_plc

```

A.2 ULS verification - Step 5

The parametric script for the ULS verification of the main girders and cable system is presented in the Python script below.

```
1  """ULS verification of the parametric model"""
2
3  import numpy as np
4  import matplotlib.pyplot as plt
5  from mpl_toolkits.mplot3d import Axes3D
6  from mpl_toolkits.mplot3d.art3d import Poly3DCollection
7  from matplotlib import cm
8  from sofistik_daten import *
9  import os
10 import platform
11 from ctypes import *
12 import tabula
13 import pandas as pd
14 import csv
15 import time
16 import sys, subprocess
17 import matplotlib.pyplot as plt
18 import re
19 import PyPDF2
20 import matplotlib.pyplot as plt
21 import matplotlib.font_manager as fm
22 from sklearn.cluster import AffinityPropagation
23 from sklearn.cluster import DBSCAN
24 from matplotlib.backends.backend_pdf import PdfPages
25 from matplotlib import image as mimg
26
27 #region Connect to CDB
28 os.add_dll_directory(r"E:\SOFiSTiK\2024\SOFiSTiK_2024\interfaces\64bit")
29 os.add_dll_directory(r"E:\SOFiSTiK\2024\SOFiSTiK_2024")
30
31 # Get the DLL functions
32 myDLL = cdll.LoadLibrary("sof_cdb_w_edu-2024.dll")
33 py_sof_cdb_get = cdll.LoadLibrary("sof_cdb_w_edu-2024.dll").sof_cdb_get
34 py_sof_cdb_get.restype = c_int
35 py_sof_cdb_kenq = cdll.LoadLibrary("sof_cdb_w_edu-2024.dll").sof_cdb_kenq_ex
36
37 # Connect to CDB
38 Index = c_int()
39 cdbIndex = 99
40
41
42 def stress_check_girder_cables(fileName):
43     uc_girder = []
44     uc_cable = []
45
46     Index.value = myDLL.sof_cdb_init(fileName.encode('utf-8'), cdbIndex)
47
48     cdbStat = c_int() # get the CDB status
49     cdbStat.value = myDLL.sof_cdb_status(Index.value)
50
51     print("CDB_Status:", cdbStat.value)
52
53     a = c_int()
54     ie = c_int(0)
55     RecLen = c_int(sizeof(cbeam_de0))
56
57     while ie.value < 2:
58         ie.value = py_sof_cdb_get(Index, 107, 100, byref(cbeam_de0), byref(RecLen), 1)
59         uc_girder.append(cbeam_de0.m_tcf)
60         break
61
62     while ie.value < 2:
63         ie.value = py_sof_cdb_get(Index, 107, 110, byref(cbeam_de0), byref(RecLen), 1)
64         uc_girder.append(cbeam_de0.m_tcf)
65         break
66
67     while ie.value < 2:
68         ie.value = py_sof_cdb_get(Index, 107, 120, byref(cbeam_de0), byref(RecLen), 1)
69         uc_girder.append(cbeam_de0.m_tcf)
70         break
71
72     while ie.value < 2:
73         ie.value = py_sof_cdb_get(Index, 107, 130, byref(cbeam_de0), byref(RecLen), 1)
74         uc_girder.append(cbeam_de0.m_tcf)
```

```

75         break
76
77     while ie.value < 2:
78         ie.value = py_sof_cdb_get(Index, 107, 140, byref(cbeam_de0), byref(RecLen), 1)
79         uc_girder.append(cbeam_de0.m_tcf)
80         break
81
82
83     a = c_int()
84     ie = c_int(0)
85     RecLen = c_int(sizeof(ccabl_str))
86
87     while ie.value < 2:
88         ie.value = py_sof_cdb_get(Index, 165, 160, byref(ccabl_str), byref(RecLen), 1)
89         uc_cable.append((ccabl_str.m_sig / 1000) / 460)
90         break
91
92     myDLL.sof_cdb_close(0)
93
94     return max(uc_girder), uc_cable

```

A.3 Mode shape clustering - Step 7

The parametric script for the mode shape clustering according to the DBSCAN algorithm for crowd-induced loading is presented in the code below.

```
1  """Mode shape clustering algorithm and conversion to SOFiSTiK for crowd-induced loading"""
2
3  from sofistik_daten import *
4  import os
5  import platform
6  from ctypes import *
7  import pandas as pd
8  import matplotlib.pyplot as plt
9  import numpy as np
10 from sklearn.cluster import AffinityPropagation
11 from mpl_toolkits.mplot3d import Axes3D
12 from sklearn.cluster import DBSCAN
13 from matplotlib.lines import Line2D
14 from matplotlib.backends.backend_pdf import PdfPages
15 from matplotlib import image as mpimg
16
17
18 #region Connect to CDB
19 os.add_dll_directory(r"E:\SOFiSTiK\2024\SOFiSTiK_2024\interfaces\64bit")
20 os.add_dll_directory(r"E:\SOFiSTiK\2024\SOFiSTiK_2024")
21
22
23 # Get the DLL functions
24 myDLL = cdll.LoadLibrary("sof_cdb_w_edu-2024.dll")
25 py_sof_cdb_get = cdll.LoadLibrary("sof_cdb_w_edu-2024.dll").sof_cdb_get
26 py_sof_cdb_get.restype = c_int
27 py_sof_cdb_kenq = cdll.LoadLibrary("sof_cdb_w_edu-2024.dll").sof_cdb_kenq_ex
28
29
30 # Connect to CDB
31 Index = c_int()
32 cdbIndex = 99
33
34 def modeshape_loading(filename, fileName, lc):
35
36     lc_dyn = []
37     ef_dyn = []
38     point = []
39     pointnr = []
40
41     with open(rf"E:\Python-CDB\Runs\{filename}load.txt", "w") as file:
42         file.write(f"+prog_sofload\n")
43
44     for num, k in enumerate(lc): #CHANGE BACK TO ALL lc
45
46         # important: Unicode call!
47         Index.value = myDLL.sof_cdb_init(fileName.encode('utf-8'), cdbIndex)
48
49         cdbStat = c_int() # get the CDB status
50         cdbStat.value = myDLL.sof_cdb_status(Index.value)
51
52         # Print the Status of the CDB
53         print("CDB_Status:", cdbStat.value)
54
55         pos = c_int(0)
56         datalen = c_int(0)
57
58         a = c_int()
59         ie = c_int(0)
60         datalen.value = sizeof(CNODE)
61         RecLen = c_int(sizeof(cnode))
62
63         nr, x, y, z = [], [], [], []
64         id, uz = [], []
65
66         while ie.value < 2:
67             ie.value = py_sof_cdb_get(Index, 20, 0, byref(cnode), byref(RecLen), 1)
68             if 2.555 > cnode.m_xyz[1] > 0.05 or - 2.555 < cnode.m_xyz[1] < -0.05:
69                 nr.append(cnode.m_nr) # Node number
70                 x.append(cnode.m_xyz[0]) # X coordinates
71                 y.append(cnode.m_xyz[1]) # Y coordinates
72                 z.append(cnode.m_xyz[2]) # Z coordinates
73
74         a = c_int()
```

```

75     ie = c_int(0)
76     datalen.value = sizeof(CN_DISPC)
77     RecLen = c_int(sizeof(cn_dispc))
78
79     while ie.value < 2:
80         ie.value = py_sof_cdb_get(Index, 24, k, byref(cn_dispc), byref(RecLen), 1)
81         id.append(cn_dispc.m_id)          # node-number
82         uz.append(cn_dispc.m_uz)         # displacement Z
83
84     ef = []
85
86     ie = c_int(0)
87     RecLen = c_int(sizeof(clc_eige))
88
89     while ie.value < 2:
90         ie.value = py_sof_cdb_get(Index, 12, k, byref(clc_eige), byref(RecLen), 1)
91         ef.append(clc_eige.m_omega / (2*np.pi))    # node-number
92
93     myDLL.sof_cdb_close(0)
94
95     ef_dyn.append(ef[-1])
96
97     nr2 = []
98     uz2 = []
99     x2 = []
100    y2 = []
101    z2 = []
102
103    for i in range(len(nr)):
104        for j in range(len(id)):
105            if nr[i] == id[j]:
106                nr2.append(nr[i])
107                x2.append(x[i])
108                y2.append(y[i])
109                uz2.append(uz[j])
110                z2.append(z[i])
111
112    # Create the node variable with 3D points
113    node = [(x2[i], y2[i], uz2[i]) for i in range(len(x2))]
114
115    # Create points to read maximum acceleration
116    max_node = max(node, key=lambda n: n[2])
117    index_max = node.index(max_node, key=lambda n: n[2])
118    min_node = min(node, key=lambda n: n[2])
119    index_min = node.index(min_node, key=lambda n: n[2])
120
121    # Compare the absolute values of uz2
122    if abs(max_node[2]) > abs(min_node[2]):
123        point.append((x2[index_max], y2[index_max], z2[index_max])) # point.append(nr[
124                                index_max])
125    else:
126        point.append((x2[index_min], y2[index_min], z2[index_min]))
127
128    # Separate points into pos_node and neg_node based on z2 (uz2)
129    pos_node = []
130    neg_node = []
131
132    # Now, node contains all the 3D points
133    for i in range(len(node)):
134        if node[i][2] < 0 and -2.555 < node[i][1] < 2.555 and -51.51 < node[i][0] <
135            51.51:
136            neg_node.append(node[i])
137        elif node[i][2] > 0 and -2.555 < node[i][1] < 2.555 and -51.51 < node[i][0] <
138            51.51:
139            pos_node.append(node[i])
140
141    # Convert pos_node and neg_node to NumPy arrays for clustering
142    neg_node_array = np.array(neg_node)
143    pos_node_array = np.array(pos_node)
144
145    # Check if neg_node contains enough points for clustering
146    if len(neg_node_array) < 2:
147        raise ValueError("Not enough points in neg_node for clustering.")
148    if len(pos_node_array) < 2:
149        raise ValueError("Not enough points in pos_node for clustering.")
150
151    # Apply DBSCAN with Manhattan distance for neg_node
152    dbscan_neg = DBSCAN(eps=0.5, min_samples=7, metric='euclidean')
153    labels_neg = dbscan_neg.fit_predict(neg_node_array)

```

```

152
153 # Apply DBSCAN with Manhattan distance for pos_node
154 dbscan_pos = DBSCAN(eps=0.5, min_samples=7, metric='euclidean')
155 labels_pos = dbscan_pos.fit_predict(pos_node_array)
156
157 # Get the unique cluster labels (excluding noise)
158 unique_labels_neg = np.unique(labels_neg[labels_neg != -1])
159 unique_labels_pos = np.unique(labels_pos[labels_pos != -1])
160
161 # Initialize a dictionary to hold the boundaries for each cluster in neg_node
162 boundaries_neg = {}
163 boundaries_pos = {}
164
165 # Plot the clusters for neg_node
166 for i, label in enumerate(unique_labels_neg):
167
168     # Filter points belonging to the current cluster
169     cluster_points = neg_node_array[labels_neg == label]
170
171     # Calculate boundaries
172     min_x = np.min(cluster_points[:, 0])
173     max_x = np.max(cluster_points[:, 0])
174     min_y = np.min(cluster_points[:, 1])
175     max_y = np.max(cluster_points[:, 1])
176
177     # Find the index of the points with min_x, max_x, min_y, max_y
178     min_x_idx = np.where(cluster_points[:, 0] == min_x)[0][0]
179     max_x_idx = np.where(cluster_points[:, 0] == max_x)[0][0]
180     min_y_idx = np.where(cluster_points[:, 1] == min_y)[0][0]
181     max_y_idx = np.where(cluster_points[:, 1] == max_y)[0][0]
182
183     # Store boundaries and indices in the dictionary
184     boundaries_neg[label] = {
185         'min_x': min_x,
186         'max_x': max_x,
187         'min_y': min_y,
188         'max_y': max_y,
189         'min_x_idx': min_x_idx,
190         'max_x_idx': max_x_idx,
191         'min_y_idx': min_y_idx,
192         'max_y_idx': max_y_idx
193     }
194
195 # Plot the clusters for pos_node
196 for i, label in enumerate(unique_labels_pos):
197
198     # Filter points belonging to the current cluster
199     cluster_points = pos_node_array[labels_pos == label]
200
201     # Calculate boundaries
202     min_x = np.min(cluster_points[:, 0])
203     max_x = np.max(cluster_points[:, 0])
204     min_y = np.min(cluster_points[:, 1])
205     max_y = np.max(cluster_points[:, 1])
206
207     # Find the index of the points with min_x, max_x, min_y, max_y
208     min_x_idx = np.where(cluster_points[:, 0] == min_x)[0][0]
209     max_x_idx = np.where(cluster_points[:, 0] == max_x)[0][0]
210     min_y_idx = np.where(cluster_points[:, 1] == min_y)[0][0]
211     max_y_idx = np.where(cluster_points[:, 1] == max_y)[0][0]
212
213     # Store boundaries and indices in the dictionary
214     boundaries_pos[label] = {
215         'min_x': min_x,
216         'max_x': max_x,
217         'min_y': min_y,
218         'max_y': max_y,
219         'min_x_idx': min_x_idx,
220         'max_x_idx': max_x_idx,
221         'min_y_idx': min_y_idx,
222         'max_y_idx': max_y_idx
223     }
224
225 lc_dyn.append(300 + num)
226
227 output = f"\nLC_{300+num}_type_{none}_fact_{1}_TITL_\\"Dyn._ped._loading\\"_"
228
229 # Define function for loading (psi)
230 if 1.25 < ef[-1] < 1.70:
231     psi = 0.25 + (1.00 - 0.25) / (1.70 - 1.25) * (ef[-1] - 1.25)

```

```

232     elif 1.70 < ef[-1] < 2.10:
233         psi = 1.00
234     elif 2.10 < ef[-1] < 2.30:
235         psi = 1 + ((0.25 - 1.00) / (2.30 - 2.10) * (ef[-1] - 2.10))
236     else:
237         psi = 0.25
238
239     n_p = (10.8 * np.sqrt((5.10 * 103 * 0.5) * 0.002)) / (5.10 * 103)
240
241     q = 0.280 * n_p * psi
242
243     # Print the boundaries and indices for each cluster
244     for label, bounds in boundaries_neg.items():
245         output += (f"\nAREA_{AUTO}_{TITL}\NEG_{CLUSTER_{label+1}}\_{PROJ}_{ZZ}_{WIDE}_{5}_{TYPE}_{PG}"
246                   f"P1_{1}_{bounds['min_x']:.3f}_{bounds['min_y']:.3f}_{7}"
247                   f"P2_{1}_{bounds['min_x']:.3f}_{bounds['max_y']:.3f}_{7}"
248                   f"P3_{1}_{bounds['max_x']:.3f}_{bounds['max_y']:.3f}_{7}"
249                   f"P4_{1}_{bounds['max_x']:.3f}_{bounds['min_y']:.3f}_{7}")
250
251     for label, bounds in boundaries_pos.items():
252         output += (f"\nAREA_{AUTO}_{TITL}\POS_{CLUSTER_{label+1}}\_{PROJ}_{ZZ}_{WIDE}_{5}_{TYPE}_{PG}"
253                   f"P1_{1}_{bounds['min_x']:.3f}_{bounds['min_y']:.3f}_{7}"
254                   f"P2_{1}_{bounds['min_x']:.3f}_{bounds['max_y']:.3f}_{7}"
255                   f"P3_{1}_{bounds['max_x']:.3f}_{bounds['max_y']:.3f}_{7}"
256                   f"P4_{1}_{bounds['max_x']:.3f}_{bounds['min_y']:.3f}_{7}")
257
258     output += f"\nFUNC_{T_{1}}/{round(ef[-1],3)}_{F_{q}}_{TMIN_{round(ef[-1]/(2*np.pi),3)}"
259             "
260
261     with open(rf"E:\Python-CDB\Runs\{filename}load.txt", "a") as file:
262         file.write(f"{output}\n")
263
264     with open(rf"E:\Python-CDB\Runs\{filename}load.txt", "a") as file:
265         file.write(f"\nEND")
266
267     #store points with maximum displacement for dynamic analysis
268
269     for i in range(len(point)):
270         current_point = point[i]
271         while point.count(current_point) > 1: # Check if duplicates exist
272             # Increment x2 value by 0.1
273             current_point = (current_point[0] + 0.1, current_point[1], current_point[2])
274             point[i] = current_point # Update the point in the list
275
276     output = ""
277
278     for i in range(len(point)):
279         output += f"\nSPT_{3000+i}_{X_{point[i][0]}}_{point[i][1]}}_{point[i][2]}}_{TITL}"
280             Point\"
281         pointnr.append(3000 + i)
282
283     with open(rf"E:\Python-CDB\Runs\{filename}-part2.2.txt", "w") as file:
284         file.write(f"{output}\n")
285
286     return pointnr, ef_dyn, lc_dyn

```


A.4 MOPSO algorithm - Steps 1, 9 & 10

The Python script for the multi-objective particle swarm optimisation of the case study and the generated Pareto front with its export to a csv-file is presented in the code below.

