

PERFORMANCE ENHANCEMENT OF ACTIVE STRUCTURES DURING SERVICE LIVES

THÈSE N° 2750 (2003)

PRÉSENTÉE À LA FACULTÉ ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT

SECTION DE GÉNIE CIVIL

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Bernd DOMER

Dipl.-Ing., Ruhr-Universität Bochum, Allemagne
et de nationalité allemande

acceptée sur proposition du jury:

Prof. I. Smith, directeur de thèse
Prof. J. Giovanola, rapporteur
Prof. L. Hovestadt, rapporteur
Prof. C. Könke, rapporteur
Dr B. Raphael, rapporteur

Lausanne, EPFL
2003

Acknowledgements

First, I would like to thank my supervisor, Prof. Ian F. C. Smith, for giving me the opportunity to work at the applied computing and mechanics laboratory (IMAC). His guidance and encouragement, as well as the discussions during the preparation of several publications, have been a source of constant inspiration throughout this research.

I thank the members of the examination jury for the time they spent reading and commenting this thesis. Discussions during intermediate stages of this work gave important directions. Prof. Jacques Giovanola from STI-IPR-LCSM, EPF-Lausanne, Prof. Ludger Hovestadt from CAAD-ARCH., ETH Zürich, Prof. Carsten Könke from ISM, Professur für Baustatik, Bauhaus Universität Weimar, Dr. Benny Raphael from ENAC-IS-IMAC, EPF-Lausanne and Prof. André-Gilles Dumont, ENAC-ICARE-LAVOC, EPF-Lausanne as President of the Jury.

It has been a pleasure to work with Dr. Benny Raphael. His open attitude during discussions and his presence at IMAC have been highly stimulating. I never left his office without having learnt something.

I would like to underline the important impact student projects had on this thesis. In particular: Vikram Lalit from the Indian Institute of Technology (IIT) Delhi with whom I worked on neural networks, Gaurav Gupta, IIT Kanpur, who helped me to adapt and test genetic algorithms for the control task, Lukas Bieri, EPFL, who did the beta-testing of the Tensegrity Structure Analysis and Control Software and Antje Landschulz, Technische Universität Darmstadt, who evaluated the first case-base reasoning prototype on the three-module structure.

There was an unofficial thesis committee that commented on the very first draft of this document. Dr. Micheal LeVine from the Brookhaven National Laboratory, Long Island, USA, Dr. Andreas Pfeiffer and Dr. Salvatore Mele, both from CERN Geneva, have to be acknowledged for taking their task seriously and straightening out ambiguities. I would like to thank Jean Louis Guignard from IMAC for sharing his experience with me to improve the quality of the figures.

My thanks go to Dr. Kristina Shea, now university Lecturer at the department of engineering at Cambridge University, who launched together with Prof. Ian F. C. Smith the initial Swiss National Science Foundation project on intelligent structural control. It was Dr. Etienne Fest who constructed the full-scale active tensegrity structure. Without his achievements this thesis would just be based on theoretical considerations. This also gives me the opportunity to highlight the excellent collaboration with laboratory technicians at IMAC. I was very grateful for the presence of Raymond Délez, Patrice Gallay, Charles Gilliard and Manuel Pascual.

I am indebted to Marco Viviani who supported me during a critical phases of the research in the creation of the case-base used in the latter stages of the project. Dr. Pierino Lestuzzi is thanked for dedicating time and intellect in giving useful comments whenever I was preparing presentations. Thanks to Tim Schumacher for his final proof reading. It flattered me that he asked for a copy afterwards.

A thesis like this would not have been possible without the creative, communicative and competent team spirit present at IMAC. This spirit is kept alive to a big extent, thanks to people like Francine Laferrière, Dr. Pascal Kronenberg and Yvan Robert-Nicoud.

The Swiss Government through the Swiss National Science Foundation contributed to this project by the grant with the number No. 2000-061756. The support is greatly acknowledged.

Last, but not least, I would like to express my gratitude to my parents for their encouragement and faith in me. And to Petra, my wife, for her continuous love and support.

Abstract

This thesis describes a successful application of advanced computational methods to tasks in the field of active structural control. The control-task involves finding good control movements for a highly coupled, non-linear structure. It is demonstrated how these methods improve the accuracy of the analytical model. Also, stochastic search techniques are compared for the same task. Furthermore, the performance of the system can be enhanced during service life by storing, retrieving and adapting good solutions.

The structure studied, a Tensegrity, is a special type of cable structure. Tensegrities stimulate the imagination of artists, researchers and engineers. Varying the amount of selfstress changes structural shape as well as the load-bearing capacity. They offer unique applications, as deployable structures in the context of aerospace applications and more generally, as actively controlled structures. However, the non-linear behavior of tensegrities is difficult to model.

Aspects of this work involve subjects such as tensegrity structures, active structural control, search algorithms and artificial intelligence. The focus of this thesis is on the last two subjects. This work demonstrates how advanced computing techniques can be used in order to increase solution quality. A hybrid approach, employing neural networks, increases the accuracy of the analytical model that is employed for simulating tensegrity structures. A comparison of three stochastic search techniques shows that computational time, first estimated to take centuries when adapting a “brute-force” approach, can be reduced to hours.

Case-based reasoning (CBR) is used for a further tenfold decrease in computation time. The time needed to find good control solutions decreased from hours, when stochastic search is used, to minutes with CBR. CBR also provides possibilities for improving performance over service life. Successfully solved situations are stored as cases in a case-base. In new situations, a case close to the new situation is retrieved and then adapted. By storing additional cases, the system is able to retrieve better cases for adaptation. With increasing case-base size, adaptation time decreases.

The combination of these techniques has much potential for improving the performance of complex structures during service lives. Results should contribute to the development of innovative structural solutions. Finally, it is expected that the findings in this thesis will become points of departure for subsequent studies.

Keywords: structural control, active control, intelligent structures, stochastic search, non-linear analysis, artificial intelligence, tensegrity structures, case based reasoning

Version abrégée

Cette thèse décrit une application réussie des techniques d'informatique avancées au domaine du contrôle des structures actives. La tâche de contrôle consiste à trouver les bons mouvements pour une structure non-linéaire hautement couplée. On y démontre comment ces méthodes améliorent la précision du modèle analytique. En outre, différentes méthodes de recherche stochastique sont comparées pour une même tâche. De plus, la performance du système peut être améliorée durant son exploitation, par une mémorisation des informations, leur récupération et leur adaptation pour de bonnes solutions.

La structure étudiée, une tensegrité, est un type spécial de structures en câbles. Les tensegrités stimulent l'imagination des artistes, des chercheurs et des ingénieurs. En variant son état d'auto contrainte, une telle structure change sa forme ainsi que sa résistance. Elles permettent des applications originales, tel des structures déployables dans le domaine aérospatial ou plus généralement des structures actives. Cependant, le comportement non-linéaire de ce type de structure est difficile à modéliser.

Les aspects de ce travail englobent outre les structures tensegrités, des sujets tel que le contrôle actif, les algorithmes de recherche et l'intelligence artificielle. La thèse se concentre plus particulièrement sur les deux derniers sujets. On y prouve que les techniques d'informatique avancées peuvent être utilisées pour améliorer la qualité des solutions. Une approche hybride, utilisant des réseaux de neurones, améliore la précision du modèle analytique utilisé pour la modélisation des structures tensegrités. Une comparaison entre trois techniques de recherche stochastique montre que le temps de calcul, estimé en premier lieu à plusieurs siècles par une approche de « force brute », peut être réduit à quelques heures.

Le raisonnement par cas (Case-Based Reasoning, CBR) est utilisé pour une diminution du temps de traitement. Le temps nécessaire pour trouver une bonne solution, évalué en heures par une recherche stochastique, diminue à quelques minutes par la méthode du raisonnement par cas. Le CBR fournit des possibilités d'amélioration des performances durant la vie de service. Les situations résolues avec succès sont stockées dans une base de cas. Ainsi, dans des situations nouvelles, un cas proche de cette situation peut être réutilisé et adapté. En mettant en mémoire des cas supplémentaires, le système est capable de retrouver les meilleurs cas et de les adapter. Grâce à une augmentation de la taille de la base de cas, le temps d'adaptation diminue.

La combinaison de ces techniques a un grand potentiel pour augmenter les performances des structures complexes pour leur état de service. Les résultats devraient contribuer au développement de solutions de structures innovantes. Finalement, les conclusions de cette thèse devront servir de points de départ pour d'autres projets de recherche.

Zusammenfassung

Die vorliegende Arbeit beschreibt die erfolgreiche Anwendung fortschrittlicher Computermethoden auf dem Gebiet der aktiven Kontrolle von Tragwerken. Die Kontrollaufgabe bestand in der Suche nach optimalen Kommandos für eine stark gekoppelte, geometrisch nicht lineare Struktur. Es wird demonstriert, wie diese Methoden die Genauigkeit des analytischen Modells verbessern. Verschiedene Suchtechniken werden miteinander verglichen. Darüber hinaus kann die Leistung des Systems während seiner Lebensdauer durch Speicherung, Abruf und Anpassung guter Kontrollkommandos gesteigert werden.

Tensegrities sind eine spezielle Art von Seilnetzkonstruktionen. Sie inspirieren Künstler, Forscher und Ingenieure gleichermaßen. Variationen der Eigenspannung verändern ihre Gestalt sowie die Lastkapazität. Sie eröffnen damit einzigartige Möglichkeiten als faltbare Strukturen für Anwendungen im All, sowie als aktiv kontrollierte Tragstrukturen. Ihr nicht lineares Verhalten ist jedoch schwer zu modellieren.

Diese Arbeit behandelt die Forschungsgebiete Tensegrity-Strukturen, aktive Kontrolle von Tragwerken, Suchalgorithmen und wissensbasierte Systeme. Das Hauptaugenmerk ist auf die letzten beiden Themen gerichtet. Es wird demonstriert, wie fortschrittliche Computermethoden zur Verbesserung der Ergebnisqualität eingesetzt werden können. Ein hybrider Ansatz, welcher neurale Netzwerke benutzt, erhöht die Präzision der zur Modellierung von Tensegrities verwendeten Tragwerkmodells. Drei stochastische Suchtechniken werden auf ihre Eignung für die Suche nach Kontrollkommandos miteinander verglichen. Die Rechenzeit, welche bei Einsatz eines „brute force“ Ansatzes zunächst auf Jahrhunderte abgeschätzt wurde, verringerte sich auf einige Stunden.

Fallbasiertes Schließen (Case-based reasoning, CBR) wird für weitere Leistungssteigerung verwendet. Die Rechenzeit, welche bei Anwendung von stochastischen Suchmethoden noch Stunden betrug, verringerte sich auf Minuten. CBR bietet darüber hinaus die Möglichkeit für Leistungssteigerungen während der Lebensdauer der Struktur. Erfolgreich gelöste Kontrollaufgaben werden als Fälle in einer Falldatenbank gespeichert. In neuen Situationen wird ein ähnlicher Fall in dieser Datenbank gesucht und angepasst. Mit dem Speichern von zusätzlichen Fällen in der Datenbank kann das System immer bessere Fälle zur Anpassung vorschlagen. Mit wachsender Größe der Datenbank verringert sich die zur Anpassung benötigte Zeit.

Die Kombination von unterschiedlichen Computermethoden zeigt Möglichkeiten zur Leistungssteigerung von komplexen Tragwerken während ihrer Lebensdauer. Ergebnisse sollten zur Entwicklung innovativer Strukturen beitragen. Es wird gehofft, daß diese Arbeit als Ausgangspunkte für weitere Forschung benutzt werden kann.

Abreveations

A	=	equilibrium matrix of cable structures
B	=	compatibility matrix of cable structures
J	=	number of non-constrained joints
q	=	number of mechanisms
r	=	rank of equilibrium matrix H
m	=	number of links (bars and cables) of cable structures
s	=	stress states of cable structures

A	=	area
E	=	module of elasticity
P	=	single load
w	=	deflection

C	=	damping matrix
d	=	vector of nodal displacements
δ	=	deformation
K, K^I	=	stiffness matrix
K^{NL}	=	non-linear stiffness matrix
L_m⁰	=	Initial length of a bar/link
L_m^(t+Δt)	=	Length of bar/link at time
M	=	mass matrix
P(t)	=	vector of nodal forces, varying with time
R	=	vector of residual forces
V	=	vector of nodal velocities
t	=	time
T_m⁰	=	Initial prestress in a link
T_m^(t+Δt)	=	Force in bar/link at time

Table of contents

1	Introduction.....	1
1.1	Context	1
1.2	Motivation	2
1.3	Objectives	3
1.4	Contributions	3
1.5	Document organization	4
2	Literature review.....	5
2.1	Tensile structures and tensegrities.....	5
2.1.1	Structural design of tensile structures	6
2.1.1.1	Form finding with experimental methods	6
2.1.1.2	Analytical form finding with the force density method	9
2.1.1.3	Static analysis of tensile structures: the dynamic relaxation method	10
2.1.2	Tensegrities	13
2.1.3	Structural particularities of tensegrity structures.....	15
2.1.4	Deployable tensegrities	17
2.2	Structural control and intelligent systems	18
2.2.1	Passive control.....	20
2.2.2	Active control	22
2.2.3	Semi-active control.....	24
2.2.4	Control of tensegrity systems	24
2.2.5	Intelligent structural control	25
2.3	Optimization and stochastic search	26
2.3.1	Simulated annealing	28
2.3.1.1	An introduction to simulated annealing	28
2.3.1.2	Schedule parameter setting.....	29
2.3.2	Probabilistic global search Lausanne (PGSL).....	31
2.3.2.1	An introduction to PGSL.....	31
2.3.2.2	Adjusting the parameters of PGSL.....	34
2.3.3	Genetic algorithms	35
2.3.3.1	An introduction to genetic algorithms.....	35
2.3.3.2	Choosing parameters for genetic algorithms.....	36
2.3.4	Comparison of different algorithms	37