```
1  """MOPSO algorithm with Pareto front generation and export to csv-file"""
2
3  import numpy as np
4  import matplotlib.pyplot as plt
5  from mpl_toolkits.mplot3d import Axes3D
6  from mpl_toolkits.mplot3d.art3d import Poly3DCollection
7  from matplotlib import cm
8  from sofiistik_daten import *
9  import os
10 import platform
11 from ctypes import *
12 import tabula
13 import pandas as pd
14 import csv
15 import time
16 import sys, subprocess
17 import matplotlib.pyplot as plt
18 import re
19 import PyPDF2
20 import matplotlib.pyplot as plt
21 import matplotlib.font_manager as fm
22 import multiprocessing
23 from sklearn.cluster import AffinityPropagation
24 from matplotlib.cm import get_cmap
25 from sklearn.cluster import DBSCAN
26 from matplotlib.backends.backend_pdf import PdfPages
27 from matplotlib import image as mpimg
28 from Parametric_model import par_model
29 from Stresses_AQB import stress_check_girder_cables
30 from Read_eigenfrequencies import read_eigenfrequencies
31 from Modeshape_loading import modeshape_loading
32 from Dynamic_analysis import dynamic_analysis
33 from Dynamic_analysis import dynamic_results
34 from Optimisation_script import optimisation
35 import shutil
36 import platform
37
38 start_time = time.time()
39
40
41 # PSO Parameters
42 num_particles = 15
43 num_iterations = 100
44 inertia = 0.7 # Inertia parameter
45 c1 = 1.5 # Memory parameter 1.5
46 c2 = 1.5 # Social parameter 1.5
47
48
49 # Initialize bounds for each objective (min-max normalisation)
50 objective_bounds = [
51     [0, 4], # Bounds for 'a_max'
52     [350, 450], # Bounds for 'mass'
53 ]
54
55
56 # Set the uc_total constraint bounds and penalty parameters
57 uc_lower_bound = 0.5 # Lower bound for uc_total
58 uc_upper_bound = 1.0 # Upper bound for uc_total
59 penalty_factor = 1.0
60
61
62
63 # Define utility functions (1/2)
64 def dominates(score1, score2):
65     """
66     Check if score1 dominates score2.
67     A solution dominates another if it's no worse in all objectives
68     and strictly better in at least one.
69     """
70     return np.all(score1 <= score2) and np.any(score1 < score2)
71
72
73 # Define utility functions (1/2)
74 def find_global_best_position(personal_best_scores):
```

```

75 """
76 Find the global best position using Pareto dominance.
77 A solution is considered the best if it is not dominated by any other.
78 If no dominance occurs, return the first particle.
79 """
80 num_particles = len(personal_best_scores)
81 dominated = [False] * num_particles # To track if a particle is dominated by any other
82
83 for i in range(num_particles):
84     for j in range(num_particles):
85         if i != j and dominates(personal_best_scores[j], personal_best_scores[i]):
86             dominated[i] = True
87             break # No need to check further if already dominated
88
89 # Get the indices of the non-dominated particles (Pareto-optimal)
90 pareto_optimal_indices = [i for i, dom in enumerate(dominated) if not dom]
91
92 if not pareto_optimal_indices:
93     # If no Pareto-optimal particles found, select the first particle
94     print("No Pareto dominance found.")
95     return 0 # Index of the first particle
96
97 # Otherwise, select the first Pareto-optimal solution
98 global_best_index = pareto_optimal_indices[0]
99
100 return global_best_index
101
102
103 # Global dictionary to store evaluated particles and their fitness values
104 evaluated_particles = {}
105
106
107 # Define fitness function with caching to avoid repetitive evaluation
108 def fitness_func(particle, objective_bounds, uc_lower_bound, uc_upper_bound, penalty_factor):
109     # Convert particle to a tuple for hashing (since lists are not hashable)
110     particle_tuple = tuple(particle)
111
112     # Check if this particle has been evaluated before
113     if particle_tuple in evaluated_particles:
114         print(f"Particle {particle} has already been evaluated.")
115         return evaluated_particles[particle_tuple] # Return cached fitness value
116
117     # Extract individual parameters
118     hw, wf, hp, cp, hang_ss, hang_ms = particle
119
120     # Define the hanger penalty based on the number of hangers
121     if hang_ms == 2:
122         hanger_penalty = 0
123     elif hang_ms == 3:
124         hanger_penalty = 0.5 # Penalty for 3 hangers
125     elif hang_ms == 4:
126         hanger_penalty = 1 # Penalty for 4 hangers
127
128     # Perform computation
129     tw = 10
130     tf = 20
131     wl = 1100
132     wu = 600
133     fz_p = 10
134     hang_ms_p = 4
135     runtime = 8000 # Adjust runtime as needed
136     stepsize = 40
137
138
139
140 # Run optimization and compute objectives
141 a_max, a_max_index, a_max_p, mode, ef_dyn, uc_girder, uc_cable, mass = optimisation(
142     hp, int(hang_ss), int(hang_ms), int(hang_ms_p), fz_p, hw, wf, tw, tf, wl, wu, cp,
143     runtime, stepsize
144 )
145
146 # Compute unity_checks, uc_total, and penalty
147 unity_checks = [uc_girder, uc_cable[0]]
148 uc_total = np.max(unity_checks)
149 penalty = 0
150 if uc_total < uc_lower_bound:
151     penalty = penalty_factor
152 elif uc_lower_bound > uc_total > uc_upper_bound:
153     penalty = abs(penalty_factor * ((uc_total - uc_upper_bound) / (uc_lower_bound -
154         uc_upper_bound)))

```

```

153     else:
154         penalty = penalty_factor
155
156     # Normalize objectives
157     objectives = [a_max, mass[0]]
158     normalized_objectives = []
159     for i, obj in enumerate(objectives):
160         lower_bound, upper_bound = objective_bounds[i]
161         # Dynamically update bounds
162         objective_bounds[i][0] = min(obj, lower_bound)
163         objective_bounds[i][1] = max(obj, upper_bound)
164         # Normalize
165         normalized_value = abs((obj - lower_bound) / (upper_bound - lower_bound))
166         normalized_objectives.append(normalized_value)
167
168     normalized_objectives = np.array(normalized_objectives)
169
170     # Total fitness
171     total_fitness = normalized_objectives + penalty + hanger_penalty
172
173     # Nominal values
174     nominal_values = [a_max, mass[0], uc_total]
175
176     # Cache the fitness value for the particle
177     evaluated_particles[particle_tuple] = (total_fitness, (a_max_index, a_max_p, mode, ef_dyn
178         ), nominal_values, unity_checks)
179
180     return total_fitness, (a_max_index, a_max_p, mode, ef_dyn), nominal_values, unity_checks
181
182
183 # Initialize particles
184 girder = ['500x180', '700x250', '900x320', '1100x390', '1200x425']
185 girder_mapping = {girder: idx for idx, girder in enumerate(girder)}
186
187 particle_cs = np.random.choice(list(girder_mapping.keys()), num_particles)
188
189 particle_hw = np.array([int(cs.split('x')[0]) for cs in particle_cs]) # Girder height
190 particle_wf = np.array([int(cs.split('x')[1]) for cs in particle_cs]) # Girder width
191
192 hp_bounds = [10, 22] # Height of pylon
193 particle_hp = np.round(np.random.randint(hp_bounds[0], hp_bounds[1], num_particles), 0)
194
195 hangers = ['1x2', '1x3', '2x4']
196 hanger_mapping = {hanger: idx for idx, hanger in enumerate(hangers)}
197
198 particle_hangers = np.random.choice(list(hanger_mapping.keys()), num_particles)
199
200 particle_hang_ss = np.array([int(cs.split('x')[0]) for cs in particle_hangers]) # Hangers (
201     short span)
202 particle_hang_ms = np.array([int(cs.split('x')[1]) for cs in particle_hangers]) # Hangers (
203     main span)
204
205 cp_bounds = [-0.5, 0.5] # Some control parameter
206 particle_cp = np.random.choice(cp_bounds, num_particles)
207
208 # Combine particles into one array
209 particles = np.column_stack((particle_hw, particle_wf, particle_hp, particle_cp,
210     particle_hang_ss, particle_hang_ms))
211 # print("Particles:\n", particles)
212
213 # Determine ranges for each variable
214 hw_range = max(particle_hw) - min(particle_hw)
215 wf_range = max(particle_wf) - min(particle_wf)
216 hp_range = hp_bounds[1] - hp_bounds[0]
217 cp_range = max(cp_bounds) - min(cp_bounds)
218 hang_ss_range = max(particle_hang_ss) - min(particle_hang_ss)
219 hang_ms_range = max(particle_hang_ms) - min(particle_hang_ms)
220
221 # Ranges for velocity scaling
222 ranges = np.array([hw_range, wf_range, hp_range, cp_range, hang_ss_range, hang_ms_range])
223
224 # Initialize velocities scaled to the ranges
225 velocities = np.random.uniform(-ranges, ranges, particles.shape)
226
227 # Determine personal and global best
228 personal_best_positions = particles.copy()
229
230 # Determine personal and global best scores

```

```

229 personal_best_scores = [fitness_func(p, objective_bounds, uc_lower_bound, uc_upper_bound,
    penalty_factor)[0] for p in particles]
230
231
232 # Retrieve global best index
233 global_best_index = find_global_best_position(personal_best_scores)
234
235
236 # Retrieve the global best position
237 global_best_position = personal_best_positions[global_best_index]
238
239
240 # Initialize total archive and Pareto archive
241 total_archive = []
242 pareto_archive = []
243
244
245 # Define pareto update function
246 def update_pareto_archive(pareto_archive, particle, objectives, metadata):
247     """
248     Update the Pareto archive with a new particle.
249
250     Parameters:
251         pareto_archive (list): Current Pareto archive [(particle, objectives, metadata)].
252         particle (ndarray): New particle to consider for the archive.
253         objectives (ndarray): Objectives of the new particle.
254         metadata (dict): Additional information about the particle.
255
256     Returns:
257         list: Updated Pareto archive.
258     """
259     to_remove = []
260     for i, (archive_particle, archive_objectives, archive_metadata) in enumerate(
        pareto_archive):
261         if dominates(objectives, archive_objectives):
262             to_remove.append(i) # Current particle dominates archive_particle
263         elif dominates(archive_objectives, objectives):
264             return pareto_archive # Current particle is dominated, no update needed
265
266     # Add the current particle and its objectives to the archive
267     pareto_archive.append((particle, objectives, metadata))
268
269     # Remove dominated particles
270     for index in reversed(to_remove):
271         pareto_archive.pop(index)
272
273     return pareto_archive
274
275
276 # Define back to csv function
277 def save_archive_to_csv(archive, filename, archive_type="total"):
278     """
279     Save the archive to a CSV file, including total fitness values.
280     """
281     with open(filename, "w", newline="") as csvfile:
282         writer = csv.writer(csvfile)
283         if archive_type == "total":
284             # Include total fitness as a new column
285             writer.writerow(["Particle_Parameters(hw,wf,hp,cp,hang_ss,hang_ms)", "
                Metadata(a_max_index,a_max_p,mode,ef_dyn)", "Nominal_values(a_max,mass,
                uc_total)", "UnityChecks(uc_girder,uc_cable)"])
286             for particle_data, metadata in archive:
287                 # Calculate the fitness for each particle
288                 total_fitness, _, nominal_values, unity_checks = fitness_func(np.array(
                    particle_data), objective_bounds, uc_lower_bound, uc_upper_bound,
                    penalty_factor)
289                 writer.writerow([particle_data, metadata, nominal_values, total_fitness,
                    unity_checks])
290             elif archive_type == "pareto":
291                 # Include total fitness as a new column
292                 writer.writerow(["Particle_Parameters(hw,wf,hp,cp,hang_ss,hang_ms)", "
                    Metadata(a_max_index,a_max_p,mode,ef_dyn)", "Total_Fitness(a_max,mass)
                    ", "UnityChecks(uc_girder,uc_cable)"])
293                 for particle_data, total_fitness, metadata in archive:
294                     # Calculate the fitness for each particle
295                     score, _, _, unity_checks = fitness_func(np.array(particle_data),
                        objective_bounds, uc_lower_bound, uc_upper_bound, penalty_factor)
296                     writer.writerow([particle_data, metadata, total_fitness, unity_checks])
297
298

```

```

299 # PSO Algorithm
300 history = [] # To store positions for visualization
301 fitness_history = [] # To store global best fitness over iterations
302
303
304 # Define bounds for clipping
305 hw_bounds = [min(particle_hw), max(particle_hw)] # Use min/max of particle_hw
306 wf_bounds = [min(particle_wf), max(particle_wf)] # Use min/max of particle_wf
307 hp_bounds = [min(particle_hp), max(particle_hp)] # Use min/max of particle_hp
308 cp_bounds = [min(particle_cp), max(particle_cp)] # Use min/max of particle_cp
309 hang_ss_bounds = [min(particle_hang_ss), max(particle_hang_ss)] # Use min/max of
    particle_hang_ss
310 hang_ms_bounds = [min(particle_hang_ms), max(particle_hang_ms)] # Use min/max of
    particle_hang_ms
311
312
313 # Main optimization loop
314 for iteration in range(num_iterations):
315     start_time = time.time()
316
317     # Evaluate fitness for each particle
318     for i, particle in enumerate(particles):
319         objectives, metadata, nominal_values, unity_checks = fitness_func(
320             particle, objective_bounds, uc_lower_bound, uc_upper_bound, penalty_factor
321         )
322         total_archive.append((particle.tolist(), metadata)) # Store in total archive
323         pareto_archive = update_pareto_archive(pareto_archive, particle.tolist(), objectives,
            metadata)
324
325         # Update personal best if dominated
326         if dominates(objectives, personal_best_scores[i]):
327             personal_best_positions[i] = particle
328             personal_best_scores[i] = objectives
329
330         # Update global best using Pareto dominance
331         non_dominated_indices = [
332             i for i in range(num_particles)
333             if not any(dominates(personal_best_scores[j], personal_best_scores[i]) for j in range
                (num_particles) if j != i)
334         ]
335
336         global_best_position = personal_best_positions[np.random.choice(non_dominated_indices)]
337         # Random non-dominated
338
339         # Track Pareto front
340         pareto_front = [personal_best_scores[i] for i in non_dominated_indices]
341         fitness_history.append(pareto_front)
342
343         # Update particle velocities and positions
344         for i, particle in enumerate(particles):
345             r1, r2 = np.random.rand(), np.random.rand()
346             cognitive = c1 * r1 * (personal_best_positions[i] - particle)
347             social = c2 * r2 * (global_best_position - particle)
348
349             # Update velocity and position
350             velocities[i] = inertia * velocities[i] + cognitive + social
351             particles[i] += velocities[i]
352
353             # Clip particle values to the nearest specified range or value
354
355             # Cross section dimension (continuous range)
356             particles[i][0] = round(np.clip(particles[i][0], hw_bounds[0], hw_bounds[1]), 2)
357             # Nearest height
358             particles[i][1] = round(np.clip(particles[i][0] / 2.8, wf_bounds[0], wf_bounds[1]),
                2) # Nearest width
359
360             # Pylon height (continuous range)
361             particles[i][2] = round(np.clip(particles[i][2], hp_bounds[0], hp_bounds[1]), 2)
362
363             # Coupler plate (discrete values)
364             # cp_candidates = np.array(cp_bounds)
365             # particles[i][3] = cp_candidates[np.argmin(np.abs(cp_candidates - particles[i][3]))]
366
367             # Coupler plate (continuous range)
368             particles[i][3] = round(np.clip(particles[i][3], cp_bounds[0], cp_bounds[1]), 2)
369
370
371

```