2.4	Artificial intelligence methods for use in structural engineering	38
2.4.1	Neural Networks	39
2.4.2	Case-based reasoning	42
2.4.2.1	Case retrieval and adaptation	43
2.4.2.2	The utility-problem – case-based maintenance	45
2.4.3	Clustering	47
2.5	Conclusions	48
3	Computational model	49
3.1	Tensegrity structure at IMAC	49
3.1.1	Full scale tensegrity structure.....	49
3.1.2	Five-module structure	53
3.1.3	Measuring cable forces.....	53
3.2	Geometry and topology.....	55
3.3	Tensegrity structure analysis and control software (TSACS)	57
3.3.1	Module Festorder	58
3.3.2	Module Tensgraph.....	59
3.3.3	Module Dynarex.....	59
3.4	Improvement of the computational model.....	60
3.4.1	Motivation	60
3.4.2	Artificial neural network software applied.....	60
3.4.3	Test description	61
3.4.4	Enhancing the Accuracy of the Dynamic Relaxation Method.....	65
3.4.4.1	Using measured results.....	65
3.4.4.2	Using the average of the measured values	66
3.4.4.3	Using Jenkin’s hypercube for the selection of training data	68
3.5	Substituting the dynamic relaxation method completely with artificial neural networks.....	68
3.6	Testing the hybrid approach on the five-module structure.....	70
3.6.1	Obtaining new training data	70
3.6.2	Training of a new neural network	72
3.6.3	Testing robustness of the network.....	73
3.7	Discussion of results	75
4	Stochastic search algorithms for active structural control	77
4.1	Complexity of the control task	77
4.2	Control objective and constraints.....	79
4.3	Control system architecture	80
4.4	Test description	82
4.4.1	Test configuration	82

4.5	Comparison of stochastic search techniques.....	84
4.5.1	Simulated annealing	84
4.5.2	Probabilistic global search Lausanne (PGSL).....	85
4.5.3	Genetic algorithms	86
4.5.4	Results	87
4.5.4.1	Series 1	87
4.5.4.2	Series 2	88
4.5.4.3	Effect of the number of runs.....	92
4.6	Discussion of results	93
5	Improvement of structural performance over time.....	95
5.1	CBR-system.....	97
5.2	Prototype system for the three module structure.....	100
5.2.1	Case-bases for testing	100
5.2.2	Similarity measurement and retrieval.....	103
5.2.3	Tests performed on case bases	105
5.2.4	Comparing case-based reasoning with pure optimization.....	105
5.2.5	Size of the case base	106
5.2.6	Different search strategies for adaptation.....	107
5.2.7	Discussion of results.....	108
5.3	Applying results to the five module structure.....	108
5.3.1	Changing the control objective.....	108
5.3.2	Creation of a reference case-base	110
5.4	Evaluation of case-based reasoning	112
5.4.1	Similarity measurement.....	112
5.4.2	Performance increases through case-based reasoning.....	112
5.4.2.1	Test description	112
5.4.2.2	Series 1: Potential of CBR for performance improvement	114
5.4.2.3	Series 2: Take GA's advantage from being launched from a good control task.....	117
5.4.2.4	Series 3: Testing clustering to overcome maintenance problems	117
5.4.2.5	Series 4: Performance enhancement over time	119
5.5	Discussion of results	121
6	Conclusions and future directions.....	123
6.1	Conclusions	123
6.1.1	Introduction	123
6.1.2	Accuracy enhancement of the analytical model.....	123
6.1.3	Comparing stochastic search algorithm for structural control	124
6.1.4	Performance enhancement over service life.....	124
6.1.5	Summary of results.....	125
6.1.6	List of conclusions.....	126
6.2	Future directions	127
6.2.1	Study of the distance metric	127
6.2.2	System identification	127
6.2.3	Enhancing accuracy.....	128

6.2.4	Increase adaptation performance.....	128
6.2.5	Online training of an existing neural network.....	128
6.2.6	Using a neural network during control command search.....	128
6.2.7	Controlling space of possible solutions.....	129
6.2.8	Security mechanisms.....	129
6.2.9	Control mechanics of tensegrity systems	129
6.2.10	Multi-criteria control	130
6.2.11	Potential for fault-tolerance and self-repair	130
6.3	Transferability of results	130

1 Introduction

1.1 Context

The word “tensegrity” has been coined by Richard Buckminster Fuller (1962) who concatenated the words tensile and integrity. Tensegrity structures are lightweight. They consist of tension (cables) and compression (bars) elements. Tension is equilibrated by states of inner self-stress. This property distinguishes tensegrities from cable structures where tensile forces need to be anchored. Fuller observed that tensegrities are constructed as “*small islands of compression in a sea of tension*”. The structural principle is appealing because it combines efficient use of materials with aesthetical principles in an original way. Identifying the inventor of the tensegrity is difficult. Three names arise: Buckminster Fuller himself, the sculptor Kenneth Snelson (Snelson 2002) who worked with Fuller but left him, bitterly disappointed, and their European counterpart David Georges Emmerich (Emmerich 1964). Emmerich names the Russian constructivist Karl Ioganson as being the first to present a tensegrity-like sculpture, “Study in Balance” in 1921.

Over a long period, tensegrities have only been considered as objects of art and not as examples of useful structures. Sculptures have been erected based on heuristics. A sound analytic background of particular mechanical properties is provided by Pellegrino (1990) and Motro (1987). Applications to real structures exist. The design office “Geiger Engineers” (Gossen et al. 2002) designed and constructed tensile membrane structures, such as the Olympic Fencing Arena in Seoul. Nevertheless, these cable-domes are considered as cable nets with struts in rigid compression rings rather than pure tensegrities (Kawaguchi and Lu 2002).

Tensegrities are an area of interest for a growing community of researchers who are interested in static and dynamic behavior, deployable structures, shape control and, as it is the case for this work, determining whether performance of active shape control can increase over time by learning.

Controlling self-stress states of tensegrities provides the potential to control their shape and adapt to changing tasks and environments. Although structural control is considered to be an advanced structural engineering today, most efforts concentrate mainly on a single criterion and non-changing control objectives. Expressions rely on closed formulations of system characteristics, including those linking sensors and actuators. These formulations cannot be employed for the control of tensegrity structures since they exhibit highly coupled, nonlinear behavior. Tensegrities can be dismantled and reused for different occasions at new locations. Control objectives have to be adaptable to meet these requirements. This is not possible with conventional approaches. Methodologies proposed by research into artificial intelligence

have potential for application to these tasks. Coupling with conventional approaches enhances performance. This leads to systems that improve performance over time.

Research presented in this thesis is multidisciplinary. Domains of structural engineering, computer science and control technology are linked to enhance performance of active structures. More precisely, artificial intelligence methods will be applied to control the shape of a full-scale tensegrity structure.

1.2 Motivation

Transferring the tensegrity's structural principle into a framework that enhances performance over service life will result in new perspectives for structural engineering.

A study of the structural behavior of tensegrities is a challenge and an opportunity at the same time:

- Challenge: to investigate control of shape such that new applications in structural engineering become feasible
- Opportunity: to develop and apply new concepts in active structural control.

An initial challenge consists of describing mechanical properties of such structures correctly. Research gives insight into different approaches to transfer structural behavior into computational models. Although mechanics provides possibilities to build abstractions with any degree of theoretical precision, the relevance of these models to real structural behavior is questionable. Each additional parameter increases the complexity of the model. Interdependencies between parameters, as well as changing behavior over service lives, are difficult to account for. In the context of an engineering approach, models for complex structures should be the result of a compromise between practical usefulness and complexity.

A second challenge is to generate control commands for tensegrities. As already stated, their behavior is tightly coupled and non-linear. A first control algorithm has proven to successfully evaluate control commands directly for full-scale structures (Fest 2002). Commands are found by generation, analysis, and test strategies. Attempts to evaluate all possible solutions for a given objective to access the global minima fail almost immediately, as the complexity of this task leads to unacceptable computational costs. Intelligently sampling the space of possible solutions by stochastic search methods has potential to obtain good solutions in acceptable computational time. Stochastic search provides flexibility regarding control objectives and constraints.

Control commands found by search can be stored in a database for reuse in similar situations. Such systems become intelligent when explicit knowledge representations and reuse of previous loading responses lead to an automatic improvement of performance over time. Intelligent computational control will enhance the serviceability of structures and lead to systems that sense, react, learn and plan in uncertain environments.

Improving performance of the control system over time involves improvement of the computational model according to measured behavior and application of case-based reasoning for control command reuse. In addition to a general evaluation of the importance of artificial intelligence methods and intelligent control in civil engineering, this work will lead to new alternatives for structural engineering tasks, particularly those requiring lightweight, reusable and innovative structures.

1.3 Objectives

This thesis aims to contribute to the development of structures that can be intelligently controlled and adapted to changing control objectives and environments. A methodology for active structural control that finds control commands for complex tasks is developed, implemented and tested. More precisely, the following objectives have been formulated:

- Improvement of accuracy of the computational model used to simulate tensegrity structure behavior through training with measurements (application of neural networks)
- Evaluate the potential of stochastic search for finding good control commands and test different search methods for this task
- Improving performance of active structures over service lives through application of case-based reasoning (CBR) and through addressing the issue of case-base maintenance.
- Testing and refinement of the control software through numerical simulation and laboratory tests.

Meeting these objectives contributes to progress toward development of new types of structures, thereby, providing engineers with design alternatives that previously have not been possible.

1.4 Contributions

A novel approach to modeling tensegrity structures in service is presented. The methodology is the result of a compromise between advantages of an explicit structural knowledge incorporated in a mechanical model and a calibrated “black box” system that increases accuracy through training using feedback.

Testing stochastic search methods for active structural control of a non-linear coupled system indicates advantages and disadvantages of each method. Search itself can be employed with changing control objectives and thereby provides flexibility.

Combining artificial intelligence methods in a framework for active structural control helps to develop systems that will improve performance over time. Other fields of suitable applications are proposed. This could lead to a better understanding of when such methods add value to engineering tasks.

More specific contributions are summarized at the end of each chapter and in the conclusions.

1.5 Document organization

The document is outlined as follows:

Chapter 2: The literature review provides a summary of work that is related to this thesis. It also introduces important concepts that are employed within subsequent chapters.

Chapter 3: This chapter starts with a description of the full-scale tensegrity structure employed for testing. The structure is analyzed with respect to its geometrical and topological properties. Software implemented for specialized tasks is presented. Artificial neural networks are tested to correct the analytical model for increased accuracy. With the introduction of a five-module structure and a new control objective, the computational control prototype as well as the neural network used for accuracy enhancement had to be adapted.

Chapter 4: The application of stochastic search for the identification of good control commands is examined in this chapter. After an introduction into the complexity of the task, three techniques are compared on the same task in active structural control. A structure composed of three tensegrity modules as described in Chapter 3 is used.

Chapter 5: This chapter reports on further performance increases by learning. Case-based reasoning is proposed to enhance computational control speed. Search techniques are tested for adapting solutions that worked in similar situations for the current task. A first prototype developed for the three-module structure is tested. Additional aspects concerning maintenance of the case-based system are covered.

Chapter 6: Conclusions for the results obtained from all tests. Future directions for research are also proposed.

2 Literature review

This work covers the following domains of research:

- Tensegrity structures
- Structural control
- Optimization and search
- Artificial intelligence.

Although these subjects appear to be disconnected at first sight, they will be linked throughout this thesis to form the foundation knowledge that leads to performance increases of active structures over time. There are two objectives of this literature review: to provide an overview of research activities in the fields that are relevant to the objectives of the thesis and to present concepts that are employed in the subsequent chapters.

2.1 Tensile structures and tensegrities

Nature surprises mankind with efficient constructions. Spiders fabricate fascinating webs that are lightweight and ultra-resistant at the same time. Compared to steel cables, fibers created have a three times higher tensile strength; compared to Nylon, they are three times more elastic.

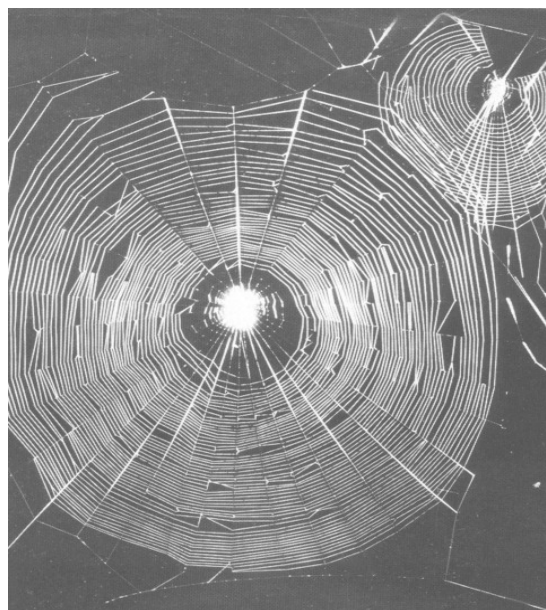


Figure 2-1 Nature's tensile structure. a spider web (Bach 1975)

Spider webs are tensile structures, they distribute load by pure tension. Such structures are appreciated for their excellent load-bearing/weight ratio. Aesthetical aspects are well combined with structural tasks. Long spans are possible with only a limited number of supports.



Figure 2-2 Munich Olympic stadium (Olympiapark 2002)

The Munich Olympic stadium (Figure 2-2) will serve to introduce the main concepts of tensile structural design. Built for the 1972 Olympic games, its design phase began in 1967. Choosing this structure as an example has also been motivated by the excellent roof design documentation available.

2.1.1 Structural design of tensile structures

Designing tensile structures is not a routine task in structural engineering. In contrast to conventional steel or concrete structures, design codes or guidelines are incomplete. Two additional aspects have to be considered:

- *Form-finding*. The initial equilibrium shape of a tensile structure has to be determined by experimental or analytical methods before analysis. This process is called form finding.
- *Geometrical non-linearity*. The assumption of small deflections is not longer true. Therefore, equilibrium has to be formulated on the deformed structure.