```

372
373     # Hangers short span and main span (discrete values)
374     hang_ss_candidates = np.array([int(cs.split('x')[0]) for cs in hangers])
375     hang_ms_candidates = np.array([int(cs.split('x')[1]) for cs in hangers])
376     particles[i][4] = hang_ss_candidates[np.argmin(np.abs(hang_ss_candidates - particles[
377         i][4]))] # Nearest short span
378     particles[i][5] = hang_ms_candidates[np.argmin(np.abs(hang_ms_candidates - particles[
379         i][5]))] # Nearest main span
380
381 # Save positions for visualization
382 history.append(particles.copy())
383
384 # Save archives to CSV
385 if iteration % 1 == 0: # Adjust frequency as needed
386     save_archive_to_csv(total_archive, rf"total_archive_iteration_{iteration}.csv",
387         archive_type="total")
388     save_archive_to_csv(pareto_archive, rf"pareto_archive_iteration_{iteration}.csv",
389         archive_type="pareto")
390     print(f"Iteration_{iteration}: Archives saved.")
391
392 elapsed_time = time.time() - start_time
393 print(f"Iteration_{iteration+1} completed in {elapsed_time:.2f} seconds.")
394
395 # Final output
396 print("\nFinal Pareto Archive:")
397 for particle_data, objectives, metadata in pareto_archive:
398     print("Particle:", particle_data)
399     print("Objectives:", objectives)
400     print("Metadata(a_max_index, a_max_p, mode, ef_dyn):", metadata)
401
402 # Randomly select global best from Pareto archive
403 global_best_position = pareto_archive[np.random.randint(len(pareto_archive))]

```

B

Appendix B - Load cases & combinations

The load cases considered for the ultimate limit state (ULS) verification are:

- Self-weight.
- Additional dead loads.
- Pedestrian loads.
- Wind loads.

Loading by special vehicles is excluded since this concerns localised effects in the deck which have been verified in the original design. Furthermore, temperature loads have been excluded since the bridge can deform horizontally due to a horizontal roller support at the north side abutment and an expansion joint.

B.1 Self-weight

The self-weight of the structure is automatically generated in the FE software SOFiSTiK and considered in the analysis.

B.2 Additional dead loads

Additional dead loads in the form of a railing for pedestrians, ramp and epoxy finishing on top of the steel deck plate are considered. These are given by:

$$q_{railing} = 0.12 \text{ kN/m} \quad (\text{B.1})$$

where:

$q_{railing}$ = dead load of the steel railing to guide pedestrians across the bridge

$$q_{ramp} = 0.24 \text{ kN/m} \quad (\text{B.2})$$

where:

q_{ramp} = dead load of the ramp to guide cyclists away from the outer edges of the bridge

$$q_{epoxy} = 0.10 \text{ kN/m}^2 \quad (\text{B.3})$$

where:

q_{epoxy} = epoxy finish layer on top of the steel deck plate

B.3 Pedestrian loads

Pedestrian loads are considered according to EN-1991-2:2019 [56] load model 4 (crowds of pedestrians):

$$q_{fk} = 2 + \frac{120}{L + 30} \quad \text{where: } q_{fk} \geq 2.5 \text{ kN/m}^2 ; q_{fk} \leq 5.0 \text{ kN/m}^2 \quad (\text{B.4})$$

where:

q_{fk} = distributed load of pedestrians

L = length of the loaded area

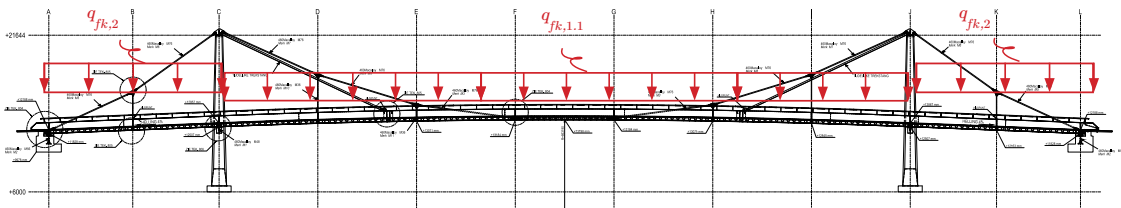
For the case study, the following crowd-loading is considered as specified below, see figure B.1. The side span has a length of 17 m and the main span has a length of 35 m.

$$q_{fk,1.1} = 2 + \frac{120}{69 + 30} = 3.25 \text{ kN/m}^2 \quad (\text{B.5})$$

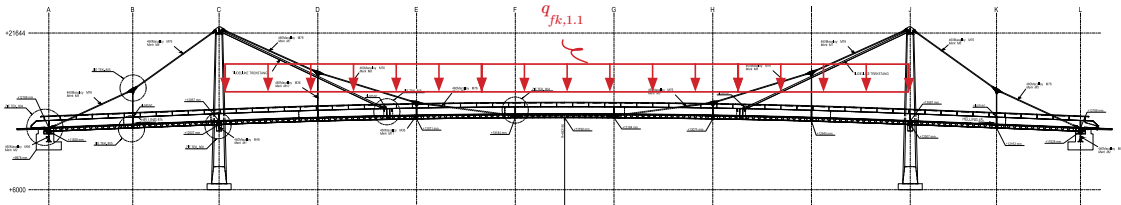
$$q_{fk,1.2} = 2 + \frac{120}{69/2 + 30} = 3.90 \text{ kN/m}^2 \quad (\text{B.6})$$

$$q_{fk,2} = 2 + \frac{120}{17 + 30} = 4.55 \text{ kN/m}^2 \quad (\text{B.7})$$

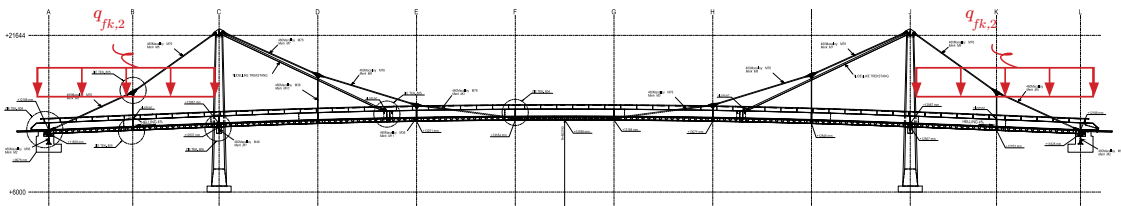
A) Fully loaded



B) Midspan loaded



C) Sidespans loaded



D) Midspan half loaded

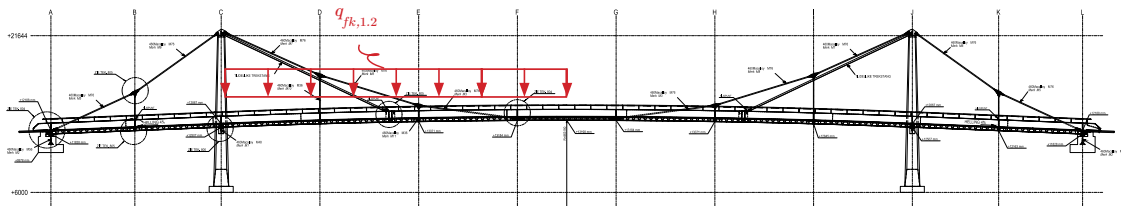


Figure B.1: Load cases considered for pedestrian loads

B.4 Wind loads

Wind loads are considered according to EN-1991-1-4-2011 in the x-, y- and z-directions. The width of the deck is 6 m and the height 1.6 m, see figure B.2.

$$q_{w,x} = 1/2 \cdot \rho \cdot v_b^2 \cdot C \cdot d_{tot} = 3.30 \text{ kN/m} \quad (\text{B.8})$$

where:

$q_{w,x}$ = Wind loading in x-direction
 ρ = air density = 1.25 kg/m^3
 v_b = basic windspeed = 24.5 m/s (zone III)
 C = $c_e \cdot c_{f,x}$ where $c_e = 2.34$, and $c_{f,x} = 1.33$
 d_{tot} = total height of the structure

$$q_{w,y} = 0.4 \cdot q_{w,x} = 1.32 \text{ kN/m} \quad (\text{B.9})$$

where:

$q_{w,y}$ = Wind loading in y-direction (40% of the loading in x-direction according to NEN-EN1991-1-4-NA-2011 [57])

$$q_{w,z} = 1/2 \cdot \rho \cdot v_b^2 \cdot C \cdot b = 3.90 \text{ kN/m} \quad (\text{B.10})$$

where:

$q_{w,z}$ = Wind loading in z-direction
 C = $c_e \cdot c_{f,z}$ where $c_e = 2.34$, and $c_{f,z} = 0.75$

$$q_{pyl,x,y,min} = 1/2 \cdot \rho \cdot v_b^2 \cdot C \cdot d_{min} = 1.50 \text{ kN/m} \quad (\text{B.11})$$

where:

$q_{pyl,x,y,min}$ = Minimum wind load in x- and y-direction for the pylons (rectangular)
 d_{min} = 0.70 m

$$q_{pyl,x,y,max} = 1/2 \cdot \rho \cdot v_b^2 \cdot C \cdot d_{max} = 3.10 \text{ kN/m} \quad (\text{B.12})$$

where:

$q_{pyl,x,y,max}$ = Maximum wind load in x- and y-direction for the pylons (rectangular)
 d_{max} = 1.5 m

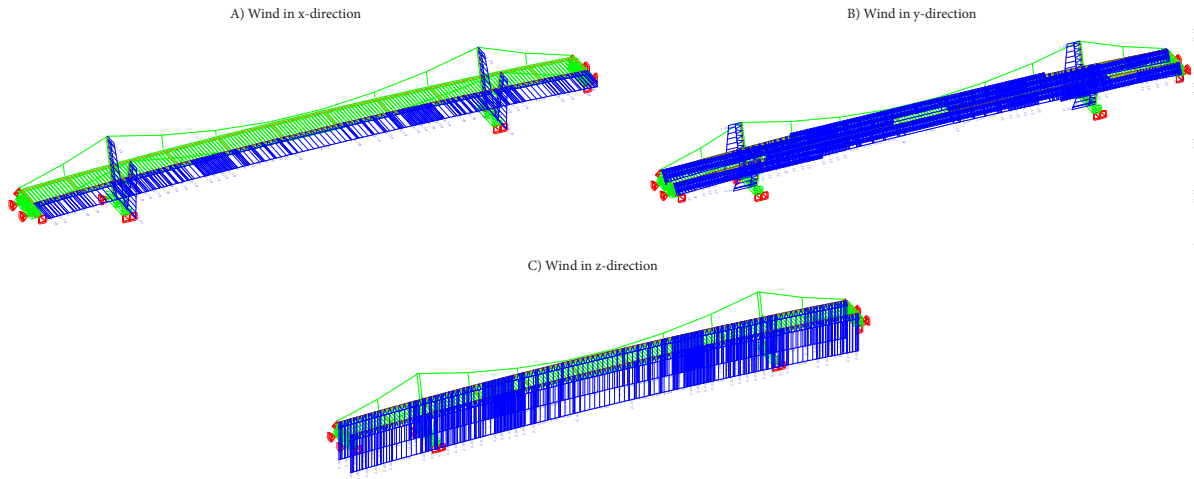


Figure B.2: Load cases considered for wind loads

B.5 Load combinations

The load combinations for ULS verification by equation 6.10a and 6.10b are considered according to EN1990-2019 [9]:

$$\sum_{j \geq 1} \gamma_{G,j} G_{k,j} + \gamma_p P + \gamma_{Q,1} \psi_{0,1} Q_{k,1} + \sum_{i > 1} \gamma_{Q,i} \psi_{0,i} Q_{k,i} \quad (6.10a)$$

$$\sum_{j \geq 1} \xi_j \gamma_{G,j} G_{k,j} + \gamma_p P + \gamma_{Q,1} Q_{k,1} + \sum_{i > 1} \gamma_{Q,i} \psi_{0,i} Q_{k,i} \quad (6.10b)$$

According to NEN-EN1990-NA-2019 [32], the equations for footbridges specify to:

$$1.30 \cdot G_{k,perm} + 1.35 \cdot \psi_{0,ped} \cdot Q_{k,ped} + \sum_{i > 1} 1.50 \cdot \psi_{0,var} \cdot Q_{k,var} \quad (6.10a^*)$$

$$1.20 \cdot G_{k,perm} + 1.35 \cdot Q_{k,ped} + \sum_{i > 1} 1.50 \cdot \psi_{0,var} \cdot Q_{k,var} \quad (6.10b^*)$$

or

$$1.20 \cdot G_{k,perm} + 1.35 \cdot Q_{k,var} + \sum_{i > 1} 1.50 \cdot \psi_{0,ped} \cdot Q_{k,ped}$$

This means that 6.10a* dominates if the self-weight and additional dead load are larger than the variable loads, in other words:

$$1.30 \cdot G_{k,perm} - 1.20 \cdot G_{k,perm} > 1.35 \cdot Q_{ped} - 1.35 \cdot Q_{ped} \cdot \psi_{0,ped} \quad (B.13)$$

or

$$1.30 \cdot G_{k,perm} - 1.20 \cdot G_{k,perm} > 1.50 \cdot Q_{var} - 1.50 \cdot Q_{var} \cdot \psi_{0,var}$$

In the original design, it is verified that the condition stated in equation B.13 is not met, meaning that 6.10b* dominates. The contributions of variable loads are greater than permanent loads.

Figure B.3 presents the ψ -factors and load combinations to consider.

Belasting	Belastingscombinaties									
	gr1	gr2	Q_{fwk}	Onbedoeld voertuig ^b	W^c		T^d		S	A1 ^{ac}
Gelijkmatig verdeelde belasting	1	0,8	0	0	0,4	0,32	0,4	0,32	0	0,4
Horizontale belasting	1	1,0	0	0	0,4	0,4	0,4	0,4	0	0,4
Dienstvoertuig Q_{serv}	0	1,0	0	0	0	0,4	0	0,4	0,8	0
Geconcentreerde belasting Q_{fwk}	0	0	1	0	0	0	0	0		0
Onbedoeld voertuig	0	0	0	1	0		0		0	0
Wind F_{wk}	0,3	0,3	0	0	1,0	1,0	0,3	0,3	0,3	0
Temperatuur	0,3	0,3	0	0	0,3	0,3	1,0	1,0	0,3	0
Sneeuw	0	0	0	0	0	0	0	0	1,0	0
Impact op of onder de brug	0	0	0	0	0	0	0	0	0	1
Aardbevingsbelasting	0	0	0	0	0	0	0	0	0	0
^a A1 = aanrijding op of onder de brug en aanvaring. ^b Te beschouwen als gewoon belastingsgeval (geen calamiteit); zie 5.6.3. ^c Bij deze combinatie is in eerste kolom de verticale belasting vermenigvuldigd met ψ_0 en de horizontale met ψ_0^2 ; in de tweede kolom is dat omgekeerd; dit is gedaan om consistent te zijn met het gebruik van de groepen verkeersbelastingen.										

Figure B.3: ψ factors and load combinations according to NEN-EN-1990-2019 [32]

This narrows down to the following load combinations:

Combination	Self-weight	Additional deadload	Pedestrian loading				Wind loading		
			full	midspan	sidespans	half midspan	x	y	z
6.10.B.1	1.20	1.20	1.35				0.18	0.45	-0.45
6.10.B.2	1.20	1.20		1.35			0.18	0.45	-0.45
6.10.B.3	1.20	1.20			1.35		0.18	0.45	-0.45
6.10.B.4	1.20	1.20				1.35	0.18	0.45	-0.45

C

Appendix C - SOFiSTiK output

C.1 Original design - ULS & direct time integration

Voldijkbrug_definitief_V1

Design Code

EuroNorm: NEN EN 1993-1-1:2005 Design of steel structures (Netherlands) V 2024
Safety Class: A2 (Buildings Safety Class 2)

Materials

Mat	Classification
1	Structural Steel - S 355 (EN 199
2	Steel - Macalloy 460 Cables
3	Concrete Piles - C45/55 N (EN 19
4	Concrete Foundation - C 30/37 (E
5	Concrete Zero Weight - C 30/37 (
6	Concrete Pylons - C 30/37 N (EN
7	Reinforcement Steel - B 500 B (E
8	Structural Steel - S 355 (EN 199
9	Steel - Macalloy 460 Cables
10	Concrete Piles - C45/55 N (EN 19
11	Concrete Foundation - C 30/37 (E
12	Concrete Zero Weight - C 30/37 (
13	Concrete Pylons - C 30/37 N (EN
14	Reinforcement Steel - B 500 B (E
15	Steel - Macalloy 460 Cables

Maximum Utilisation Level

	N	Vy	Vz	My	Mz	Mtp	Mts	Mb	Ncr	SCL	Total
	$\sigma\text{-x}$	$\sigma\text{+x}$	τ	$\sigma\text{-v}$	$\sigma\text{-s}$	$\sigma\text{-dyn}$	As-l	As-v	crack		
Section 1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	(4)	0.553
Girder - Right	0.553	0.400	0.225	0.480	-	-	-	-	-		
Section 2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	(4)	0.559
Girder - Left	0.559	0.403	0.228	0.464	-	-	-	-	-		
Total	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	(4)	0.559
	0.559	0.403	0.228	0.480	-	-	-	-	-		
N	normal force		τ	shear stress							
Vy,Vz	shear force		$\sigma\text{-v}$	principal or von Mises stress							
My,Mz	bending		$\sigma\text{-s}$	stress in reinforcements							
Mtp,Mts	torsion (p)primary and (s)econdary		$\sigma\text{-dyn}$	stress range							
Mb	warping moment		As-l	longitudinal reinforcements							
Ncr	flexural buckling		As-v	transverse reinforcements or concrete shear strength							
SCL	cross-section class		crack	crack width							
$\sigma\text{-x}$	longitud. compressive stress		Total	most unfavorable utilisation for all checks							
$\sigma\text{+x}$	longitud. tensile stress										

Voldijkbrug_definitief_V1

Design Code

EuroNorm: NEN EN 1992-1-1:2004 (NA:2011) Design of concrete structures (Netherlands) V 2024

EuroNorm: NEN EN 1993-1-1:2005 Design of steel structures

Safety Class: A2 (Buildings Safety Class 2)

Materials

Mat	Classification
1	Structural Steel - S 355 (EN 199
2	Steel - Macalloy 460 Cables
3	Concrete Piles - C45/55 N (EN 19
4	Concrete Foundation - C 30/37 (E
5	Concrete Zero Weight - C 30/37 (
6	Concrete Pylons - C 30/37 N (EN
7	Reinforcement Steel - B 500 B (E
8	Structural Steel - S 355 (EN 199
9	Steel - Macalloy 460 Cables
10	Concrete Piles - C45/55 N (EN 19
11	Concrete Foundation - C 30/37 (E
12	Concrete Zero Weight - C 30/37 (
13	Concrete Pylons - C 30/37 N (EN
14	Reinforcement Steel - B 500 B (E
15	Steel - Macalloy 460 Cables

Maximum Utilisation Level

		N	Vy	Vz	My	Mz	Mtp	Mts	Mb	Ncr	SCL	Total
		σ -x	σ +x	τ	σ -v	σ -s	σ -dyn	As-l	As-v	crack		
Section 6		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	0.684
Primary Cable - D 72		0.000	0.684	0.000	0.684	-	-	-	-	-	-	
Section 7		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	0.615
Secondary Cable - D 36		0.000	0.615	0.000	0.615	-	-	-	-	-	-	
Total		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	0.684
		0.000	0.684	0.000	0.684	-	-	-	-	-	-	
N	normal force			τ	shear stress							
Vy,Vz	shear force			σ -v	principal or von Mises stress							
My,Mz	bending			σ -s	stress in reinforcements							
Mtp,Mts	torsion (p)primary and (s)econdary			σ -dyn	stress range							
Mb	warping moment			As-l	longitudinal reinforcements							
Ncr	flexural buckling			As-v	transverse reinforcements or concrete shear strength							
SCL	cross-section class			crack	crack width							
σ -x	longitud. compressive stress			Total	most unfavorable utilisation for all checks							
σ +x	longitud. tensile stress											

Voldijkbrug_definitief_V1
VM1

Beam Elements

Finite beam elements without intermediate sections
Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam
Primary load case 20

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	460.987	460.987	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.000	-0.002	5.016	7.179E+03	-2.750E-01	3.208E+00	
				-2.750E-01	6.333E+05	5.835E-01	
				3.208E+00	5.835E-01	6.315E+05	
active ¹	455.087	457.937	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.001	0.018	5.016	7.241E+03	-2.870E-01	4.793E-01	
				-2.870E-01	6.333E+05	-4.413E+01	
				4.793E-01	-4.413E+01	6.261E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 300

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00481	0.64977	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	169.8	0.00	0.00	0.00	

Parameter of System of Equations

Number of unknowns 43268 (Direct sparse Gauss-Solver)
Total entries 1065492
Total entries after fill in 4763489
Mass matrix 268618 (consistent), incl. rotational masses
Damping matrix 1065313 (consistent)

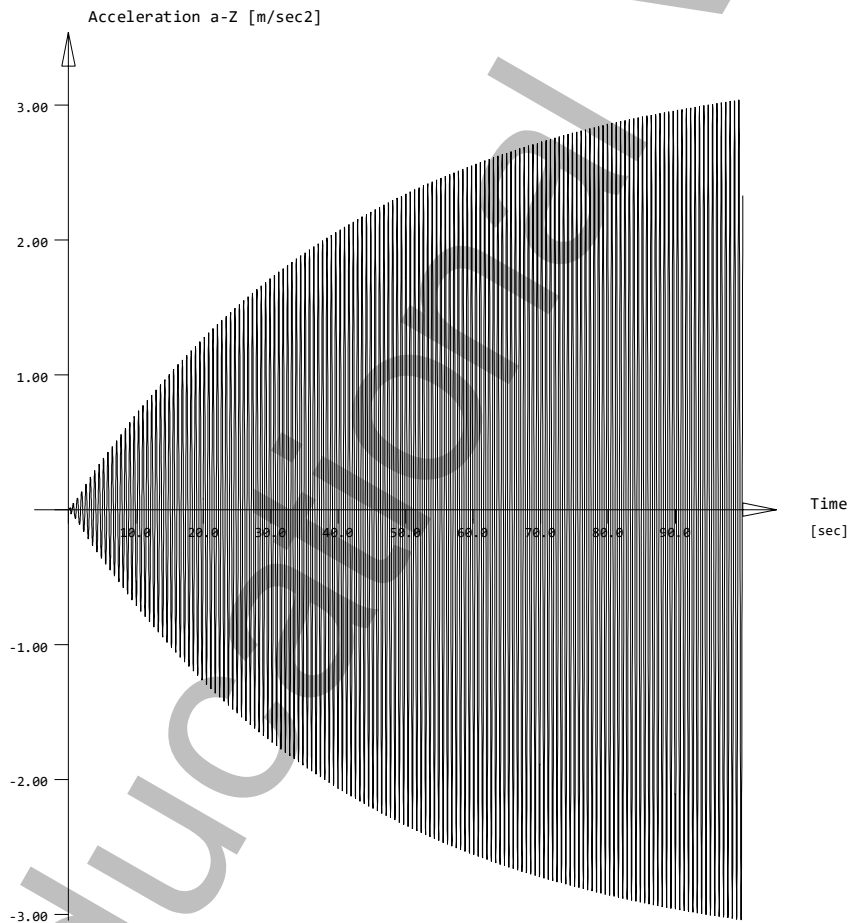
Results

Time history for Nodal accelerations

Node 901 (E 401) a-Z MINIMUM = -3.04 MAXIMUM = 3.04 [m/sec2]
stored in database for DYNR with identification no 700

Voldijkbrug_definitief_V1
Plotting results

Plotting results
Acceleration a-Z 901



Voldijkbrug_definitief_V1
TM1

System- and Control Information

Control Information

QUAD-elements with higher non conforming modes 3
QUAD-elements with all rotational degrees of freedom
Primary load case 20 t = 0.0000 [sec] +Opt:add_GS
Number of unknowns 43268
unknowns per node 6
Number of timesteps 10000
Time-step 0.0100
Printing intervall 1
damping factor A 2.529E-02
damping factor B 2.471E-04
Integr.Parameter beta 0.25
Integr.Parameter delta 0.50
Integr.Parameter theta 1.00

Groups

Grp	Option	CS	Factor	Rayleigh-A [1/sec]	Rayleigh-B [sec]	ξ [o/o]	Wind
0	FULL		1.000	0.000000	0.000000	0.00	
2	FULL		1.000	0.000000	0.000000	0.00	
4	FULL		1.000	0.000000	0.000000	0.00	
6	FULL		1.000	0.000000	0.000000	0.00	
7	FULL		1.000	0.000000	0.000000	0.00	
8	FULL		1.000	0.000000	0.000000	0.00	
9	FULL		1.000	0.000000	0.000000	0.00	
10	FULL		1.000	0.000000	0.000000	0.00	
11	FULL		1.000	0.000000	0.000000	0.00	
12	FULL		1.000	0.000000	0.000000	0.00	
13	FULL		1.000	0.000000	0.000000	0.00	
14	FULL		1.000	0.000000	0.000000	0.00	
15	FULL		1.000	0.000000	0.000000	0.00	
16	FULL		1.000	0.000000	0.000000	0.00	
17	FULL		1.000	0.000000	0.000000	0.00	
18	FULL		1.000	0.000000	0.000000	0.00	
19	FULL		1.000	0.000000	0.000000	0.00	
20	FULL		1.000	0.000000	0.000000	0.00	
21	FULL		1.000	0.000000	0.000000	0.00	
22	FULL		1.000	0.000000	0.000000	0.00	
23	FULL		1.000	0.000000	0.000000	0.00	
24	FULL		1.000	0.000000	0.000000	0.00	
25	FULL		1.000	0.000000	0.000000	0.00	
30	FULL		1.000	0.000000	0.000000	0.00	
41	FULL		1.000	0.000000	0.000000	0.00	
42	FULL		1.000	0.000000	0.000000	0.00	
43	FULL		1.000	0.000000	0.000000	0.00	
50	FULL		1.000	0.000000	0.000000	0.00	
61	FULL		1.000	0.000000	0.000000	0.00	
301	FULL		1.000	0.000000	0.000000	0.00	
302	FULL		1.000	0.000000	0.000000	0.00	
303	FULL		1.000	0.000000	0.000000	0.00	
304	FULL		1.000	0.000000	0.000000	0.00	
CS construction stage				Rayleigh-B	stiffness proportional damping ratio		
Factor Factor on stiffness				ξ	modal damping ratio		
Rayleigh-A mass proportional damping ratio				Wind	options for dynamic wind loading		

Voldijkbrug_definitief_V1
TM1

Beam Elements

Finite beam elements without intermediate sections
Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam
Primary load case 20

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	460.987	460.987	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.000	-0.002	5.016	7.179E+03	-2.750E-01	3.208E+00	
				-2.750E-01	6.333E+05	5.835E-01	
				3.208E+00	5.835E-01	6.315E+05	
active ¹	455.087	457.937	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.001	0.018	5.016	7.241E+03	-2.870E-01	4.793E-01	
				-2.870E-01	6.333E+05	-4.413E+01	
				4.793E-01	-4.413E+01	6.261E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 301

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00396	0.40519	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	0.6	0.00	0.00	0.00	

Parameter of System of Equations

Number of unknowns 43268 (Direct sparse Gauss-Solver)
Total entries 1065492
Total entries after fill in 4763489
Mass matrix 268618 (consistent), incl. rotational masses
Damping matrix 1065313 (consistent)

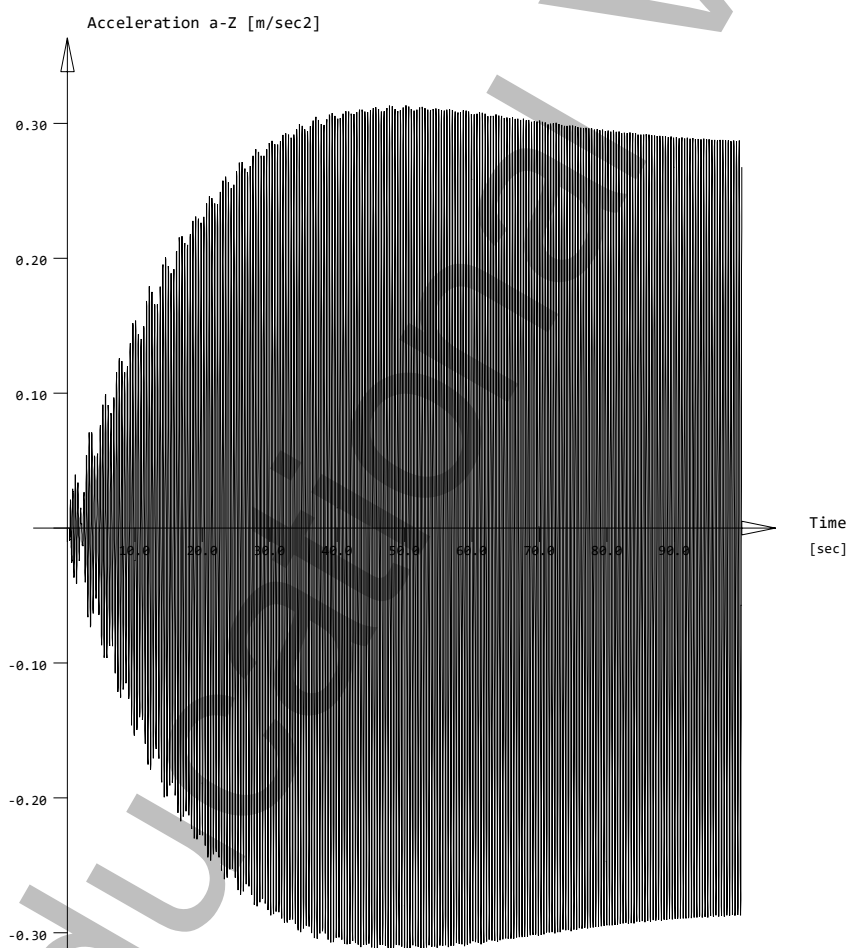
Results

Time history for Nodal accelerations

Node 901 (E 401) a-Z MINIMUM = -0.31 MAXIMUM = 0.31 [m/sec2]
stored in database for DYNR with identification no 701

Voldijkbrug_definitief_V1
Plotting results

Plotting results
Acceleration a-Z 901



Plotting results

reduced scale factor 0.917

Voldijkbrug_definitief_V1
VM2

System- and Control Information

Control Information

QUAD-elements with higher non conforming modes 3
QUAD-elements with all rotational degrees of freedom
Primary load case 20 t = 0.0000 [sec] +Opt:add_GS
Number of unknowns 43268
unknowns per node 6
Number of timesteps 10000
Time-step 0.0100
Printing intervall 1
damping factor A 2.529E-02
damping factor B 2.471E-04
Integr.Parameter beta 0.25
Integr.Parameter delta 0.50