2.1.1.1 Form finding with experimental methods

Since the early Seventies, experimental methods have been applied for form finding. The former “Institute for Lightweight Structures” (IL) at the University of Stuttgart, now called “Institut für Leichtbau, Entwerfen und Konstruktion” (ILEK), contributed to a large extent to their development. Results are documented in several publications (Bach 1975; Bach et al. 1988; Bubner 1972; Burkhardt 1974; Pankoke 1972).

Experiments started using soap films to visualize minimal surfaces between fixed boundaries.

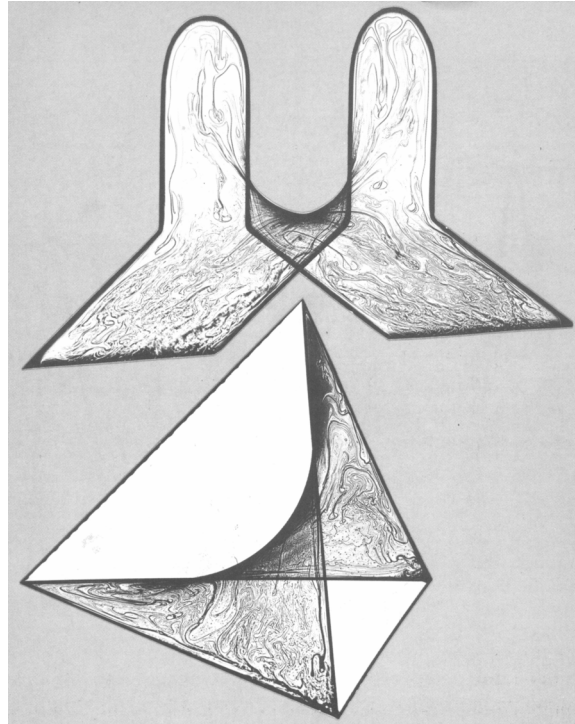


Figure 2-3 Form finding with soap films (Bach et al. 1988)

Once these surfaces are created, they need to be protected against drafts and temperature differences. Apparatus developed in Stuttgart use an air-conditioned chamber combining creation and protection of soap films. Surface topology is recorded with specialized photographic methods.

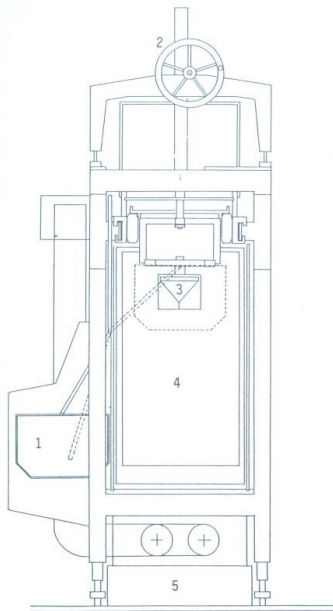


Figure 2-4 Apparatus for the creation of minimal surfaces with soap films (Bach et al. 1988)

Although this equipment reached a high level of perfection, they were inappropriate to be used with more complex structures. Additionally, tensile forces needed for dimensioning of cables and membranes cannot be measured on soap films. Models had to be created for this purpose. For example, a model of the Olympic stadium roof in Munich has been built and tested in Stuttgart (Bach 1975).

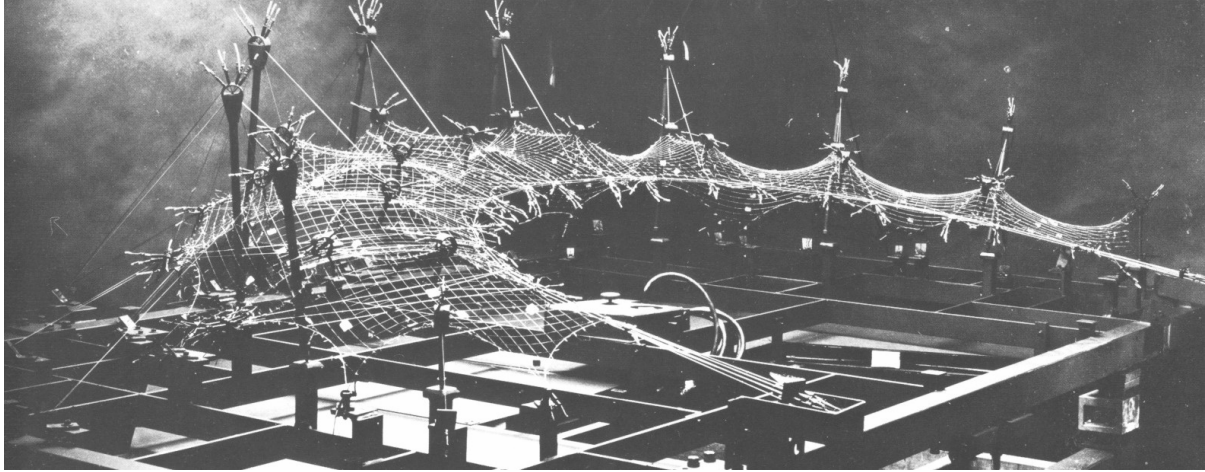


Figure 2-5 Experimental model of the Munich stadium roof (Bach 1975)

Cable forces were measured directly on the model. Two types of tension gauges have been used:

- A dial gauge
- A contact arm

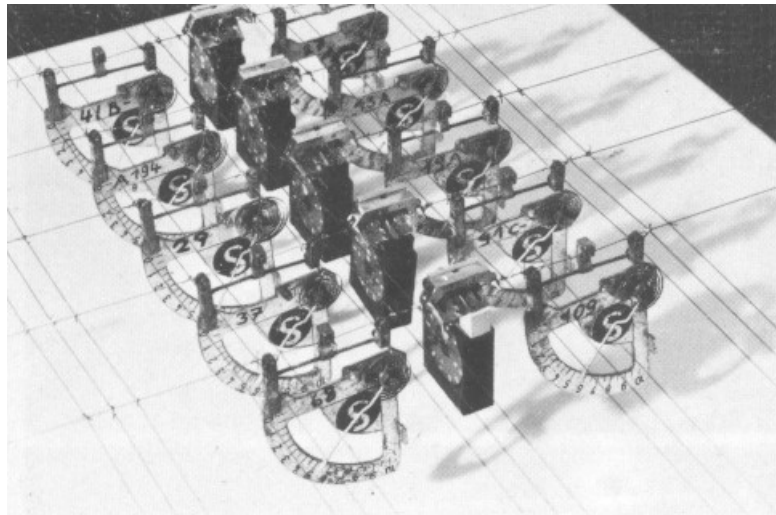
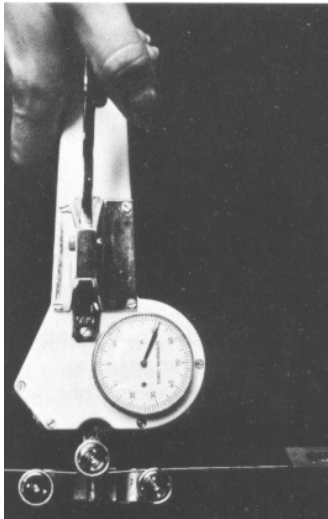


Figure 2-6 Measuring cable forces (devices specially constructed for the IL), temporary with dial gauges (left) and constant (right) measurement with contact arms (Bach 1975)

Deflections are measured by taking stereoscopic pictures of the deformed structure.

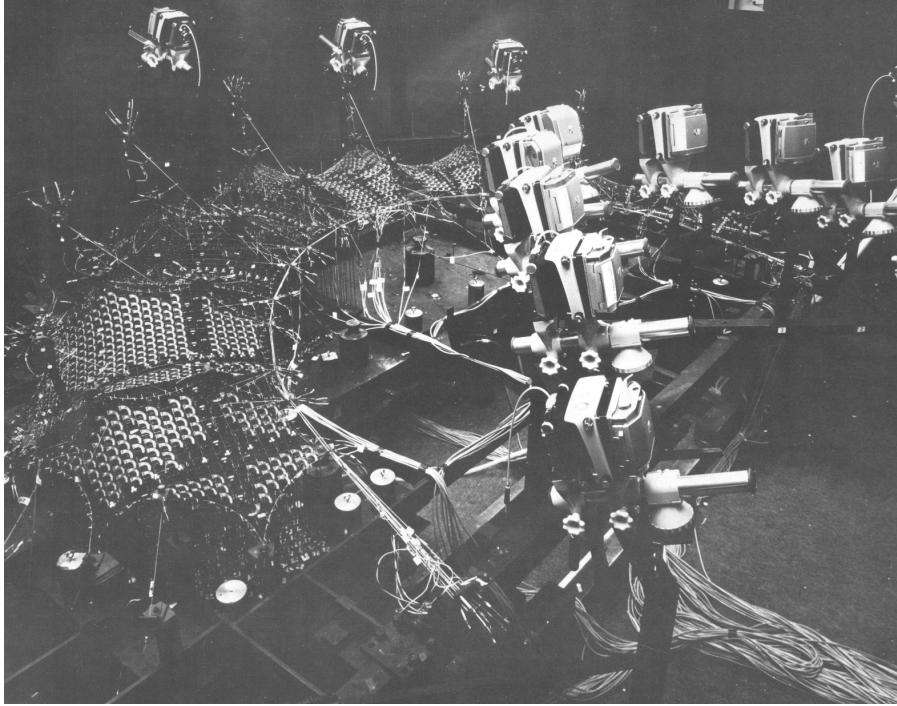


Figure 2-7 Measuring deflections with stereoscopic cameras (Bach 1975)

Methods for analytical form finding achieved a major breakthrough in this period. For the design of the Olympic Halls, the Olympic Swimming Pool and the Olympic Ice Sport Center, models have been replaced by numerical simulation.

2.1.1.2 Analytical form finding with the force density method

Experimental methods are undoubtedly the most intuitive way for engineers to understand structural behavior. Nevertheless, building models as well as the evaluation of measured results is time intensive and might be costly in the case of model failure.

In the late Sixties, research in structural engineering focused on the development of methods for computational structure analysis (Argyris et al. 1974). The force density method (Linkwitz and Schek 1971; Schek 1973) is one of the first proposals in the field of cable-structure analysis.

The main advantage of this method is that it solves only one system of linear equations. For example, nodal equilibrium at a node i of a pin-jointed framework in the direction of the x -axis of the system is formulated as

$$\sum_j \frac{t_{ij}}{l_{ij}} (x_i - x_j) = F_{ix} \quad (2-1)$$

- t_{ij} tension in the element connecting nodes i and j
- l_{ij} length of the element connecting i and j
- $x_i; x_j$ nodal coordinates in x -direction
- F_{ix} nodal force in x -direction

Similar equations are used to express equilibrium in y - and z - direction.

This is a non-linear equation, since the length l_{ij} is also dependent on the coordinates. By introducing the force density q_{ij} for each nodal connection, the expression is linearized.

$$q_{ij} = \frac{t_{ij}}{l_{ij}} \quad (2-2)$$

The force densities must be determined in advance. With a given interconnection, given load and given position of fixed points for each set of prescribed force densities, the shape of an equilibrium state can be determined.

The method has been extended to account for constraints in the choice of force densities. As defined, form finding methods provide means to find the initial equilibrium state of tensile structures. The structural analysis is a separate process that follows form finding.

2.1.1.3 Static analysis of tensile structures: the dynamic relaxation method

Analytic methods for the simulation of tensile structures may be classified into:

- Incremental methods
- Iterative methods
- Minimization methods (Barnes 1977)

A comprehensive overview of cable structure analysis methods is given by Tibert and Pellegrino (2001).

Incremental and iterative methods use the matrix formulation of finite elements. The approach consists in solving a system of equations which links the stiffness matrix K with the vector of loads P to obtain the structural displacements δ (Szilard 1982).

$$P = K \cdot \delta \quad (2-3)$$

Non-linear behavior is taken into account by adding the K^{NL} stiffness matrix to the linear stiffness matrix K . The incremental Euler-method solves this system of equations by stepwise application of the load ΔP . The stiffness matrix $K^L + K^{NL}$ has to be re-calculated during the iterations to correct for displacements:

$$\Delta P = (K^L + K^{NL}) \cdot \Delta \delta \quad (2-4)$$

Iterative methods, such as Newton/Raphston use a similar formulation. The load P is not applied stepwise but to its full extent, and the residual forces at the nodes are minimized during iteration. Both methods require that the stiffness matrix does not become singular.

As a vector-based method, dynamic relaxation does not share this problem. Dynamic relaxation decouples equilibrium and compatibility while converging to an equilibrium position.

Dynamic Relaxation is an attractive method for structures with highly nonlinear geometric and material behavior (Underwood 1983). First applications in the field of civil engineering can be found in the late 1970's. The thesis of Barnes (1977) can be identified as the beginning of increased research in that domain. Calculations are based on the dynamic equation of a damped system.

$$p(t) = M\ddot{d} + C\dot{d} + Kd \quad (2-5)$$

The state of static equilibrium is not found directly by matrix inversion as with finite elements. Instead it is determined iteratively. The motion of the structural nodes is traced until the residual forces of the nodes converge to a near '0' value. Parameters to assure convergence are the fictitious mass and damping matrices M and C and the time step, Δt .

The out of balance forces at the nodes in each direction at every point can be calculated by

$$R_{i;(x,y,z)}^t = M_{i;(x,y,z)} \cdot \dot{V}_{i;(x,y,z)}^t + C_{i;(x,y,z)} \cdot V_{i;(x,y,z)}^t \quad (2-6)$$

Boundary conditions are introduced by assigning large masses to fixed joints.

$$M_n = 10.0^{10} \quad (2-7)$$

Figure 2-8 presents the basic scheme of Dynamic Relaxation (DR in the following). Initial nodal residual forces are being evaluated on the undeformed structure. As long as their sum is still bigger than the defined threshold, new nodal velocities and residual forces are calculated. Please note that equations given in Figure 2-8 have to be formulated simultaneously in x, y and z-directions.

Calculate the initial residual forces, considering pretension, current bar tension and nodal forces.

Loop : While the sum of residual forces in the nodes is bigger than a given threshold do

Calculate the new velocity (velocities are calculated at $\Delta t/2$, $3\Delta t/2$, etc.):

$$V_{ix,y,z}^{(t+\Delta t/2)} = V_{ix}^{(t-\Delta t/2)} \left(\frac{M_{ix}/\Delta t - C_{ix}/2}{M_{ix}/\Delta t + C_{ix}/2} \right) + R_{ix}^t \frac{1}{M_{ix}/\Delta t + C_{ix}/2}$$

Calculate the new residual forces R (forces are calculated at Δt , $2\Delta t$, etc.):

$$x_i^{(t+\Delta t)} = \Delta t \cdot V_{ix}^{(t+\Delta t/2)} \quad R_{ix}^{(t+\Delta t)} = P_{ix}^{(t+\Delta t)} + \sum \frac{T_m^{(t+\Delta t)}}{L_m^{(t+\Delta t)}} (x_k^{(t+\Delta t)} - x_i^{(t+\Delta t)})$$

Check convergence:

$$\sum_{nodes i} (\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2) < \Delta_{lim}^2$$

End of Loop.

Figure 2-8 Simplified scheme of dynamic relaxation

Residual forces in trusses and cables are calculated using a simple linear truss element without bending.

$$T_m^{(t+\Delta t)} = \frac{EA_m}{L_m^0} (L_m^{(t+\Delta t)} - L_m^0) + T_m^0 \quad (2-8)$$

Convergence to near '0' values can be assured by intelligently choosing the parameters of dynamic relaxation. Papadrakakis (1981) examined several methods of automated parameter setting. Automated viscous damping, followed by kinetic damping, has attained best convergence rates.

When using kinetic damping, no damping factor has to be chosen. Kinetic energy is traced over the iterative process. Kinetic peaks are detected; nodal velocities are set to '0', since the position of a node is considered as the static position when kinetic energy stored is maximal (Figure 2-9, position $v = \max$).

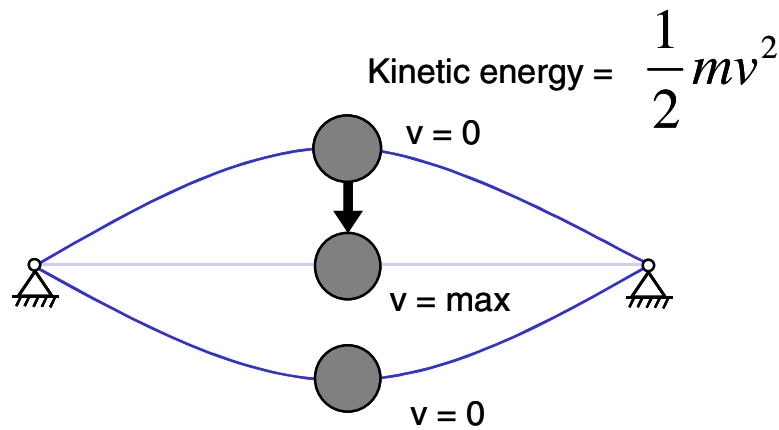


Figure 2-9 Maximum of kinetic energy in the equilibrium position of the structure

For practical problems, coordinates and velocities must be reset and the process has to continue through further peaks (Figure 2-10).

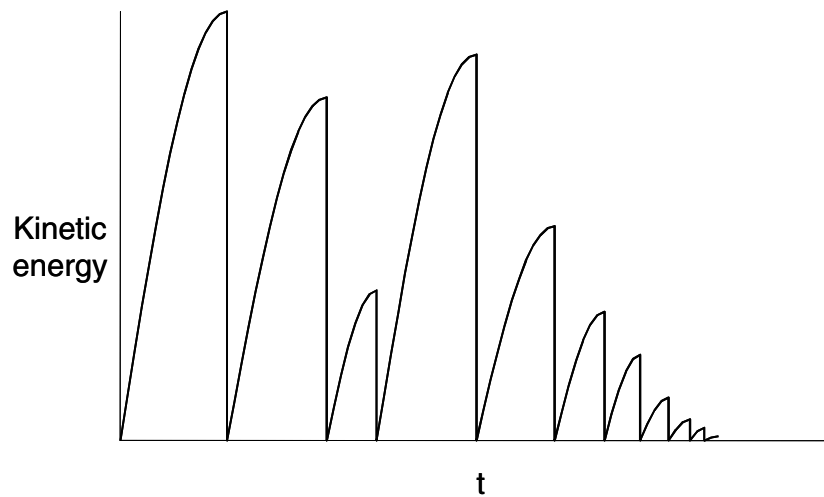


Figure 2-10 Kinetic damping

Instabilities of the calculation might occur when the time interval Δt exceeds a critical value or when fictitious nodal masses are too low. It has been shown by Barnes (1977) that, for any Δt , convergence can be achieved when the fictitious masses are chosen according to Equation 2.9. S_i is the greatest direct stiffness that may occur during analysis.

$$M_{ix} = \left(\frac{\Delta t}{0.5} \right)^2 \cdot \frac{S_i}{2} \quad (2-9)$$

Dynamic relaxation can be used for form finding with pre-defined stresses. It has been extended for the task of form finding with membranes (Adriaenssens and Barnes 2001;

Barnes 1994; Barnes 1999; Wakefield 1999). Companies specialize in the design of tension structures using their own software package based on the dynamic relaxation method (Tensys 2002).

The implementation of dynamic relaxation, which will be used throughout this thesis, has been provided by Rossier (1994). Concepts discussed before as kinetic damping and assuring convergence by setting fictitious masses according to Equation 2-9 have been incorporated.

2.1.2 Tensegrities

Tensegrities are a subclass of cable structures, where compression members are held apart by a network of tension members. The main distinction is that tensile forces need not to be anchored but are equilibrated by inner self-stress states (Motro 1992; Williamson and Skelton 1998). The most recent definition has being given by (Motro and Raducanu 2001):

“A tensegrity is a system in a stable, self-equilibrated state that contains a discontinuous set of components in compression inside a network of components in tension.”

The question of whom to credit for their invention is difficult to answer (Lalvani 1996). Fuller patented this concept in 1962 (Fuller 1962). The sculptor Kenneth Snelson met Fuller well before in 1948. Inspired by Fuller’s conceptual ideas, Snelson created his first structures employing discontinuous compression. Fuller realized the impact on his work and coined the notion of “tensegrity” by concatenating the words “tensile” and “integrity”. Snelson felt and still feels that Fuller stole his invention (Snelson 2002). Several well-known and admired tensegrity sculptures have been created by him (Figure 2-11).

David Georges Emmerich, a Hungarian architect worked from 1958 to 1996 in France. He holds a patent on tensegrities as well (Emmerich 1964). His main research interests concentrated on morphologies of tensegrity structures. In (Emmerich 1990), he presents an application of tensegrities which dates before Fuller and Snelson, the structure “Study in balance” by the Russian constructivist K. Ioganson created in 1921.

Introducing a more general approach, Ingber (1998) states that the tensegrity principle can be applied as a universal concept. Starting from comparing the human body with its 206 bones that are pulled against gravity with muscles, tendons and ligaments, the concept is also used on a microscopic level to explain cell growth and molecular arrangements.



Figure 2-11 Needle tower II, Kroller Museum in Otterlo, Holland (Snelson 2002)

In the past, tensegrities have mainly been constructed as sculptures. Nevertheless, the principle has been transferred to the construction of wide-span tensegrity cable domes. Most of them have been designed and constructed with support from Geiger Engineers (Gossen et al.) or Tensys (Tensys 2002), a design office that maintains and develops algorithms using dynamic relaxation. Both are also experienced in the construction process of such structures.

Motro and Pellegrino brought new life into tensegrity research (Motro 1997; Pellegrino 1992). The most up-to-date summary of research activity by the time this text is written is (Motro 2002). He identifies the following research directions:

- Mechanical behavior of tensegrities
- Deployable tensegrities
- Adaptive systems.

As described in the following section much work needs to be done in all three of these directions.

2.1.3 Structural particularities of tensegrity structures

Form finding and structural analysis methods developed for tensile structures are also useful for tensegrity systems. However, Calladine (1978) pointed out that the Maxwell rule, which is applied to determine whether a framework of straight bars connected at their ends by frictionless joints is determinate, a mechanism or indeterminate cannot be used in the context of tensegrity structures.

Pellegrino and Calladine (1986) use the equilibrium matrix \mathbf{A} to determine structural properties of statically and kinematically indeterminate frameworks. Results also apply to tensegrities. The assembly of the equilibrium matrix is shown in Figure 2-12.

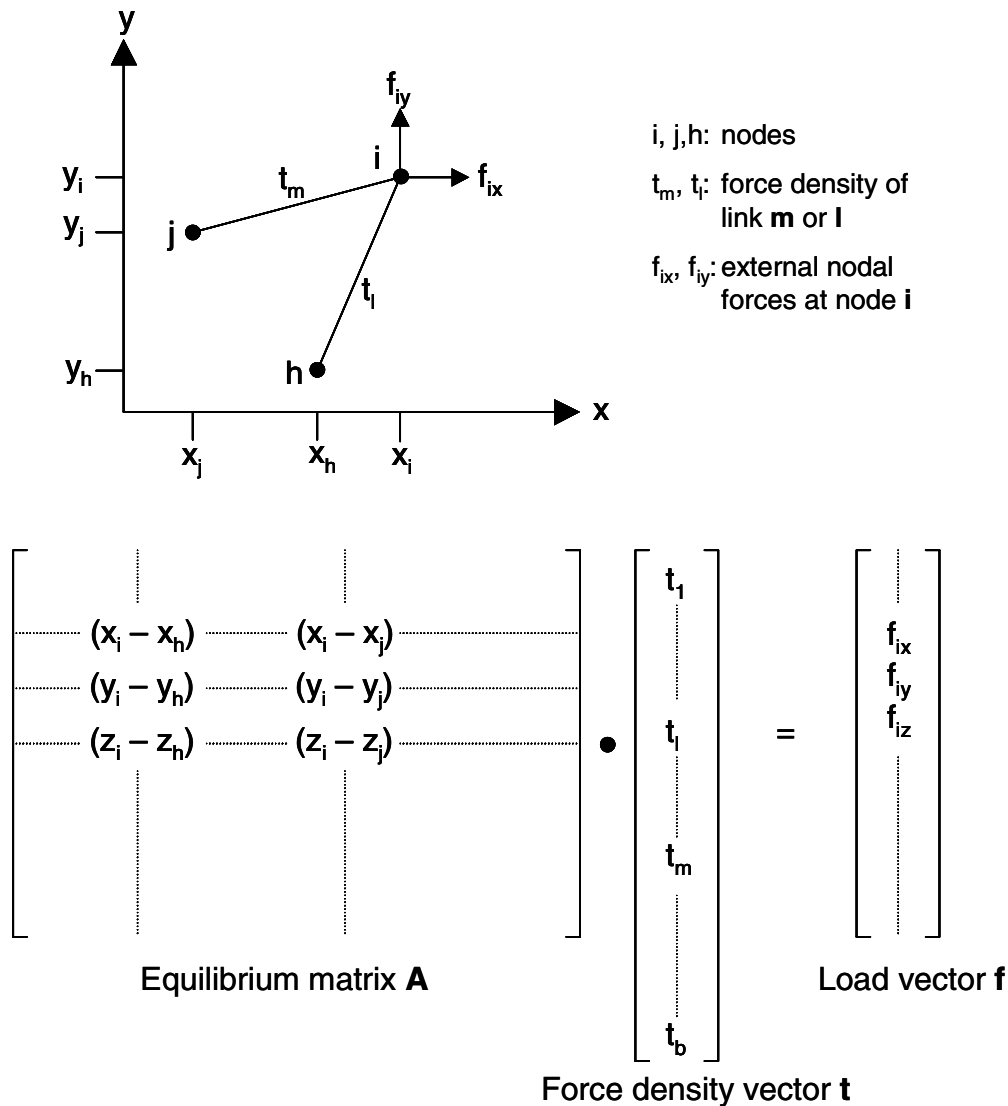


Figure 2-12 Assembly of the equilibrium matrix (Pellegrino and Calladine 1986)

The equilibrium matrix \mathbf{A} relates external with internal forces, the compatibility matrix \mathbf{B} external with internal displacements. It can be proven that

$$\mathbf{B} = \mathbf{A}^T \quad (2-10)$$

The number of stress states, s , of a cable structure can be determined using the following equations:

$$s = m - r \quad (2-11)$$

m : number of links (bars and cables) and

r : rank(\mathbf{A})

Mechanisms have to be distinguished in two categories: infinitesimal and finite. Finite mechanisms allow node displacements without changing element-length. Infinitesimal mechanisms describe nodal displacements where changes in element lengths are of lower order than changes in nodal displacements. In general, tensegrities have infinitesimal mechanisms. The number of mechanisms, q , is calculated as follows

$$q = n - r \quad (2-12)$$

n : 3J

J : number of non-constrained joints

A detailed analysis of mechanisms is provided in (Vassart et al. 2000).

Linear algebra operations applied on the equilibrium matrix \mathbf{A} provide further insight into structural behavior. The four fundamental subspaces of \mathbf{A} are intended to:

- Represent bar-tension vectors which are in equilibrium with the applied loads (row space of \mathbf{A})
- Represent all states of tension in the assembly which are in equilibrium with zero loads (states of self stress) (null space of \mathbf{A}).
- Give the range of load vector \mathbf{f} that can be supported in static equilibrium by the assembly in its original geometry (column space of \mathbf{A}).
- Represent the range of loads that cannot be carried by the assembly in its original configuration (null space of \mathbf{A}).

Recently, Murakami (Murakami 2001a; Murakami 2001b) presented equations for static and dynamic analysis of tensegrities. A nonlinear dynamic analysis methodology is described in (Kahla et al. 2000).