Integr.Parameter theta 1.00

Groups

Grp	Option	CS	Factor	Rayleigh-A [1/sec]	Rayleigh-B [sec]	ξ [o/o]	Wind
0	FULL		1.000	0.000000	0.000000	0.00	
2	FULL		1.000	0.000000	0.000000	0.00	
4	FULL		1.000	0.000000	0.000000	0.00	
6	FULL		1.000	0.000000	0.000000	0.00	
7	FULL		1.000	0.000000	0.000000	0.00	
8	FULL		1.000	0.000000	0.000000	0.00	
9	FULL		1.000	0.000000	0.000000	0.00	
10	FULL		1.000	0.000000	0.000000	0.00	
11	FULL		1.000	0.000000	0.000000	0.00	
12	FULL		1.000	0.000000	0.000000	0.00	
13	FULL		1.000	0.000000	0.000000	0.00	
14	FULL		1.000	0.000000	0.000000	0.00	
15	FULL		1.000	0.000000	0.000000	0.00	
16	FULL		1.000	0.000000	0.000000	0.00	
17	FULL		1.000	0.000000	0.000000	0.00	
18	FULL		1.000	0.000000	0.000000	0.00	
19	FULL		1.000	0.000000	0.000000	0.00	
20	FULL		1.000	0.000000	0.000000	0.00	
21	FULL		1.000	0.000000	0.000000	0.00	
22	FULL		1.000	0.000000	0.000000	0.00	
23	FULL		1.000	0.000000	0.000000	0.00	
24	FULL		1.000	0.000000	0.000000	0.00	
25	FULL		1.000	0.000000	0.000000	0.00	
30	FULL		1.000	0.000000	0.000000	0.00	
41	FULL		1.000	0.000000	0.000000	0.00	
42	FULL		1.000	0.000000	0.000000	0.00	
43	FULL		1.000	0.000000	0.000000	0.00	
50	FULL		1.000	0.000000	0.000000	0.00	
61	FULL		1.000	0.000000	0.000000	0.00	
301	FULL		1.000	0.000000	0.000000	0.00	
302	FULL		1.000	0.000000	0.000000	0.00	
303	FULL		1.000	0.000000	0.000000	0.00	
304	FULL		1.000	0.000000	0.000000	0.00	
CS	construction stage			Rayleigh-B	stiffness proportional damping ratio		
Factor	Factor on stiffness			ξ	modal damping ratio		
Rayleigh-A	mass proportional damping ratio			Wind	options for dynamic wind loading		

Voldijkbrug_definitief_V1
VM2

Beam Elements

Finite beam elements without intermediate sections
Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam
Primary load case 20

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	460.987	460.987	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.000	-0.002	5.016	7.179E+03	-2.750E-01	3.208E+00	
				-2.750E-01	6.333E+05	5.835E-01	
				3.208E+00	5.835E-01	6.315E+05	
active ¹	455.087	457.937	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.001	0.018	5.016	7.241E+03	-2.870E-01	4.793E-01	
				-2.870E-01	6.333E+05	-4.413E+01	
				4.793E-01	-4.413E+01	6.261E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 302

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00480	0.37467	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	1.3	0.00	0.00	0.00	

Parameter of System of Equations

Number of unknowns 43268 (Direct sparse Gauss-Solver)
Total entries 1065492
Total entries after fill in 4763489
Mass matrix 268618 (consistent), incl. rotational masses
Damping matrix 1065313 (consistent)

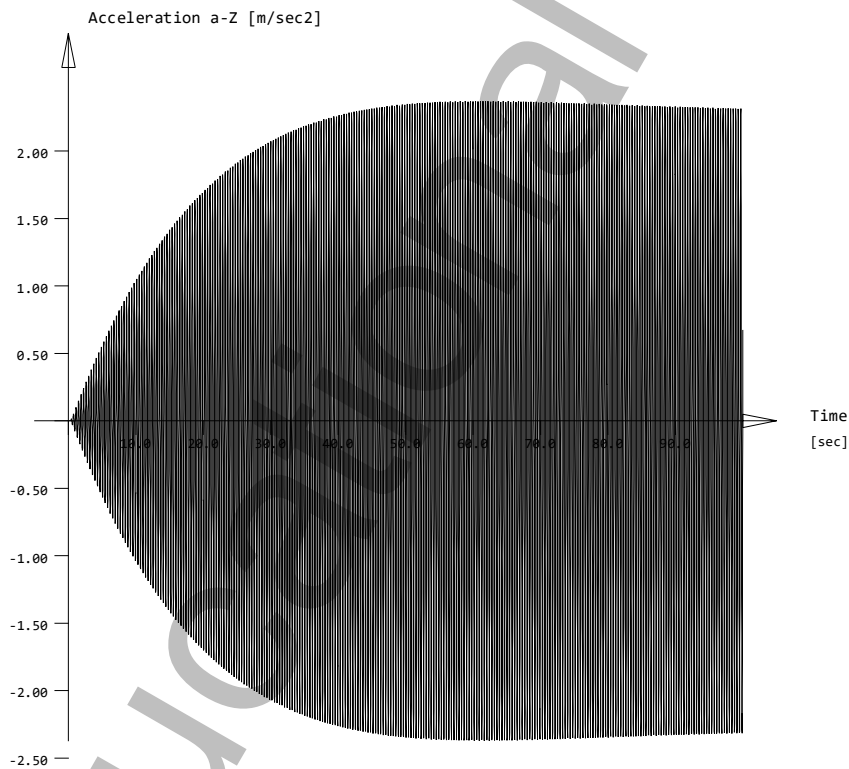
Results

Time history for Nodal accelerations

Node 1301 (E 1959) a-Z MINIMUM = -2.37 MAXIMUM = 2.37 [m/sec2]
stored in database for DYNR with identification no 702

Voldijkbrug_definitief_V1
Plotting results

Plotting results
Acceleration a-Z 1301



Voldijkbrug_definitief_V1
TM1*

System- and Control Information

Control Information

QUAD-elements with higher non conforming modes 3
QUAD-elements with all rotational degrees of freedom
Primary load case 20 t = 0.0000 [sec] +Opt:add_GS
Number of unknowns 43268
unknowns per node 6
Number of timesteps 10000
Time-step 0.0100
Printing intervall 1
damping factor A 2.529E-02
damping factor B 2.471E-04
Integr.Parameter beta 0.25
Integr.Parameter delta 0.50
Integr.Parameter theta 1.00

Groups

Grp	Option	CS	Factor	Rayleigh-A [1/sec]	Rayleigh-B [sec]	ξ [o/o]	Wind
0	FULL		1.000	0.000000	0.000000	0.00	
2	FULL		1.000	0.000000	0.000000	0.00	
4	FULL		1.000	0.000000	0.000000	0.00	
6	FULL		1.000	0.000000	0.000000	0.00	
7	FULL		1.000	0.000000	0.000000	0.00	
8	FULL		1.000	0.000000	0.000000	0.00	
9	FULL		1.000	0.000000	0.000000	0.00	
10	FULL		1.000	0.000000	0.000000	0.00	
11	FULL		1.000	0.000000	0.000000	0.00	
12	FULL		1.000	0.000000	0.000000	0.00	
13	FULL		1.000	0.000000	0.000000	0.00	
14	FULL		1.000	0.000000	0.000000	0.00	
15	FULL		1.000	0.000000	0.000000	0.00	
16	FULL		1.000	0.000000	0.000000	0.00	
17	FULL		1.000	0.000000	0.000000	0.00	
18	FULL		1.000	0.000000	0.000000	0.00	
19	FULL		1.000	0.000000	0.000000	0.00	
20	FULL		1.000	0.000000	0.000000	0.00	
21	FULL		1.000	0.000000	0.000000	0.00	
22	FULL		1.000	0.000000	0.000000	0.00	
23	FULL		1.000	0.000000	0.000000	0.00	
24	FULL		1.000	0.000000	0.000000	0.00	
25	FULL		1.000	0.000000	0.000000	0.00	
30	FULL		1.000	0.000000	0.000000	0.00	
41	FULL		1.000	0.000000	0.000000	0.00	
42	FULL		1.000	0.000000	0.000000	0.00	
43	FULL		1.000	0.000000	0.000000	0.00	
50	FULL		1.000	0.000000	0.000000	0.00	
61	FULL		1.000	0.000000	0.000000	0.00	
301	FULL		1.000	0.000000	0.000000	0.00	
302	FULL		1.000	0.000000	0.000000	0.00	
303	FULL		1.000	0.000000	0.000000	0.00	
304	FULL		1.000	0.000000	0.000000	0.00	
CS	construction stage			Rayleigh-B	stiffness proportional damping ratio		
Factor	Factor on stiffness			ξ	modal damping ratio		
Rayleigh-A	mass proportional damping ratio			Wind	options for dynamic wind loading		

Voldijkbrug_definitief_V1
TM1*

Beam Elements

Finite beam elements without intermediate sections
Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam
Primary load case 20

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	460.987	460.987	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.000	-0.002	5.016	7.179E+03	-2.750E-01	3.208E+00	
				-2.750E-01	6.333E+05	5.835E-01	
				3.208E+00	5.835E-01	6.315E+05	
active ¹	455.087	457.937	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.001	0.018	5.016	7.241E+03	-2.870E-01	4.793E-01	
				-2.870E-01	6.333E+05	-4.413E+01	
				4.793E-01	-4.413E+01	6.261E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 303

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00590	0.34542	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	1.7	0.00	0.00	0.00	

Parameter of System of Equations

Number of unknowns 43268 (Direct sparse Gauss-Solver)
Total entries 1065492
Total entries after fill in 4763489
Mass matrix 268618 (consistent), incl. rotational masses
Damping matrix 1065313 (consistent)

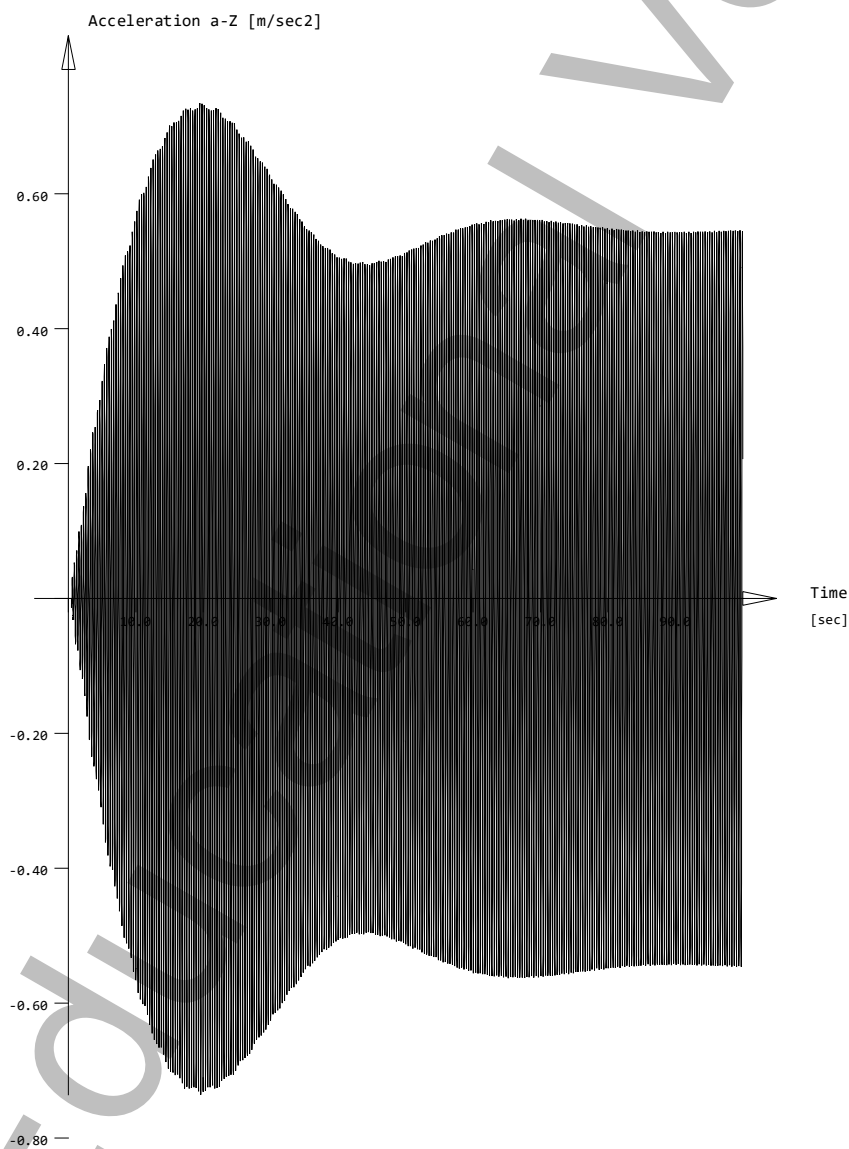
Results

Time history for Nodal accelerations

Node 901 (E 401) a-Z MINIMUM = -0.74 MAXIMUM = 0.73 [m/sec2]
stored in database for DYNR with identification no 703

Voldijkbrug_definitief_V1
Plotting results

Plotting results
Acceleration a-Z 901



Plotting results

reduced scale factor 0.917

Voldijkbrug_definitief_V1
VM3

System- and Control Information

Control Information

QUAD-elements with higher non conforming modes 3
QUAD-elements with all rotational degrees of freedom
Primary load case 20 t = 0.0000 [sec] +Opt:add_GS
Number of unknowns 43268
unknowns per node 6
Number of timesteps 10000
Time-step 0.0100
Printing intervall 1
damping factor A 2.529E-02
damping factor B 2.471E-04
Integr.Parameter beta 0.25
Integr.Parameter delta 0.50
Integr.Parameter theta 1.00

Groups

Grp	Option	CS	Factor	Rayleigh-A [1/sec]	Rayleigh-B [sec]	ξ [o/o]	Wind
0	FULL		1.000	0.000000	0.000000	0.00	
2	FULL		1.000	0.000000	0.000000	0.00	
4	FULL		1.000	0.000000	0.000000	0.00	
6	FULL		1.000	0.000000	0.000000	0.00	
7	FULL		1.000	0.000000	0.000000	0.00	
8	FULL		1.000	0.000000	0.000000	0.00	
9	FULL		1.000	0.000000	0.000000	0.00	
10	FULL		1.000	0.000000	0.000000	0.00	
11	FULL		1.000	0.000000	0.000000	0.00	
12	FULL		1.000	0.000000	0.000000	0.00	
13	FULL		1.000	0.000000	0.000000	0.00	
14	FULL		1.000	0.000000	0.000000	0.00	
15	FULL		1.000	0.000000	0.000000	0.00	
16	FULL		1.000	0.000000	0.000000	0.00	
17	FULL		1.000	0.000000	0.000000	0.00	
18	FULL		1.000	0.000000	0.000000	0.00	
19	FULL		1.000	0.000000	0.000000	0.00	
20	FULL		1.000	0.000000	0.000000	0.00	
21	FULL		1.000	0.000000	0.000000	0.00	
22	FULL		1.000	0.000000	0.000000	0.00	
23	FULL		1.000	0.000000	0.000000	0.00	
24	FULL		1.000	0.000000	0.000000	0.00	
25	FULL		1.000	0.000000	0.000000	0.00	
30	FULL		1.000	0.000000	0.000000	0.00	
41	FULL		1.000	0.000000	0.000000	0.00	
42	FULL		1.000	0.000000	0.000000	0.00	
43	FULL		1.000	0.000000	0.000000	0.00	
50	FULL		1.000	0.000000	0.000000	0.00	
61	FULL		1.000	0.000000	0.000000	0.00	
301	FULL		1.000	0.000000	0.000000	0.00	
302	FULL		1.000	0.000000	0.000000	0.00	
303	FULL		1.000	0.000000	0.000000	0.00	
304	FULL		1.000	0.000000	0.000000	0.00	
CS	construction stage			Rayleigh-B	stiffness proportional damping ratio		
Factor	Factor on stiffness			ξ	modal damping ratio		
Rayleigh-A	mass proportional damping ratio			Wind	options for dynamic wind loading		

Voldijkbrug_definitief_V1
VM3

Beam Elements

Finite beam elements without intermediate sections
Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam
Primary load case 20