Sultan (2002) provided formulations and an analytical basis for dynamic analysis of tensegrity structures. Node friction is identified as the source of inaccuracies during simulation. Nevertheless, positive effects are that friction is a source of damping. Although analytical results have been derived, they have not been validated on a full-scale structure.

Averseng et al. (2002) demonstrate a methodology to calibrate a tensegrity grid such that a targeted self-stress state is attained. Results are verified on a model. The impact of geometrical non-linear behavior is neglected. This paper is one of the rare sources where simulated behavior is compared with the real behavior of tensegrity structures.

Oppenheim and Williams (2001) relate the non-linear stiffening of a three-bar tensegrity structure to applied torque. They state that vibrations of such a system cannot be suppressed by prestress or by active control. A further observation is that additional friction effects exist mainly at the joints in real systems.

Mechanics of tensegrities have been analyzed in many ways. All presented methods are mechanically correct. Nevertheless, their application to real structures might reveal additional challenges. Some assumptions (mainly linearizations) have not been validated experimentally. Model accuracy could be insufficient for practical applications.

2.1.4 Deployable tensegrities

Examples of deployable structures can be found in structural and aerospace engineering. The most recent application in structural engineering is the construction of retractable stadium roofs.

In aerospace, the capacity to transport objects into orbit is limited. Therefore, engineers propose structures that are lightweight and expandable. For example, deployable tensegrities are used to set up big space antennas (Tibert 2002). Salama et al. (1993) used heuristic search for the control of surface accuracy of space antennas. Nevertheless, the linear behavior that was assumed could not be observed. Non-linearities caused inaccuracies.

2.2 Structural control and intelligent systems

Structural control is mainly used to protect buildings against vibrations. Such control is motivated by goals such as:

- Isolate buildings to eliminate vibrations (caused by machines, railway traffic, etc.)
- Reduce vibrations of high-rise buildings during high wind loads
- Protect buildings against earthquakes

Structural control can be subdivided into passive control, active control and semi-active control.

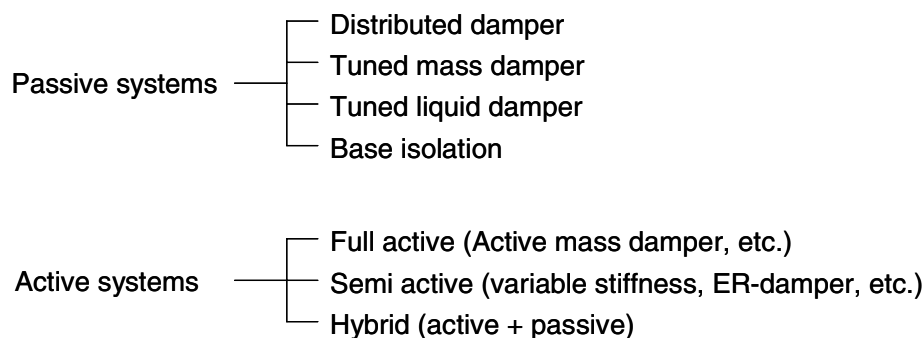


Figure 2-13 Classification of structural control

Symans and Constantinou (1999) give the following definitions:

Passive control systems: a passive control system may be defined as a control system which does not require an external power source for operation and utilizes the motion of the structure to develop control forces. Control forces are developed as a function of the response of the structure at the location of the passive control system.

Active control systems: an active control system may be defined as a system which typically requires a large power source for operation of electrohydraulic or electromechanical actuators which apply control forces to the structure. Control forces are developed based on feedback from sensors that measure the excitation and/or the response of the structure. The feedback from the structural response may be measured at locations remote from the location of the active control system.

Semi-active control systems: a semi-active control system may be defined as a system which typically requires a small external power source for operation (e.g. a battery) and utilizes the motion of the structure to develop control forces, the magnitude of which can be adjusted by the external power source. Control forces are developed based on feedback from the sensors that measure the excitation and/or the response of the structure. The feedback from the structural response may be measured at locations remote from the location of the semi-active control system.

Housner (1997) provides a compact overview of structural control. The interest of Japan and the United States of America in this domain is mainly a result of earthquake concerns. Both countries have constructed important infrastructure projects in seismic active regions. The San Francisco Bay area is one example (Figure 2-14). Focus lies on protecting bridges that are essential for the economy in this region. Main links are the San Mateo Bridge, the San

Francisco - Oakland Bay Bridge, the Golden Gate Bridge, the Richmond – San Rafael Bridge. The Loma Prieta earthquake in 1989 caused the failure of the upper deck of the San Francisco – Oakland Bay Bridge.

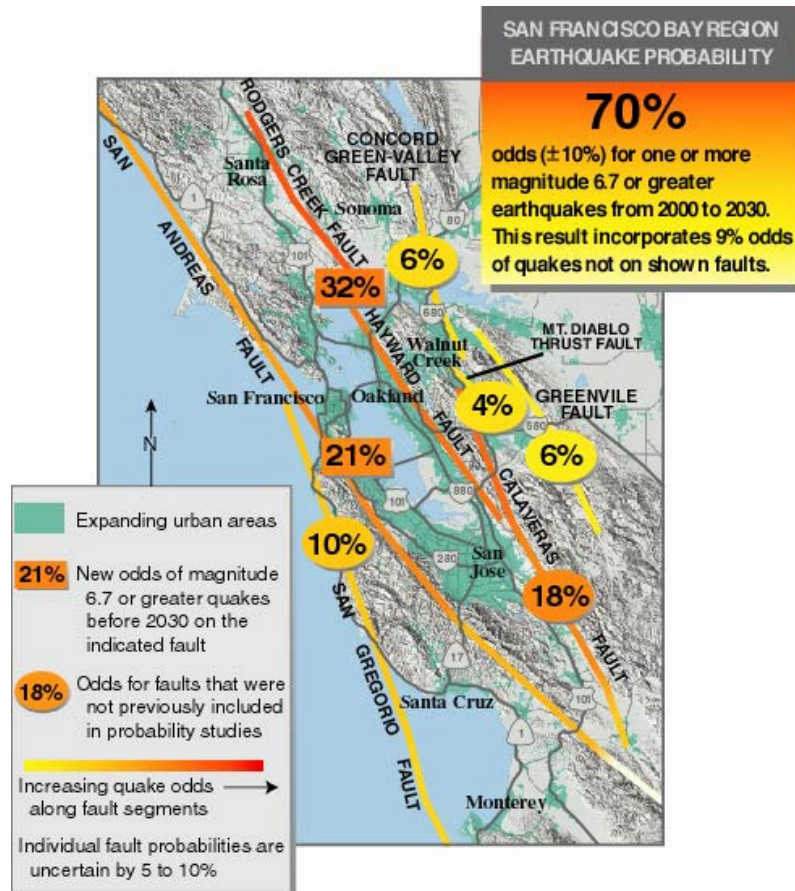


Figure 2-14 Map of seismic risks in California (U.S. Geological Survey 2002)

2.2.1 Passive control

Passive structural control design methods are well established (Chopra 2001). Once installed, such systems are intrinsically reliable, since no power supply and control computer are needed. Tuned mass dampers (TMD) reduce building vibrations by adding mass at well-defined positions of the structure. They are usually used for structures with low natural frequencies and little damping.

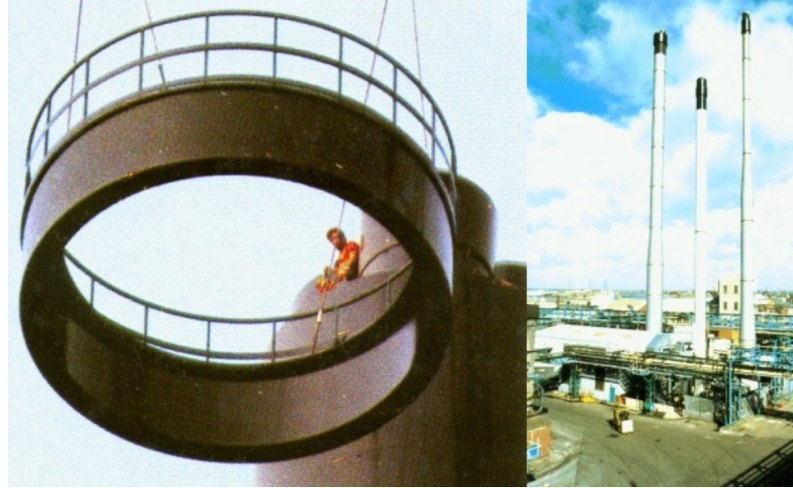


Figure 2-15 Tuned mass dampers for chimneys (Steelcon 2002)

Good performance is achieved when engineers account for passive control during initial structural design. Although the design stage is the phase where we know the least about the structure, necessity for additional damping can be checked. The structure may then be designed for possible in-service retrofits. Spring oscillating mass viscodampers are now available. For example, they have been installed at the Burj al Arab Hotel to enhance guest comfort under high wind loads.

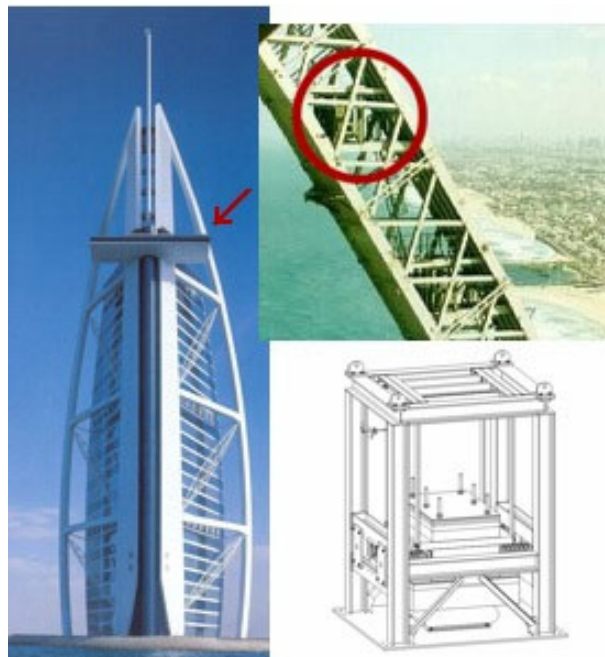


Figure 2-16 Hotel building in the Emirates (Gerb 2002)

A subsequent addition of damping devices might be necessary when structures show unexpected behavior. The Millennium bridge, London, had to be closed after one day of service, since the maximum lateral displacement of the bridge deck was 70mm. After a retrofit, vibrations are now dissipated by a combination of viscous and tuned mass dampers (Dallard et al. 2001).



Figure 2-17 Millennium bridge, London (Arup 2002)

As indicated before, passive control serves not only to reduce vibrations governed by serviceability criteria but is also used for seismic protection. Japan employs mainly base isolation (Kitagawa and Midorikawa 1998). Due to increased seismic risks more than 280 projects on seismic isolation have obtained special construction permission by the end of 1996. Nawrotzki (2001) compared four different strategies for seismic protection of buildings

- Base isolation system, which uncouples the structure horizontally
- Tuned mass damper, an additional mass on the top of the building combined with a spring/damper system
- Elastically uncoupled top story, same as the tuned mass damper, but the whole story is used as the mass.
- 3-D base control system, a combination of horizontal and vertical damping with helical springs and viscous dampers

Systems are depicted below.

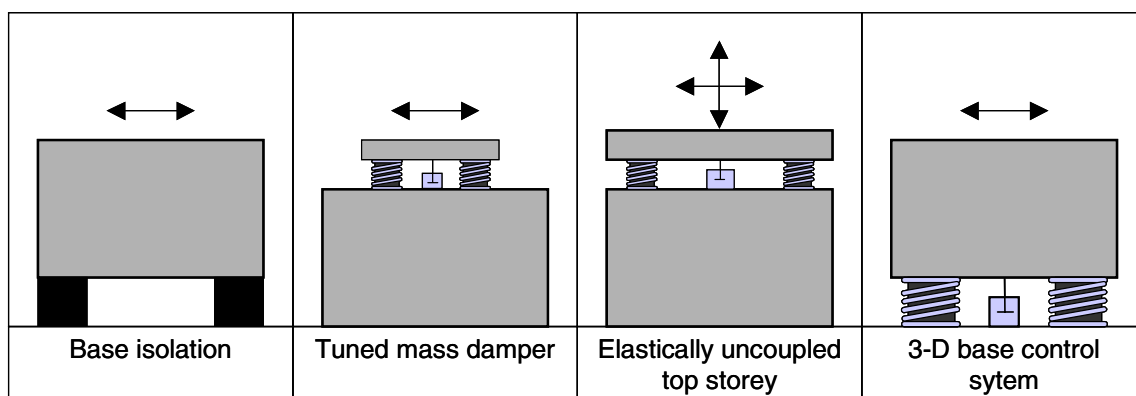


Figure 2-18 Four strategies for passive control

Best results are obtained by the 3-D base control system, regarding acceleration damping as well as reducing the displacements.

As discussed before, the San Francisco Bay area region of the United States is also a seismically active zone. Devices employed here are spherical sliding bearings to protect bridges against structural failure.

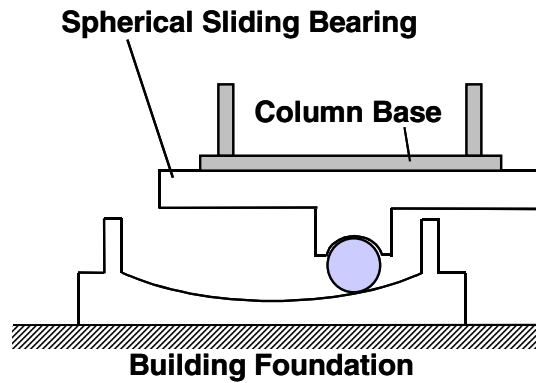


Figure 2-19 Spherical sliding bearing

Since no long-time experience exists as to how device parameters change over service life, a testing machine has been constructed in San Diego. Devices are tested on a regular basis. During the test, a spare damper replaces the tested system.

2.2.2 Active control

Compared with other control methodologies, active control has potentially the most impact on a building's behavior. Architects demonstrate their interest in actively controlled environments (Fox 2001). This technique demands investments in design and construction phase of structures as well as during their lifecycle and maintenance. Figure 2-20 shows a general schematic.

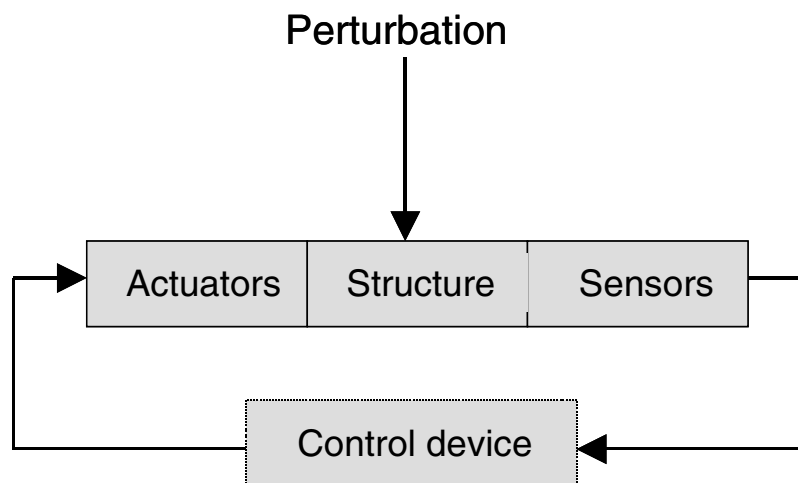


Figure 2-20 Schematic of active structural control

Active structural control uses a system consisting of sensors, a control device and actuators. Building displacements are measured by sensors and used as indicators for control commands. The control computer evaluates the implemented objective function each time sensors report changes. If necessary, a control command is calculated and sent to the actuators to attain the control objective. A methodology to obtain control forces applied by actuators is developed in (Nishimura et al. 1992; Nishimura et al. 1998).

This technique has been applied to several full-scale buildings (Spencer and Sain 1997). The first building to be equipped was the Kyobashi-Seiwa tower in Tokyo, Japan, 1989. The building's slender shape makes it susceptible to transverse vibrations. The control system was installed on the top floor.

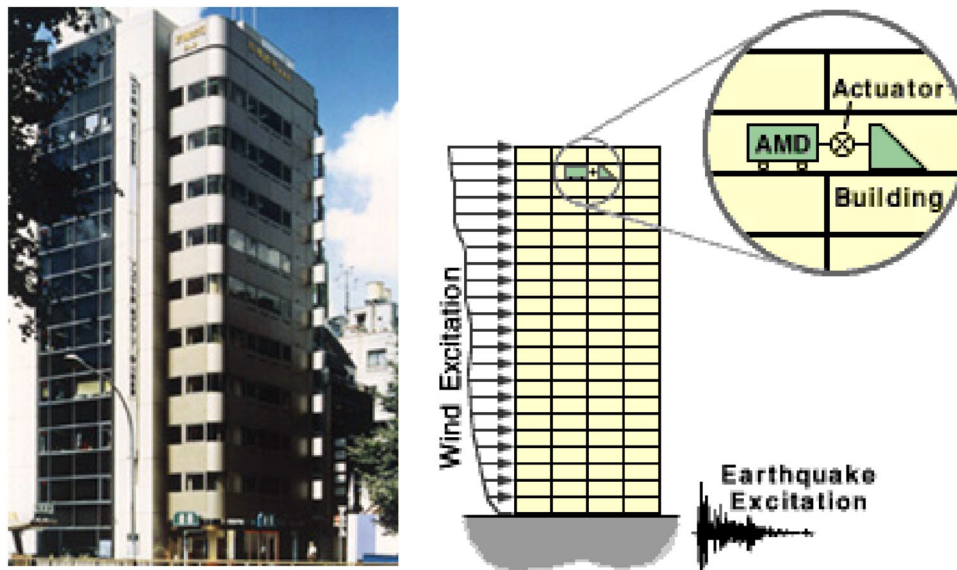


Figure 2-21 Active control for the Kyobashi-Seiwa tower (Spencer and Sain 1997)

Research in the field of active structural control aims to decrease the energy needed for actuators. One of the inventors of active structural control, Takuji Konbori, proposed an alternative to active mass damper systems: the active variable stiffness system (Konbori et al. 1993). The system has been tested on a three-story building. Variable stiffness devices are installed in steel braces. The control strategy is to change the structural system by locking or unlocking the braces.

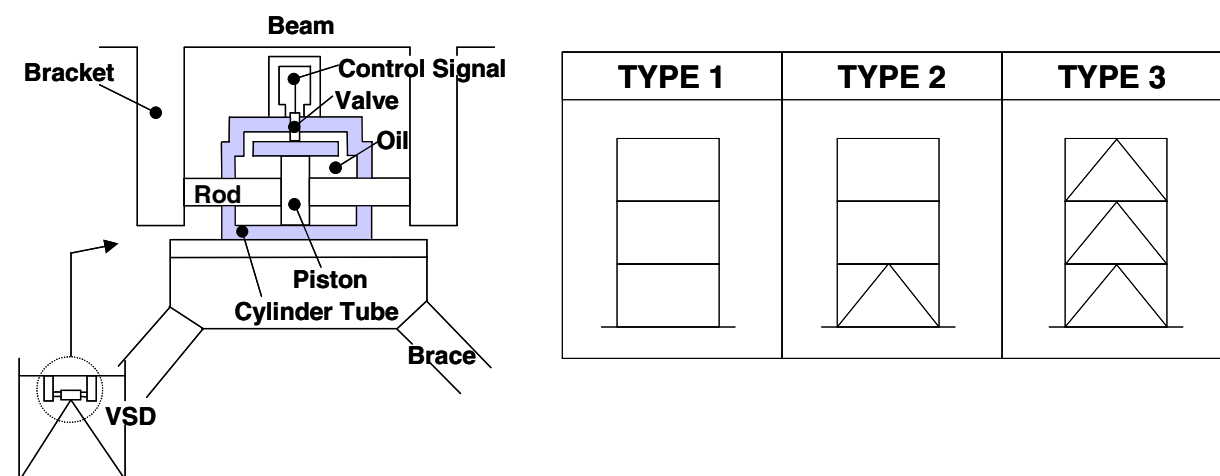


Figure 2-22 Variable stiffness device mechanism (left) and variation of building stiffness (right)

The type of building stiffness is chosen after a stiffness judgment criterion has been evaluated for the three variants. Please note that only three different stiffness types are to be evaluated in each judgment phase. Power required to operate this system is small.

Active control is not as intrinsically reliable as passive systems. These systems need to be redundant to assure operation even in cases of emergency. Another drawback is that the cost function, which is optimized to provide optimal control, is sometimes not mechanically meaningful (Housner 1997). Most systems are also limited to operate within the specifications that are fixed at the beginning of the design phase.

2.2.3 Semi-active control

The definition and functioning of semi-active control is similar to those given for active control (Ricciardelli et al. 2000). However, energy consumption is smaller than with active systems and provides, therefore, possibilities to apply such installations to control objectives that are governed by security criteria.

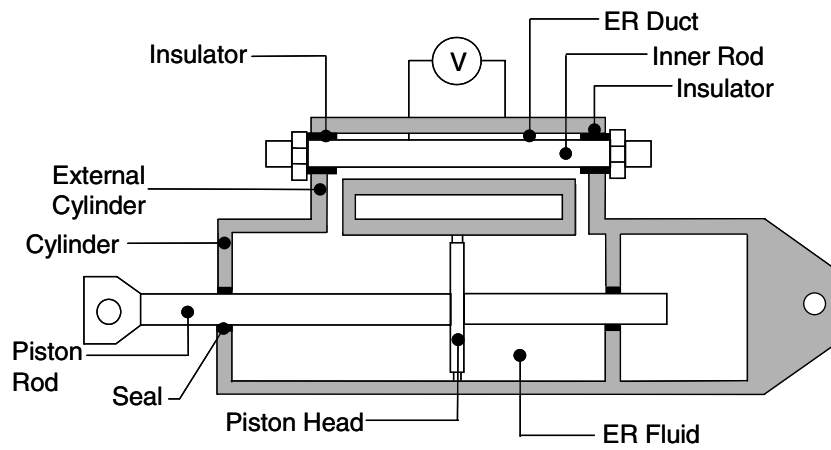


Figure 2-23 Electrorheological damper

Electrorheological (ER) dampers consist of hydraulic cylinders where the cylinders are filled with small dielectric particles. With no power applied, the device behaves like a viscous damper. By varying the electrical field, the dynamic behavior of the device can be modulated from soft to hard.

2.2.4 Control of tensegrity systems

Djouadi (1998) presents the adaptation of optimal control strategies to tensegrity structures. Weight matrices used in this approach are difficult to determine. In the described simulation, the damping of structural vibrations is the control objective. The system is controlled by virtually changing the bar length/cable length. According to the knowledge of the author, no experimental verification of this approach exists. Also constraints, which check and limit cable stresses, are not introduced.

Sultan (1999) proposes a formulation for tensegrity structure control. He illustrates the control application using the example of an aircraft motion simulator. The error between deployment path and equilibrium path is minimized. This is a conventional control approach, which cannot be adapted to multiple objectives but serves mainly deployable structures. To derive his equations, he neglects multiple mechanical effects. No verification on a full-scale structure has demonstrated that the assumed linearizations are appropriate.

Skelton (2000) states that only little energy is needed to change the shape of tensegrity structures. He concludes that they are advantageous for active control. Proposed applications are as airplane wings or in microsurgery. Active components in his concept are cables.

Fest (2002) designed and constructed a modular full-scale tensegrity structure equipped with devices for active structural control. Dynamic relaxation has been tested successfully for evaluating control commands proposed by stochastic search. Commands found have been tested on the structure. They were able to compensate deflections caused by external loading. This thesis provides an additional review of tensegrity structure analysis and tensegrity structure control.

2.2.5 Intelligent structural control

The optimal placement of sensors and actuators is a continuing research topic (Heo et al. 1997). The use of pareto-optimal curves is proposed in (Brown et al. 1999). Adeli and Saleh describe the theoretical concepts of structural control. Examples are given how to transfer these concepts into algorithms for multi-processor systems (Adeli and Saleh 1999). A drawback of current active control systems is their limitation to satisfy only one design goal. They are not able to improve performance from past experience (learning) nor can they set priorities to reach control goals (planning).

Traditional control systems can be improved through coupling them with strategies developed in the field of artificial intelligence (Shoureshi 1995). The potential of neural networks has led to the proposals to use them for structural control (Rehak and Garrett 1992). Zagar and Delic developed a computational model of a bridge, which was controlled by a neural network (Zagar and Delic 1993).

The term “intelligent structures” has been applied in multiple distinct ways to civil engineering. This thesis employs a definition given in (Smith and Shea 1999). Artificial intelligence methods have the potential to extend structural control:

- To overcome the limitations of current systems that use only one control objective
- To provide means for the control of highly coupled, non-linear systems
- To construct active structures which increase performance over time

Under present conditions, such a system is not practical for structures that are governed by safety criteria, such as earthquake damage protection, since reliability requirements would be costly to satisfy. Structures that are governed by serviceability requirements are more appropriate applications.

2.3 Optimization and stochastic search

Optimization in structural engineering often focuses on minimization of construction costs. In these examples, the objective function often evaluates total structural weight only, since the amount of material employed is assumed (erroneously) to be equivalent to the cost. An optimization algorithm identifies a set of design variables implemented in the objective function such that costs are minimized. (Bullock et al. 1995; Cagan et al. 1998; Ceranic et al. 1999; Deb and Gulati 2001; Koumousis and Georgiou 1994).

A definition of search has been given in (Leake 2001):

Search is a process of formulating and examining alternatives. It starts with an initial state, a set of candidate actions, and criteria for identifying the goal state. [...] Starting from the initial state, the search process selects actions to transform that state into new states, which themselves are transformed into more new states, until a goal state is generated.

The number of applicable techniques is vast. It is reduced for problems where solutions cannot be found by a closed mathematical formulation and complexity is exponential. There is a certain class of problems where the application of deterministic techniques is not tractable. Intractability means that execution time increases exponentially with the number of optimization variables. One frequently used example of an intractable problem is the “traveling salesman problem”. The search for an optimal arrangement of cities to be visited by the salesman with respect to minimizing the tour length increases exponentially with the number of cities on the tour.

Although one objective of search is to converge as fast as possible to the optimal value, another is to visit a sufficient number of candidate solutions to avoid local minima. The following sections will discuss three techniques, which use different strategies:

- Simulated annealing (SA)
- Probabilistic global search Lausanne (PGSL)
- Genetic algorithms (GA).

Salama et al. (1993) proposed the use of stochastic search in conjunction with a structural control task. Although the method applied (simulated annealing) found a set of good control commands, the cost of the simulated solution differed significantly from the measured response of the actively controlled system. Control movements induced deflections that were in the magnitude of microns. These deflections were the result of structural non-linear behavior. The linear model used to evaluate the objective function encountered inaccuracies during the search process.

Farsangi (2002) employed genetic algorithms for topology optimization of a double layer grid. The use of a static configuration in his approach does not reflect different load cases. The objective function used employs a sensitivity metric to determine the importance of structural members. This metric changes, however, with changing loads.

Finding the right parameters for an optimization task can be considered as an optimization task in itself (Grefenstette 1986). Salajegheh and Lotfi (2002) tested genetic algorithms for the optimization of a double-layer grid. Throughout the process, algorithm parameters such as probabilities for crossover and mutation are constantly being adapted by simulated annealing. These are normally constant throughout one optimization task. As a result, convergence is smooth and fewer generations (number of iterations) are needed for good results.