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	460.987	460.987	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.000	-0.002	5.016	7.179E+03	-2.750E-01	3.208E+00	
				-2.750E-01	6.333E+05	5.835E-01	
				3.208E+00	5.835E-01	6.315E+05	
active ¹	455.087	457.937	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.001	0.018	5.016	7.241E+03	-2.870E-01	4.793E-01	
				-2.870E-01	6.333E+05	-4.413E+01	
				4.793E-01	-4.413E+01	6.261E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 304

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00481	0.22346	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	25.8	0.00	0.00	0.00	

Parameter of System of Equations

Number of unknowns 43268 (Direct sparse Gauss-Solver)
Total entries 1065492
Total entries after fill in 4763489
Mass matrix 268618 (consistent), incl. rotational masses
Damping matrix 1065313 (consistent)

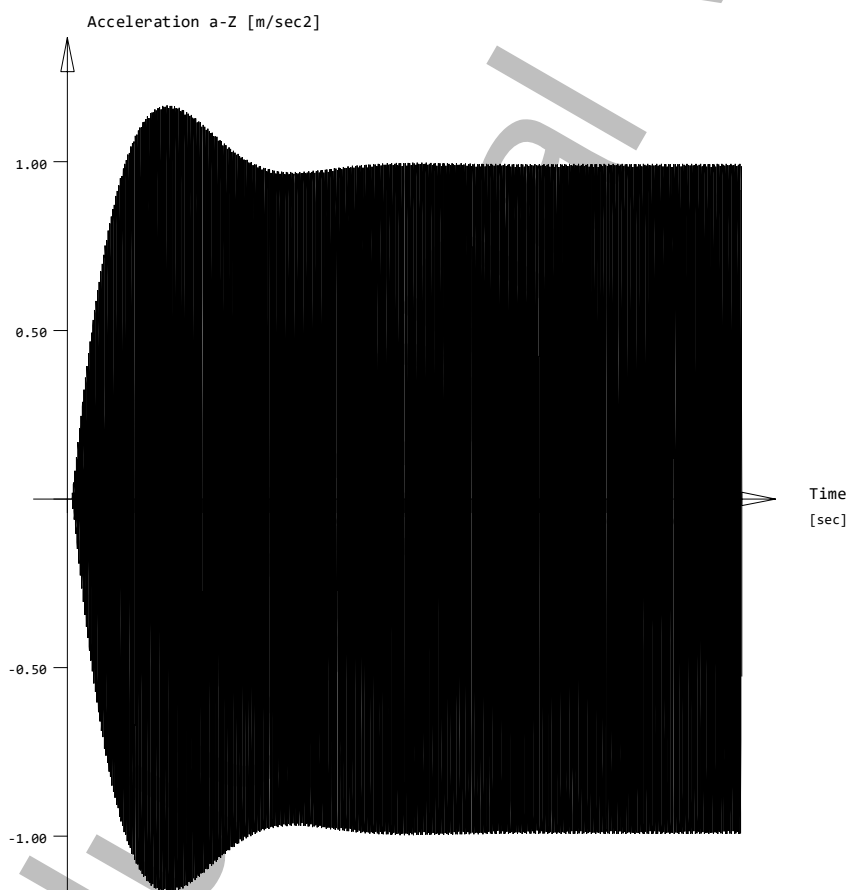
Results

Time history for Nodal accelerations

Node 901 (E 401) a-Z MINIMUM = -1.17 MAXIMUM = 1.17 [m/sec2]
stored in database for DYNR with identification no 704

Voldijkbrug_definitief_V1
Plotting results

Plotting results
Acceleration a-Z 901



Voldijkbrug_definitief_V1
TM2

System- and Control Information

Control Information

QUAD-elements with higher non conforming modes 3
QUAD-elements with all rotational degrees of freedom
Primary load case 20 t = 0.0000 [sec] +Opt:add_GS
Number of unknowns 43268
unknowns per node 6
Number of timesteps 10000
Time-step 0.0100
Printing intervall 1
damping factor A 2.529E-02
damping factor B 2.471E-04
Integr.Parameter beta 0.25
Integr.Parameter delta 0.50
Integr.Parameter theta 1.00

Groups

Grp	Option	CS	Factor	Rayleigh-A [1/sec]	Rayleigh-B [sec]	ξ [o/o]	Wind
0	FULL		1.000	0.000000	0.000000	0.00	
2	FULL		1.000	0.000000	0.000000	0.00	
4	FULL		1.000	0.000000	0.000000	0.00	
6	FULL		1.000	0.000000	0.000000	0.00	
7	FULL		1.000	0.000000	0.000000	0.00	
8	FULL		1.000	0.000000	0.000000	0.00	
9	FULL		1.000	0.000000	0.000000	0.00	
10	FULL		1.000	0.000000	0.000000	0.00	
11	FULL		1.000	0.000000	0.000000	0.00	
12	FULL		1.000	0.000000	0.000000	0.00	
13	FULL		1.000	0.000000	0.000000	0.00	
14	FULL		1.000	0.000000	0.000000	0.00	
15	FULL		1.000	0.000000	0.000000	0.00	
16	FULL		1.000	0.000000	0.000000	0.00	
17	FULL		1.000	0.000000	0.000000	0.00	
18	FULL		1.000	0.000000	0.000000	0.00	
19	FULL		1.000	0.000000	0.000000	0.00	
20	FULL		1.000	0.000000	0.000000	0.00	
21	FULL		1.000	0.000000	0.000000	0.00	
22	FULL		1.000	0.000000	0.000000	0.00	
23	FULL		1.000	0.000000	0.000000	0.00	
24	FULL		1.000	0.000000	0.000000	0.00	
25	FULL		1.000	0.000000	0.000000	0.00	
30	FULL		1.000	0.000000	0.000000	0.00	
41	FULL		1.000	0.000000	0.000000	0.00	
42	FULL		1.000	0.000000	0.000000	0.00	
43	FULL		1.000	0.000000	0.000000	0.00	
50	FULL		1.000	0.000000	0.000000	0.00	
61	FULL		1.000	0.000000	0.000000	0.00	
301	FULL		1.000	0.000000	0.000000	0.00	
302	FULL		1.000	0.000000	0.000000	0.00	
303	FULL		1.000	0.000000	0.000000	0.00	
304	FULL		1.000	0.000000	0.000000	0.00	
CS	construction stage			Rayleigh-B	stiffness proportional damping ratio		
Factor	Factor on stiffness			ξ	modal damping ratio		
Rayleigh-A	mass proportional damping ratio			Wind	options for dynamic wind loading		

Voldijkbrug_definitief_V1
TM2

Beam Elements

Finite beam elements without intermediate sections
Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam
Primary load case 20

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	460.987	460.987	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.000	-0.002	5.016	7.179E+03	-2.750E-01	3.208E+00	
				-2.750E-01	6.333E+05	5.835E-01	
				3.208E+00	5.835E-01	6.315E+05	
active ¹	455.087	457.937	460.987	4.279E+01	5.873E+01	3.563E+01	-
	S[m] ²			RM(S) ³			
	0.001	0.018	5.016	7.241E+03	-2.870E-01	4.793E-01	
				-2.870E-01	6.333E+05	-4.413E+01	
				4.793E-01	-4.413E+01	6.261E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 305

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00104	0.21939	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	-1.6	0.00	0.00	0.00	

Parameter of System of Equations

Number of unknowns 43268 (Direct sparse Gauss-Solver)
Total entries 1065492
Total entries after fill in 4763489
Mass matrix 268618 (consistent), incl. rotational masses
Damping matrix 1065313 (consistent)

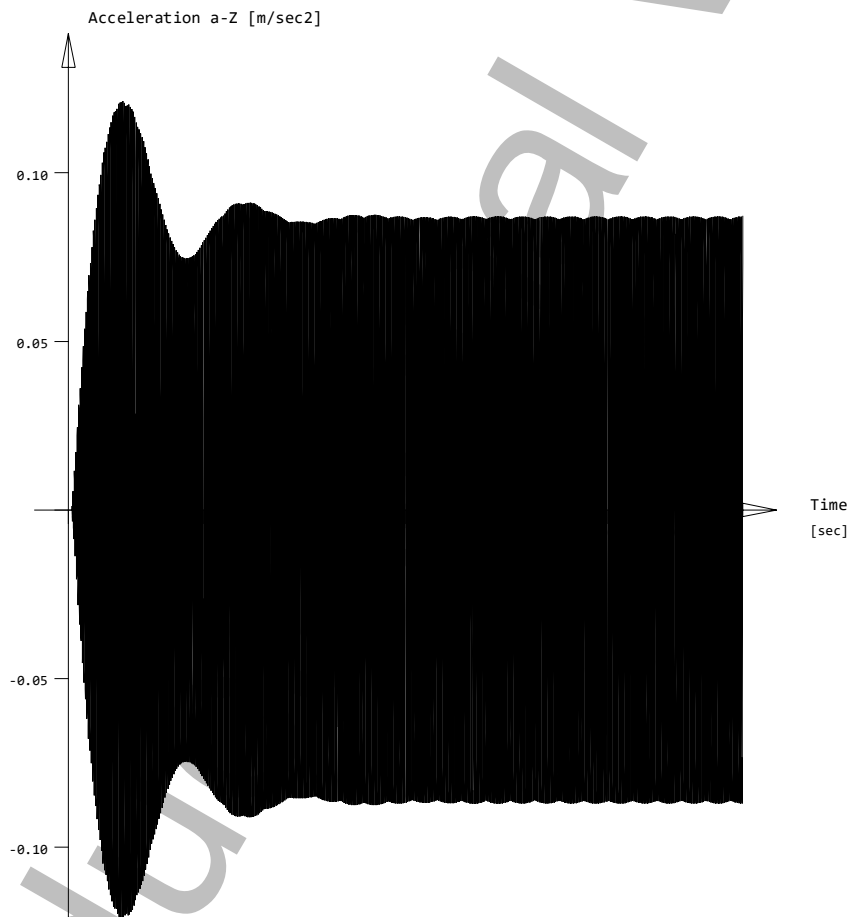
Results

Time history for Nodal accelerations

Node 1301 (E 1959) a-Z MINIMUM = -0.12 MAXIMUM = 0.12 [m/sec2]
stored in database for DYNR with identification no 705

Voldijkbrug_definitief_V1
Plotting results

Plotting results
Acceleration a-Z 1301



C.2 Optimised design - ULS & direct time integration

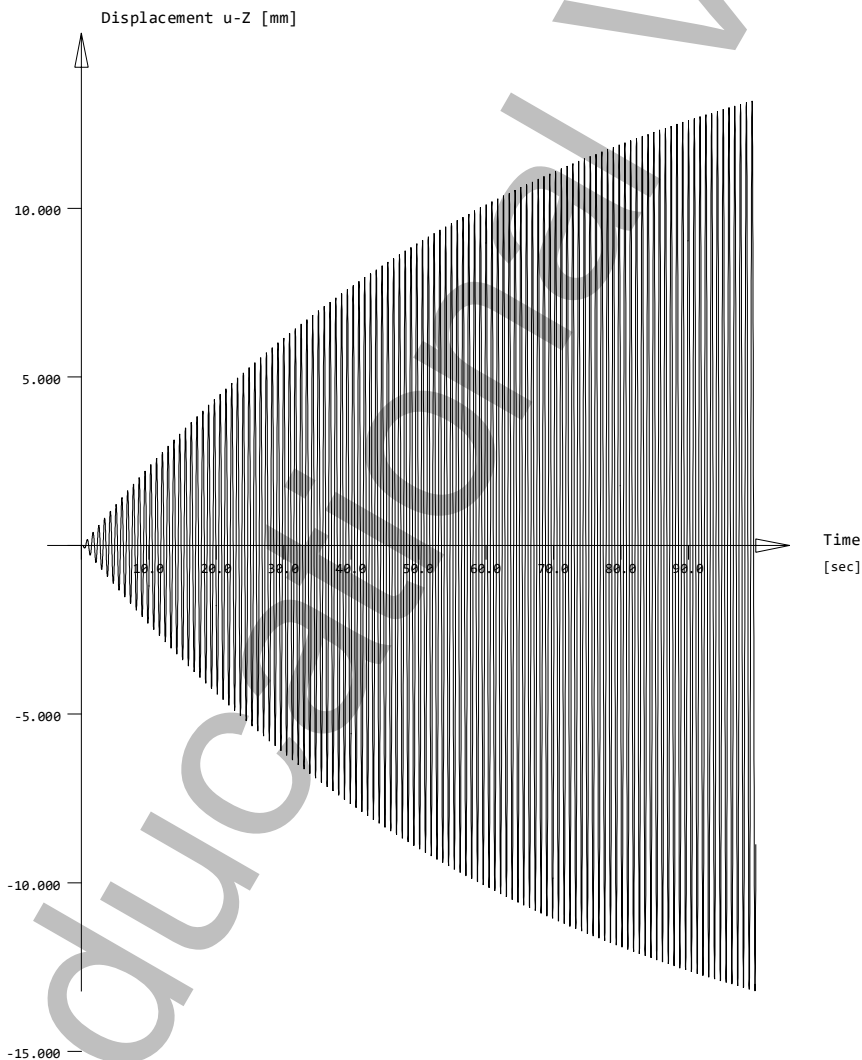
Elastic Stress Check

Maximum Utilisation Level

		N	Vy	Vz	My	Mz	Mtp	Mts	Mb	Ncr	SCL	Total
		σ -x	σ +x	τ	σ -v	σ -s	σ -dyn	As-l	As-v	crack		
Section 6		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	0.987
Primary Cable - D 72		0.000	0.987	0.000	0.987	-	-	-	-	-	-	
Section 7		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	0.251
Secondary Cable - D 36		0.000	0.251	0.000	0.251	-	-	-	-	-	-	
Total		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-	-	0.987
		0.000	0.987	0.000	0.987	-	-	-	-	-	-	
N	normal force			τ	shear stress							
Vy,Vz	shear force			σ -v	principal or von Mises stress							
My,Mz	bending			σ -s	stress in reinforcements							
Mtp,Mts	torsion (p)primary and (s)econdary			σ -dyn	stress range							
Mb	warping moment			As-l	longitudinal reinforcements							
Ncr	flexural buckling			As-v	transverse reinforcements or concrete shear strength							
SCL	cross-section class			crack	crack width							
σ -x	longitud. compressive stress			Total	most unfavorable utilisation for all checks							
σ +x	longitud. tensile stress											

Plotting results

Plotting results
Displacement u-Z 3000



Plotting results

reduced scale factor 0.917

Test according to EC

System- and Control Information

Control Information

QUAD-elements with higher non conforming modes 3
 QUAD-elements with all rotational degrees of freedom
 Primary load case 40 t = 0.0000 [sec] +Opt:add_GS
 Number of unknowns 22230
 unknowns per node 6
 Number of timesteps 10000
 Time-step 0.0100
 Printing intervall 1
 damping factor A 2.685E-02
 damping factor B 1.490E-04
 Integr.Parameter beta 0.25
 Integr.Parameter delta 0.50
 Integr.Parameter theta 1.00

Groups

Grp	Option	CS	Factor	Rayleigh-A [1/sec]	Rayleigh-B [sec]	ξ [o/o]	Wind
0	FULL		1.000	0.000000	0.000000	0.00	
2	FULL		1.000	0.000000	0.000000	0.00	
4	FULL		1.000	0.000000	0.000000	0.00	
5	FULL		1.000	0.000000	0.000000	0.00	
6	FULL		1.000	0.000000	0.000000	0.00	
8	FULL		1.000	0.000000	0.000000	0.00	
21	FULL		1.000	0.000000	0.000000	0.00	
22	FULL		1.000	0.000000	0.000000	0.00	
23	FULL		1.000	0.000000	0.000000	0.00	
24	FULL		1.000	0.000000	0.000000	0.00	
25	FULL		1.000	0.000000	0.000000	0.00	
30	FULL		1.000	0.000000	0.000000	0.00	
41	FULL		1.000	0.000000	0.000000	0.00	
42	FULL		1.000	0.000000	0.000000	0.00	
43	FULL		1.000	0.000000	0.000000	0.00	
61	FULL		1.000	0.000000	0.000000	0.00	
110	FULL		1.000	0.000000	0.000000	0.00	
111	FULL		1.000	0.000000	0.000000	0.00	
112	FULL		1.000	0.000000	0.000000	0.00	
113	FULL		1.000	0.000000	0.000000	0.00	
114	FULL		1.000	0.000000	0.000000	0.00	
115	FULL		1.000	0.000000	0.000000	0.00	
116	FULL		1.000	0.000000	0.000000	0.00	
117	FULL		1.000	0.000000	0.000000	0.00	
118	FULL		1.000	0.000000	0.000000	0.00	
119	FULL		1.000	0.000000	0.000000	0.00	
120	FULL		1.000	0.000000	0.000000	0.00	
301	FULL		1.000	0.000000	0.000000	0.00	
302	FULL		1.000	0.000000	0.000000	0.00	
303	FULL		1.000	0.000000	0.000000	0.00	
304	FULL		1.000	0.000000	0.000000	0.00	
CS		construction stage		Rayleigh-B		stiffness proportional damping ratio	
Factor		Factor on stiffness		ξ		modal damping ratio	
Rayleigh-A		mass proportional damping ratio		Wind		options for dynamic wind loading	

Beam Elements

Finite beam elements without intermediate sections
 Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam

Test according to EC

Beam Elements

Primary load case 40

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	417.357	417.357	417.357	4.111E+01	5.664E+01	3.097E+01	-
	S[m] ²			RM(S) ³			
	0.001	-0.002	4.490	4.662E+03	1.278E-03	1.166E+00	
				1.278E-03	5.802E+05	6.994E-01	
				1.166E+00	6.994E-01	5.795E+05	
active ¹	411.457	414.269	417.319	4.111E+01	5.664E+01	3.097E+01	-
	S[m] ²			RM(S) ³			
	0.003	0.020	4.490	4.707E+03	-2.350E-02	-1.296E+00	
				-2.350E-02	5.801E+05	-3.920E+01	
				-1.296E+00	-3.920E+01	5.740E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 301

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00356	0.46751	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	-0.7	0.80	0.00	0.00	