Optimization techniques are often proposed as ‘generic’. That means that they should be applicable to a wide range of tasks through tuning parameters. In reality, the scope of application is not universal: each technique has a range of applicability. Nevertheless, the choice of parameters of each algorithm can have significant impact on convergence behavior and attainable threshold. For this reason, the following sections present, after describing each algorithm, a paragraph that explains criteria to choose algorithm parameters. It is acknowledged that approaches other than the ones described exist.

Shea et al. (2002) propose a system for intelligent structural control of tensegrity structures. However, no experimental verification was reported and no comparison of algorithms was performed.

Concepts and techniques discussed in the following are used in Chapters 4 and 5.

2.3.1 Simulated annealing

2.3.1.1 An introduction to simulated annealing

Simulated annealing (Dowsland 1995; Kirkpatrick et al. 1983) stems from an analogy to the annealing of metals where temperature schedules are used to control the arrangement of atoms during their crystallization process. It is a step-wise technique that allows moves to inferior solutions and, therefore, is more able to overcome local minima (Figure 2-24).

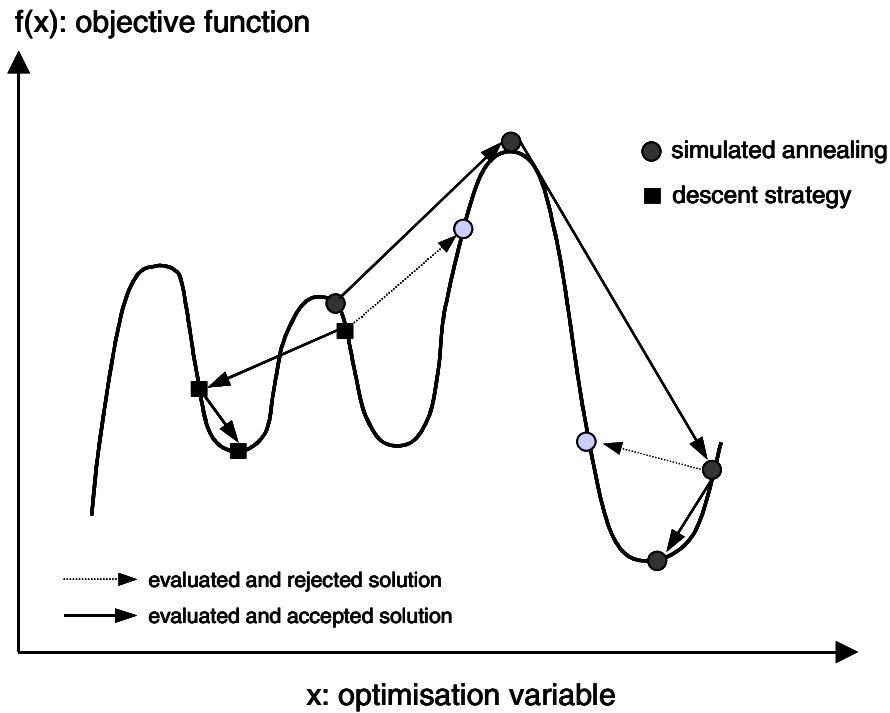


Figure 2-24 Schematic comparison of simulated annealing (SA) and a descent strategy (DS)

Whereas the likelihood to accept worse solutions in the beginning of the annealing process is high, it drops to ‘0’ at close to the end. This process is driven by the function

$$P_{accept} = e^{-\frac{\Delta C}{T}} \quad (2-13)$$

The change in the objective function, or cost, between two moves is denoted by ΔC . P_{accept} is compared to a randomly generated value between 0 and 1, and the inferior candidate solution is accepted when the random value is less than P_{accept} . In general, the temperature “T” at the beginning of the process is fixed for each problem as a schedule parameter and is then reduced to zero during the optimization process according to an “annealing schedule”. During the last section of this schedule, the “freezing” stage, only better moves are accepted and simulated annealing behaves as a descent strategy.

Simulated annealing algorithms are reasonably robust if the parameters controlling the cooling curve are assigned values that reflect the complexity of the solution space. Starting from an initial temperature, $T_{initial}$, the temperature update is defined as:

$$T^{n+1} = \frac{1 - (accept_rate_{actual} - accept_rate_{target})}{K} \times T^n \quad (2-14)$$

where K is a constant; a value of 10 has been found to be effective.

Other schedule parameters affecting the performance of this schedule include the number of candidate solutions considered in each iteration, the number of iterations in a complete search process, how often the temperature is updated and over what statistical interval.

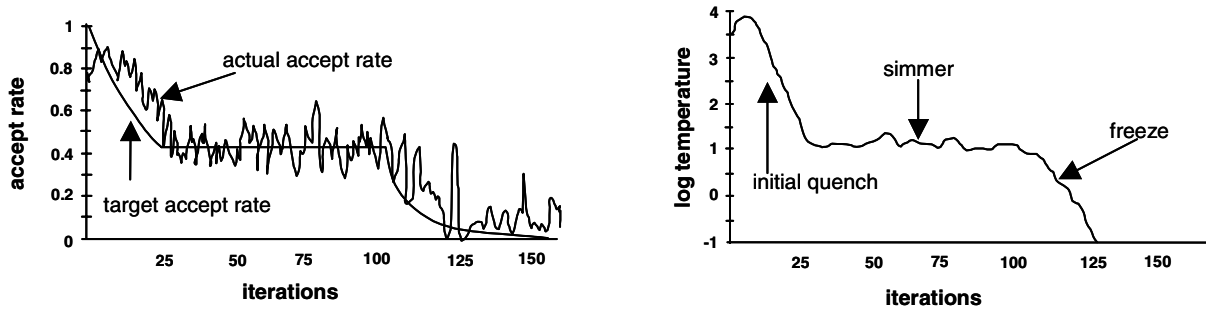


Figure 2-25 Example of accept rate and temperature schedules

From the initial state, candidate solutions are generated by selecting a single system variable at random and perturbing it within the allowable variable ranges.

2.3.1.2 Schedule parameter setting

Several approaches for schedule parameter setting of simulated annealing exist. While newer annealing schedules may be available compared to those described, these techniques have been proven successful in the domain of structural topology optimization (Shea and Smith 1999).

The modified Lam-Delosme annealing schedule (Swartz and Sechen 1990) operates by assuming an optimal profile for the percentage of candidate solutions that should be accepted at each iteration of the search process and adjusting the temperature over a statistical interval to achieve this target accept rate.

Each iteration consists of a number of moves that try to direct the solution towards the optimum. Hustin move sets (Hustin 1988) assign each “move” a sub-range of the maximum allowed variable change. Since the move sub-range only defines the upper limit of a variable change, smaller moves are always possible. Each move range, r , is then assigned a quality, Q_r :

$$Q_r = \frac{\sum_{acceptrules} |\Delta C(r)|}{number_of_attempted_rules} \quad (2-15)$$

according to the change in cost of past applications of the rule, $\Delta C(r)$. Rule qualities are used to update the probability of selecting a move sub-range in subsequent perturbations of the solution. Generally, larger moves are used in the beginning of the process whereas smaller moves are used towards the end as the solution converges.

A guideline given in (Swartz and Sechen 1990) for the number of moves per iteration is

$$10 \times n_{variables}^{\frac{4}{3}} \quad (2-16)$$

with $n_{variables}$ = number of variables.

Leite (1999) proposed to accelerate the convergence process by implementing a parallel simulated annealing algorithm. It has also been applied to multiobjective optimization (Chattopadhyaya and Seely 1994).

2.3.2 Probabilistic global search Lausanne (PGSL)

2.3.2.1 An introduction to PGSL

PGSL is a newly developed search technique (Raphael and Smith 2000). It is based on the assumption that sets of better values are more likely to be found in the neighborhood of sets of good values and, therefore, intensifies search in regions that contain sets of good solutions. Gradients are not required in the search. The algorithm itself consists of four nested loops.

```

Set the complete search space as the current subdomain
Loop 1: Repeat for NSDC (Number of Sub-Domain Cycles) iterations
    assume a uniform probability density function (PDF) for all
    variables in the current subdomain.
    Loop 2: Repeat for NFC (Number of Focusing Cycles) iterations
        Loop 3: Repeat for NPUC (Number of Probability Upsampling Cycles)
        iterations
            Loop 4: Repeat for NSC (Number of Sampling Cycles) iterations
                Generate a solution using the current PDF
            End of Loop 4
            Select the best solution in Loop 4.
            For each variable, locate the interval containing the best
            value.
            Increase the probability of this interval.
        End of Loop 3:
        Select the best solution in Loop 3: Subdivide the interval
        containing the best solution.
        Assume a uniform probability within the best interval.
        Assume an exponentially decreasing distribution away from the
        best interval.
    End of Loop 2:
    Select a smaller subdomain centered around the best solution so
    far. The width of this subdomain is chosen after performing certain
    checks to prevent premature convergence

End of Loop 1.

```

Figure 2-26 Algorithm of the four nested loops of PGSL

At the beginning of the optimization process, an axis representing minimum and maximum values for each variable is created. The interval of solutions is subdivided into 20 intervals with an equal probability of 0.05.

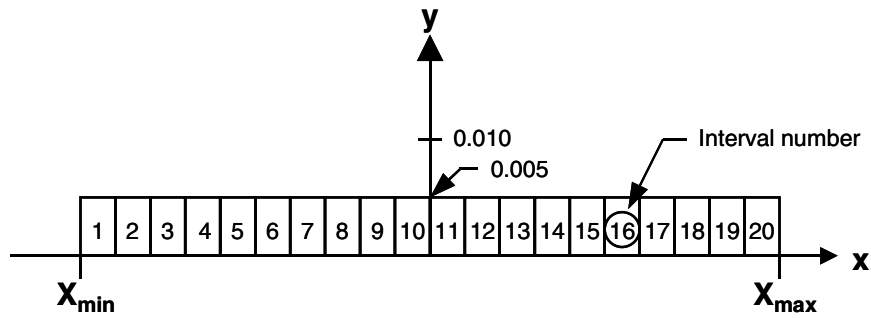


Figure 2-27 Probability distribution at the beginning of the PGSL algorithm

Possible solutions are now generated randomly but with respect to the probability distribution. In the probability updating cycle, probabilities are increased in regions that evaluated good solutions and decreased in other regions. This is done such that overall probability stays constant.

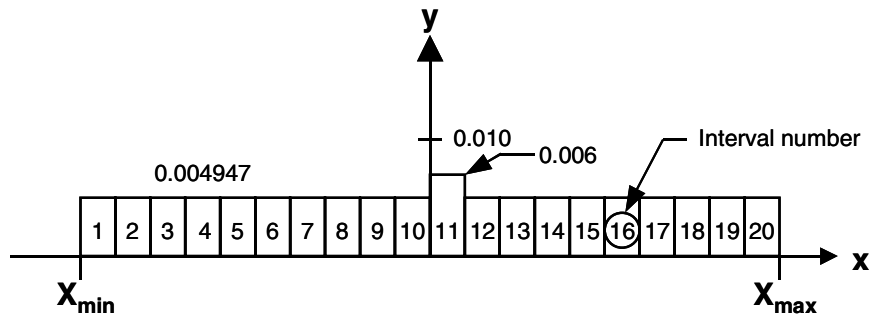


Figure 2-28 Change of PDF in the PUC

The next loop, called focusing cycle, further subdivides regions with good solutions and merges regions with bad solutions. Search is focused now on regions containing good solutions. The number of intervals is kept constant (20).

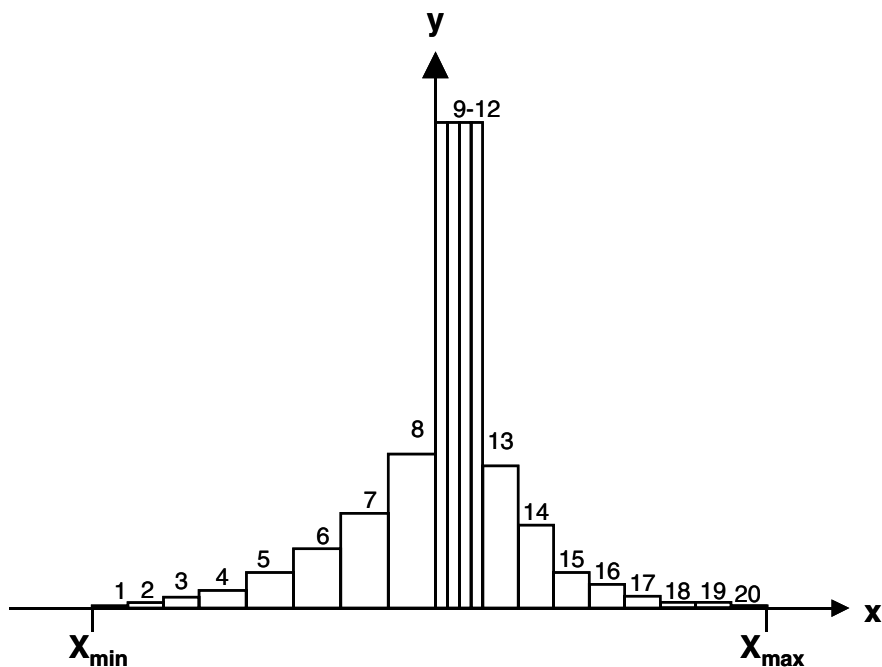


Figure 2-29 Change of segmentation in the FC

The last cycle, the subdomain cycle, cuts out regions where no good solutions have been found. A distribution near the end of the subdomain cycle is similar to the one presented in Figure 2-30.