Loads on kinematic constraints have been transferred to their master

Parameter of System of Equations

Number of unknowns 22230 (Direct sparse Gauss-Solver)
 Total entries 518538
 Total entries after fill in 1781586
 Mass matrix 120897 (consistent), incl. rotational masses
 Damping matrix 518522 (consistent)

Results

Time history for Nodal accelerations

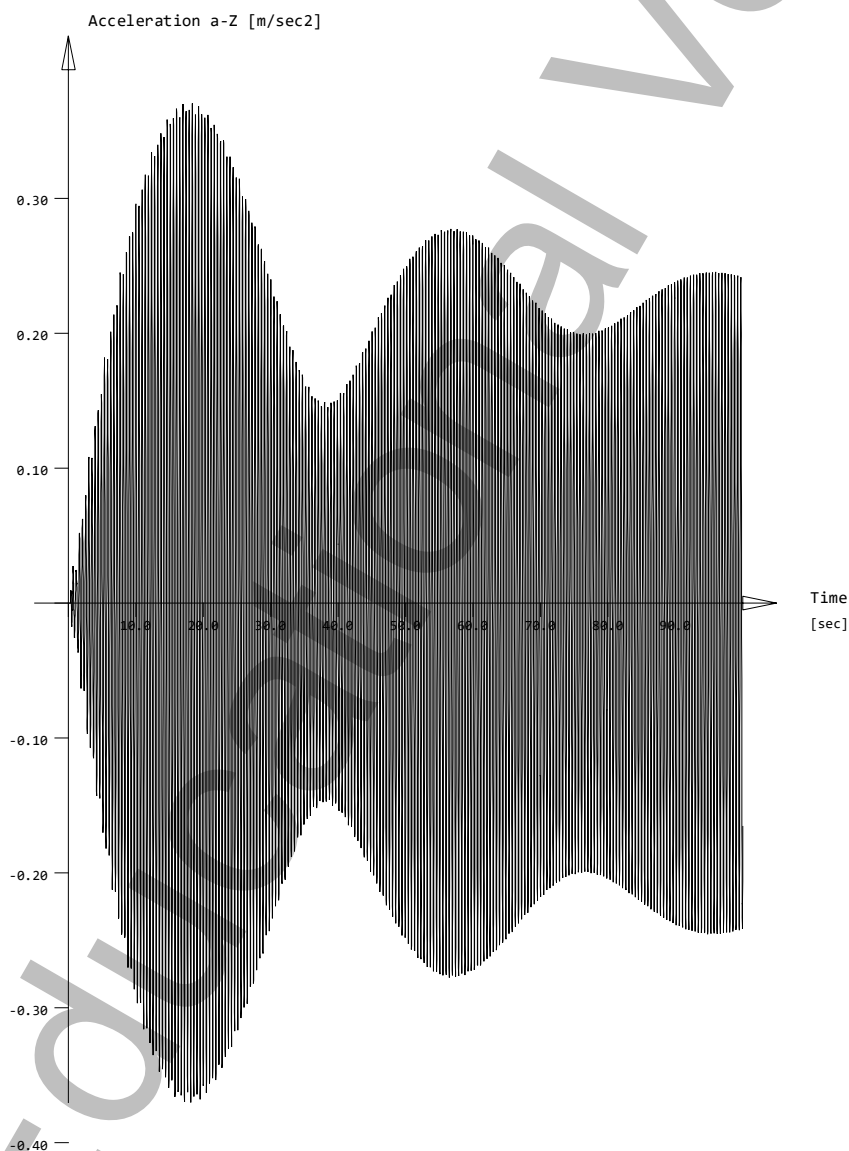
Node 3001 (E 2643) a-Z MINIMUM = -0.37 MAXIMUM = 0.37 [m/sec2]
 stored in database for DYNR with identification no 3001

Time history for Nodal Displacements

Node 3001 (E 2643) u-Z MINIMUM = -2.016 MAXIMUM = 2.014 [mm]
 stored in database for DYNR with identification no 4001

Plotting results

Plotting results
Acceleration a-Z 3001

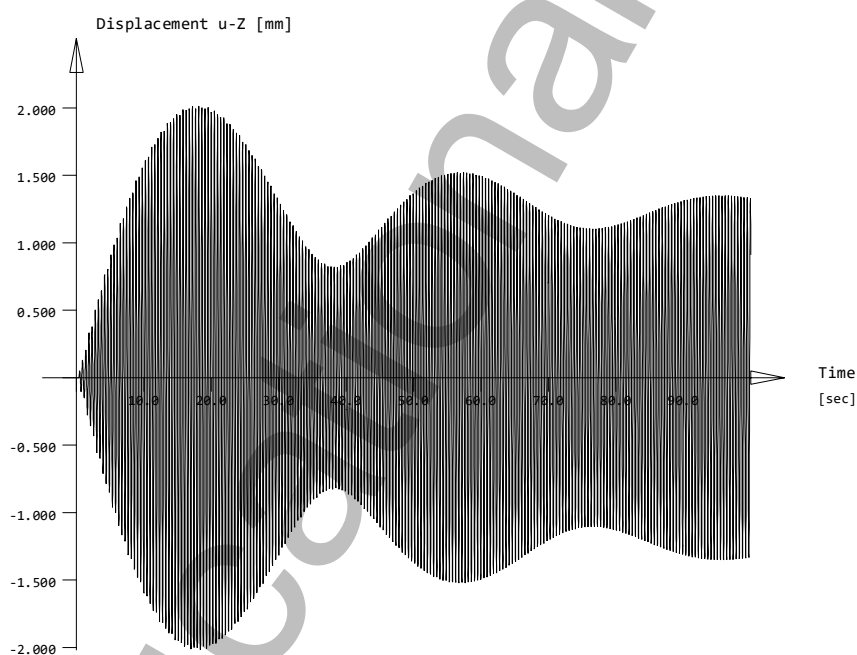


Plotting results

reduced scale factor 0.917

Plotting results

Plotting results
Displacement u-Z 3001



Test according to EC

System- and Control Information

Control Information

QUAD-elements with higher non conforming modes 3
 QUAD-elements with all rotational degrees of freedom
 Primary load case 40 t = 0.0000 [sec] +Opt:add_GS
 Number of unknowns 22230
 unknowns per node 6
 Number of timesteps 10000
 Time-step 0.0100
 Printing intervall 1
 damping factor A 3.203E-02
 damping factor B 1.249E-04
 Integr.Parameter beta 0.25
 Integr.Parameter delta 0.50
 Integr.Parameter theta 1.00

Groups

Grp	Option	CS	Factor	Rayleigh-A [1/sec]	Rayleigh-B [sec]	ξ [o/o]	Wind
0	FULL		1.000	0.000000	0.000000	0.00	
2	FULL		1.000	0.000000	0.000000	0.00	
4	FULL		1.000	0.000000	0.000000	0.00	
5	FULL		1.000	0.000000	0.000000	0.00	
6	FULL		1.000	0.000000	0.000000	0.00	
8	FULL		1.000	0.000000	0.000000	0.00	
21	FULL		1.000	0.000000	0.000000	0.00	
22	FULL		1.000	0.000000	0.000000	0.00	
23	FULL		1.000	0.000000	0.000000	0.00	
24	FULL		1.000	0.000000	0.000000	0.00	
25	FULL		1.000	0.000000	0.000000	0.00	
30	FULL		1.000	0.000000	0.000000	0.00	
41	FULL		1.000	0.000000	0.000000	0.00	
42	FULL		1.000	0.000000	0.000000	0.00	
43	FULL		1.000	0.000000	0.000000	0.00	
61	FULL		1.000	0.000000	0.000000	0.00	
110	FULL		1.000	0.000000	0.000000	0.00	
111	FULL		1.000	0.000000	0.000000	0.00	
112	FULL		1.000	0.000000	0.000000	0.00	
113	FULL		1.000	0.000000	0.000000	0.00	
114	FULL		1.000	0.000000	0.000000	0.00	
115	FULL		1.000	0.000000	0.000000	0.00	
116	FULL		1.000	0.000000	0.000000	0.00	
117	FULL		1.000	0.000000	0.000000	0.00	
118	FULL		1.000	0.000000	0.000000	0.00	
119	FULL		1.000	0.000000	0.000000	0.00	
120	FULL		1.000	0.000000	0.000000	0.00	
301	FULL		1.000	0.000000	0.000000	0.00	
302	FULL		1.000	0.000000	0.000000	0.00	
303	FULL		1.000	0.000000	0.000000	0.00	
304	FULL		1.000	0.000000	0.000000	0.00	
CS		construction stage		Rayleigh-B		stiffness proportional damping ratio	
Factor		Factor on stiffness		ξ		modal damping ratio	
Rayleigh-A		mass proportional damping ratio		Wind		options for dynamic wind loading	

Beam Elements

Finite beam elements without intermediate sections
 Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam

Test according to EC

Beam Elements

Primary load case 40

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	417.357	417.357	417.357	4.111E+01	5.664E+01	3.097E+01	-
	S[m] ²			RM(S) ³			
	0.001	-0.002	4.490	4.662E+03	1.278E-03	1.166E+00	
				1.278E-03	5.802E+05	6.994E-01	
				1.166E+00	6.994E-01	5.795E+05	
active ¹	411.457	414.269	417.319	4.111E+01	5.664E+01	3.097E+01	-
	S[m] ²			RM(S) ³			
	0.003	0.020	4.490	4.707E+03	-2.350E-02	-1.296E+00	
				-2.350E-02	5.801E+05	-3.920E+01	
				-1.296E+00	-3.920E+01	5.740E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 302

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00104	0.39231	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	-0.1	0.00	0.00	0.00	

Loads on kinematic constraints have been transferred to their master

Parameter of System of Equations

Number of unknowns 22230 (Direct sparse Gauss-Solver)
 Total entries 518538
 Total entries after fill in 1781586
 Mass matrix 120897 (consistent), incl. rotational masses
 Damping matrix 518522 (consistent)

Results

Time history for Nodal accelerations

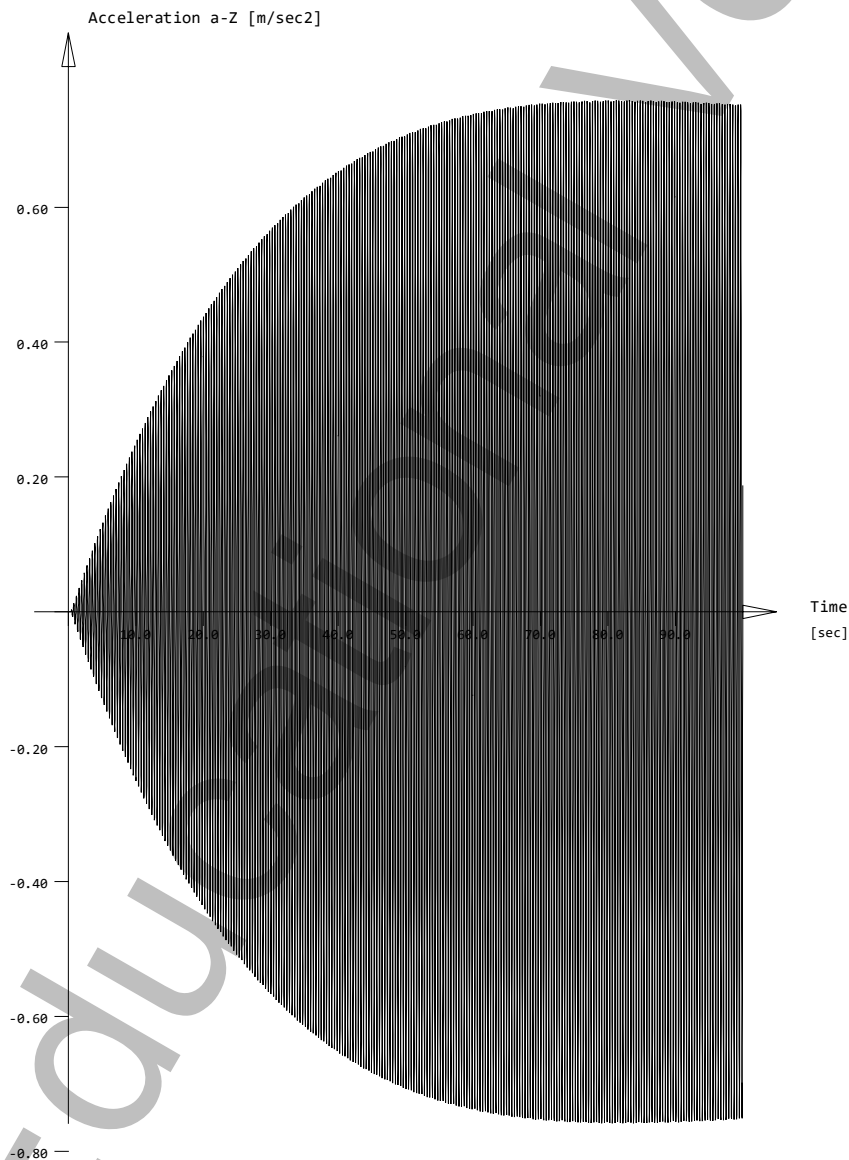
Node 3002 (E 2649) a-Z MINIMUM = -0.76 MAXIMUM = 0.76 [m/sec2]
 stored in database for DYNR with identification no 3002

Time history for Nodal Displacements

Node 3002 (E 2649) u-Z MINIMUM = -2.945 MAXIMUM = 2.946 [mm]
 stored in database for DYNR with identification no 4002

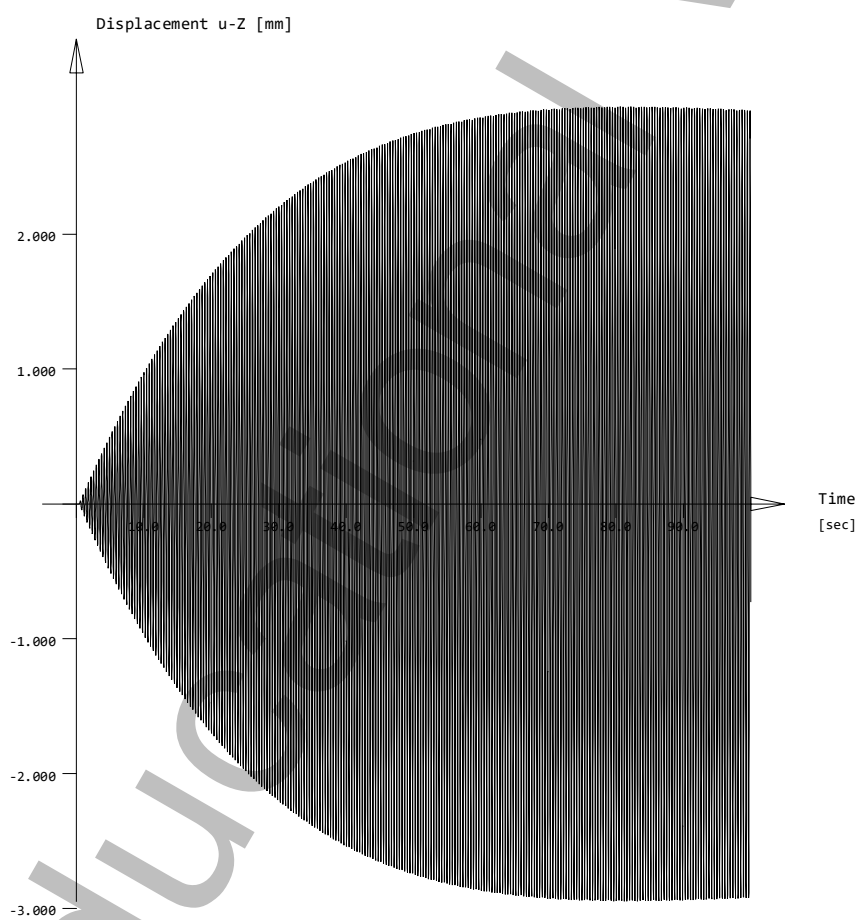
Plotting results

Plotting results
Acceleration a-Z 3002



Plotting results

Plotting results
Displacement u-Z 3002



Test according to EC

System- and Control Information

Control Information

QUAD-elements with higher non conforming modes 3
 QUAD-elements with all rotational degrees of freedom
 Primary load case 40 t = 0.0000 [sec] +Opt:add_GS
 Number of unknowns 22230
 unknowns per node 6
 Number of timesteps 10000
 Time-step 0.0100
 Printing intervall 1
 damping factor A 3.759E-02
 damping factor B 1.064E-04
 Integr.Parameter beta 0.25
 Integr.Parameter delta 0.50
 Integr.Parameter theta 1.00

Groups

Grp	Option	CS	Factor	Rayleigh-A [1/sec]	Rayleigh-B [sec]	ξ [o/o]	Wind
0	FULL		1.000	0.000000	0.000000	0.00	
2	FULL		1.000	0.000000	0.000000	0.00	
4	FULL		1.000	0.000000	0.000000	0.00	
5	FULL		1.000	0.000000	0.000000	0.00	
6	FULL		1.000	0.000000	0.000000	0.00	
8	FULL		1.000	0.000000	0.000000	0.00	
21	FULL		1.000	0.000000	0.000000	0.00	
22	FULL		1.000	0.000000	0.000000	0.00	
23	FULL		1.000	0.000000	0.000000	0.00	
24	FULL		1.000	0.000000	0.000000	0.00	
25	FULL		1.000	0.000000	0.000000	0.00	
30	FULL		1.000	0.000000	0.000000	0.00	
41	FULL		1.000	0.000000	0.000000	0.00	
42	FULL		1.000	0.000000	0.000000	0.00	
43	FULL		1.000	0.000000	0.000000	0.00	
61	FULL		1.000	0.000000	0.000000	0.00	
110	FULL		1.000	0.000000	0.000000	0.00	
111	FULL		1.000	0.000000	0.000000	0.00	
112	FULL		1.000	0.000000	0.000000	0.00	
113	FULL		1.000	0.000000	0.000000	0.00	
114	FULL		1.000	0.000000	0.000000	0.00	
115	FULL		1.000	0.000000	0.000000	0.00	
116	FULL		1.000	0.000000	0.000000	0.00	
117	FULL		1.000	0.000000	0.000000	0.00	
118	FULL		1.000	0.000000	0.000000	0.00	
119	FULL		1.000	0.000000	0.000000	0.00	
120	FULL		1.000	0.000000	0.000000	0.00	
301	FULL		1.000	0.000000	0.000000	0.00	
302	FULL		1.000	0.000000	0.000000	0.00	
303	FULL		1.000	0.000000	0.000000	0.00	
304	FULL		1.000	0.000000	0.000000	0.00	
CS		construction stage		Rayleigh-B		stiffness proportional damping ratio	
Factor		Factor on stiffness		ξ		modal damping ratio	
Rayleigh-A		mass proportional damping ratio		Wind		options for dynamic wind loading	

Beam Elements

Finite beam elements without intermediate sections
 Shear deformations accounted for with nonconforming SOFiSTiK-Timoshenko beam

Test according to EC

Beam Elements

Primary load case 40

Sum of masses and mass moments of inertia

Node	TM			RM			RMB
	X[t]	Y[t]	Z[t]	X[tm2]	Y[tm2]	Z[tm2]	
total ¹	417.357	417.357	417.357	4.111E+01	5.664E+01	3.097E+01	-
	S[m] ²			RM(S) ³			
	0.001	-0.002	4.490	4.662E+03	1.278E-03	1.166E+00	
				1.278E-03	5.802E+05	6.994E-01	
				1.166E+00	6.994E-01	5.795E+05	
active ¹	411.457	414.269	417.319	4.111E+01	5.664E+01	3.097E+01	-
	S[m] ²			RM(S) ³			
	0.003	0.020	4.490	4.707E+03	-2.350E-02	-1.296E+00	
				-2.350E-02	5.801E+05	-3.920E+01	
				-1.296E+00	-3.920E+01	5.740E+05	
¹ sum of the total and the active nodal masses							
² coordinates of the center of gravity							
³ 3x3 rotational mass matrix at the center of gravity							
TM translational masses in X-, Y- and Z-direction							
RM rotational masses about X-, Y- and Z-axis							
RMB warping mass							

Processing

Load Cases

-- Loadcase 303

	amplitude	period	phase	T-min	T-max	S[-]	
	0.00104	0.33411	0.00000				
Node	PX	PY	PZ	MX	MY	MZ	Mb
	[kN]	[kN]	[kN]	[kNm]	[kNm]	[kNm]	[kNm2]
sum	0.0	0.0	0.1	0.63	0.00	0.00	

Loads on kinematic constraints have been transferred to their master

Parameter of System of Equations

Number of unknowns 22230 (Direct sparse Gauss-Solver)
 Total entries 518538
 Total entries after fill in 1781586
 Mass matrix 120897 (consistent), incl. rotational masses
 Damping matrix 518522 (consistent)

Results

Time history for Nodal accelerations

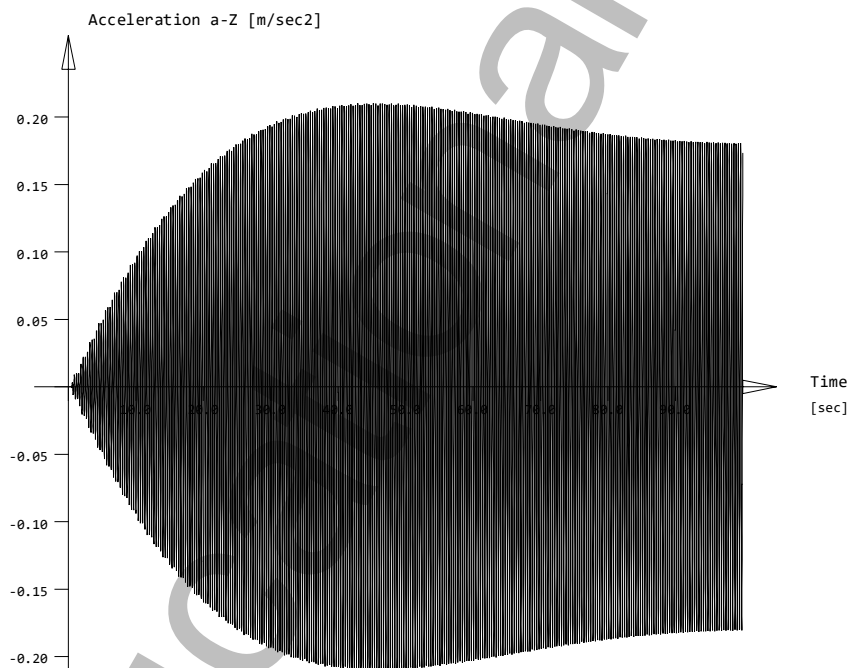
Node 3003 (E 2655) a-Z MINIMUM = -0.21 MAXIMUM = 0.21 [m/sec2]
 stored in database for DYNR with identification no 3003

Time history for Nodal Displacements

Node 3003 (E 2655) u-Z MINIMUM = -0.592 MAXIMUM = 0.592 [mm]
 stored in database for DYNR with identification no 4003

Plotting results

Plotting results
Acceleration a-Z 3003



Plotting results

Plotting results
Displacement u-Z 3003