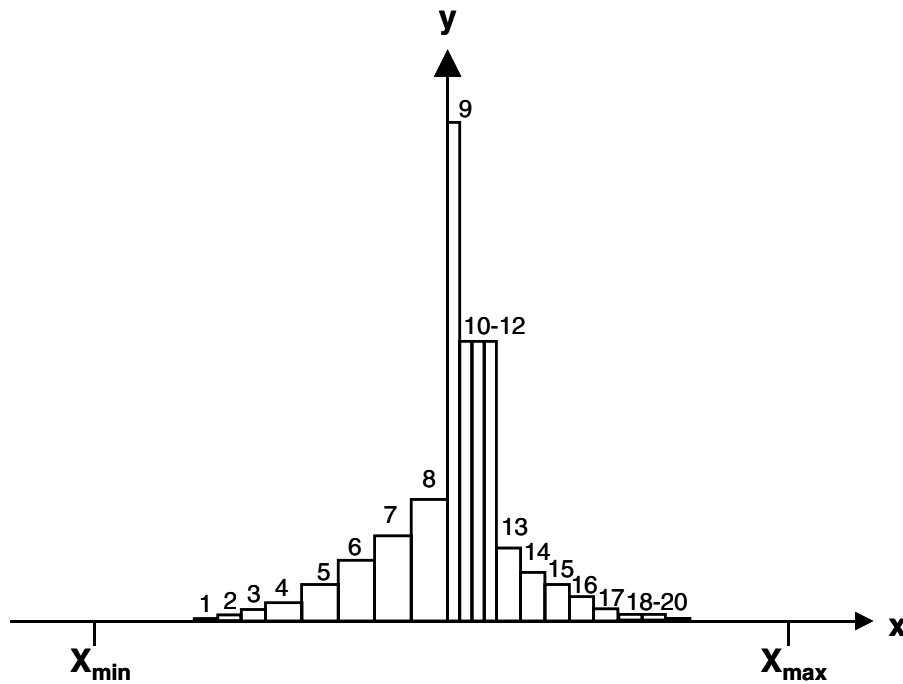


Figure 2-30 Subdomain cycle

A feature that PGSL shares with other random search methods, such as adaptive random search, and controlled random search is the use of a PDF (Probability Density Function). However, the following differences between PGSL and other random methods are:

1. Other random methods that make use of an explicitly defined PDF follow a "creep" procedure similar to simulated annealing. They aim for a point-to-point improvement by restricting search to a region around the current point. The PDF is used to search within a small neighborhood. On the other hand, PGSL works by global sampling. There is no point-to-point movement.
2. The four nested cycles in PGSL are not similar to any features of other algorithms.
3. Representation of probabilities is unique in PGSL. Other methods make use of a mathematical function with a single peak (e.g. gaussian) for the PDF. PGSL uses a histogram - a discontinuous function with multiple peaks. This allows fine control over probabilities in small regions by subdividing intervals.
4. Probabilities are updated in different ways. The primary mechanism for updating probabilities in other methods is by changing the standard deviation. In PGSL, the entire shape and form of the PDF can be changed by subdividing intervals as well as by directly increasing probabilities of intervals.

The algorithm has been tested on non-linear benchmark problems and compared with results from genetic algorithms applied to the same problems (Raphael and Smith 2000). When no problem-specific knowledge is employed, PGSL performs as well as genetic algorithms. It has already been applied to several tasks in the field of structural engineering, such as

optimization of timber shear wall structures (Svanerudh et al. 2002) and bridge diagnosis (Robert-Nicoud et al. 2000).

2.3.2.2 Adjusting the parameters of PGSL

Parameters to be adjusted are the number of iterations for each one of the four nested loops. For the detection of optimal parameters, the following procedure was employed. Drawing from experiences made with other optimization problems, the number of sampling cycles is set at two and the number of probability updating cycles at one. This means that effectively only two parameters need to be adjusted

- The number of iterations in the focusing cycle (NFC)
- The number of iterations in the subdomain cycle (NSDC)

The number of iterations for the third loop (NFC) should be fixed at $P \times$ number of variables, where P varies from 10 to 20. Values for the number of iterations in the subdomain cycle must be determined by experiment. Focusing cycles (NFC) and subdomain cycles (NSDC) need to be adjusted to fix the total number of evaluations of the objective function. This demonstrates the ease and simplicity of fixing PGSL parameters.

2.3.3 Genetic algorithms

2.3.3.1 An introduction to genetic algorithms

Genetic Algorithms (GA's) are an analogy of Darwin's theory of evolution. It observes that nature produced highly adapted creatures over a long process and only the fittest had the possibility to survive and reproduce themselves. This analogy has been transferred to computer programs. Evaluating an objective function indicates fitness. GA's have been successfully tested on design problems in engineering (De Jong et al. 1999); latest developments lead to application in the field of multiple criteria optimization (Coello Coello 1993; Fleming and Purshouse 2001; Zitzler 1999).

With the aid of Figure 2-31, genetic algorithm concepts are explained (Goldberg 1989a; Reeves 1995).

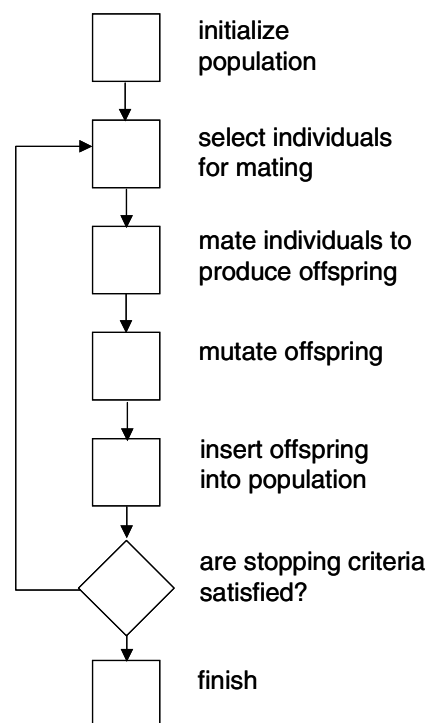


Figure 2-31 Basic scheme of genetic algorithms (Wall 1996)

Before initializing a population of candidate solutions, a form of representation for optimization variables has to be chosen. This is in contrast to other techniques, where variables are simple numerical values. The population generated contains solutions modeled as strings. Genetic operators work best with binary encoding. This representation has one drawback. Solutions, which are close to each other in a decimal representation, are not neighboring solutions in binary encoding. Numbers 15 and 16 could be used to illustrate this fact. Although they are adjacent, their binary representations (10000 and 01111 respectively) are not neighboring. It is impossible to “mutate” from 01111 to 10000 by changing only one bit. Gray scale encoding provides search space and representation adjacency.

The size of the initial population, that is the number of candidate solutions to begin with, is task dependent. An initial population is randomly generated according to the number and representation chosen. The objective function then judges the fitness of each population

member. A selection operator decides which population member should survive, mate and produce offspring. In many cases, tournament selection is employed. Two members of the population are chosen and the fitter one is copied into an intermediate population. This procedure is repeated for the remaining parents and then repeated once more assuring an intermediate population that is as big as the initial population. During crossover, members of the initial population are reproduced. As with selection, different operators exist. One point crossover chooses a random point in the representation of the chromosomes where crossover takes place. Two parents will only crossover when a generated random number is less than the parameter p_{cross} of the procedure.

Chromosome 1	111011	00100110110
Chromosome 2	111011	11000011110
Offspring 1	111011	11000011110
Offspring 2	111011	00100110110

Figure 2-32 Crossover

Crossover creates two children which are then inserted in the next population of the genetic algorithm.

Mutation randomly flips one bit in the generation of GA's. It prevents the algorithm from being caught in local optima as well as to search the neighborhood when the algorithm converges at the end to a near optimal solution. Bitwise mutation, which is often used, generates a random number for each bit of the chromosome and if this number is smaller than a predefined mutation probability p_{mut} , the bit is switched.

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

Figure 2-33 Mutation

The members generated by crossover and mutation form the new population. Elitism is the process of copying some of the good solutions of the old solution directly into the new population. Generally, this factor is kept very low (1-2%).

Genetic algorithms have been used to optimize structural design of frames (Bel Hadj Ali et al. 2002) as well as for the detection of structural damage detection (Chou and Ghaboussi 2001). Rossier (1994) applied it to the optimization of cable structures.

2.3.3.2 Choosing parameters for genetic algorithms

Parameters to be adjusted with genetic algorithms are population size, mutation probability, crossover probability and the number of generations. The population size has to be chosen such that the system equilibrates between premature convergence and waste samplings. It has been shown that the width of the genome used to represent one solution can be related to the population size needed for good solution space coverage. Nevertheless, this number turns out to be too big in most cases (Goldberg 1989b). Fortunately, empirical results suggest that a population size as small as 30 members is quite adequate in most cases (Schaffer et al. 1989).

Schaffer indicates as well, that in conjunction with a rather small population, a crossover rate between 0.75 and 0.95 and a mutation rate of 0.05 to 0.01 provide good results. Different sets of parameters have to be evaluated on the specific task for optimal choice.

Number of generations is an iterative parameter rather than to be seen in close conjunction with the other parameters explained. It has to be increased in the case the initial population does not evaluate to the optimal result needed, together with the population size.

2.3.4 Comparison of different algorithms

Comparison of different search techniques applied to the same task exist (Connor and Shea 2000; El-Beltagy and Keane 1999; Manoharan and Shanmuganathan 1999). Nevertheless, tests are not task independent. Wolpert and Mcready (1997) propose “no free lunch theorems” for optimization algorithms that do not use problem-specific tuning. Algorithms that perform well for one class of tasks do not necessarily produce good results for other classes. Generally, no one algorithm is best for all classes. Therefore, engineering studies are needed in a range of applications to determine most suitable match between algorithm and task.

2.4 Artificial intelligence methods for use in structural engineering

Computers are known to perform mathematical operations much faster than humans can. Nevertheless, humans outperform computers in tasks that can be found in areas such as:

- Games
- Natural language (understanding and translating)
- Pattern matching
- Most engineering tasks
- ...

Methods of artificial intelligence try to model human reasoning on computers and, thereby, link computational speed with human intelligence.

Opinions related to artificial intelligence are divided into two groups. One group hopes that these methods can be applied to liberate humans from routine tasks, while the other group fears that this might lead to replace humans entirely by computers. In the field of civil engineering, most effort has focused on the design task. Some AI systems have been a disappointment to engineers. Smith (2002) identifies reasons for this. One of the “bad ideas” stated is the missing interactivity of “intelligent” design tools. Another one is the maintenance that is problematic with knowledge-based systems. Maintenance cost might outweigh benefits.

Fenves draws an analysis why expert systems have not lived up to expectations (Fenves 1989):

- Knowledge engineering (extracting, compiling and organizing knowledge) is harder than anticipated.
- Systems are mainly suited for interpretative or diagnostic tasks and not for design and planning
- Knowledge-based systems have difficulties modifying their behavior based on their performance.

Nevertheless, modern systems, as presented by Miles et al.(1998) use software familiar to the designer and newly structured knowledge bases in order to overcome these problems. Concepts developed in the area of artificial intelligence are beneficially applied when clear task-solution descriptions exist.

Such a case is presented by Bruno et al. (1994). Due to practical limitations, the number of measurements as well as the number of active devices of on-orbit space structures is limited. These structures need, however, an accurate shape control for operation. He successfully uses neural networks to estimate the shape from incomplete measurements.

Chou and Ghaboussi (2001) employ a genetic algorithm to detect structural damage. Only a limited number of static measurements of displacements to identify properties of structural members is needed.

Examples, such as those mentioned above, encouraged exploring the potential of artificial intelligence in structural control. In this work, they are combined, with search techniques described before, to a framework that enables active structures to sense and react in uncertain environments. No previous work has been found that studies such a combination.

2.4.1 Neural Networks

The analogy used in neural networks is derived from human reasoning and involves the human brain. The human brain consists of roughly 10^{11} neurons which are highly interconnected. Incoming information is transmitted through this complex biological creation by electrical impulses. When enough impulses reach one neuron within a certain small time interval, this neuron sends out signals itself. Thereby, complex situations can be modeled (Pfeiffer and Scheier 1999).

Neural networks that are used in computation are much simpler. They establish a relation between an m-dimensional input vector and an n-dimensional output vector. The core component of a neural net is the neuron or node. Nodes are connected between the input and the output layer. Different topologies like feedback and feedforward connections exist.

In a feedforward network, the input layer is projected on the output layer but not vice versa. Feedback has connections pointing also in the direction of the input layer.

Neural networks can establish relations between data where no functional/analytical relation can be found. When analytical models exist, they might decrease the time needed to evaluate this model. This can be essential when the analytical model is used within an optimization process (Hajela and Berke 1991).

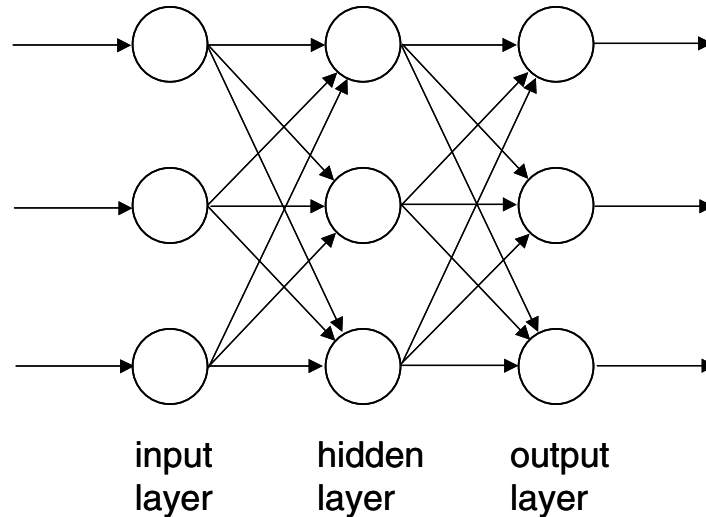


Figure 2-34 Principle elements of a neural network: input layer, hidden layer(s) output layers

The nodes of the one or more hidden layers and the output layer receive input from the previous layer. This input is multiplied with the weight, $w_{i,j}$, of each internodal connection and summed up. The total activation, x_j , of the node is then calculated by subtracting the internal threshold T_j from this sum.

$$x_j = \sum w_{i,j} \cdot i_i - T_j \quad (2-17)$$

x_j is passed to the transfer function, F , of the node which determines the final output, o_j .

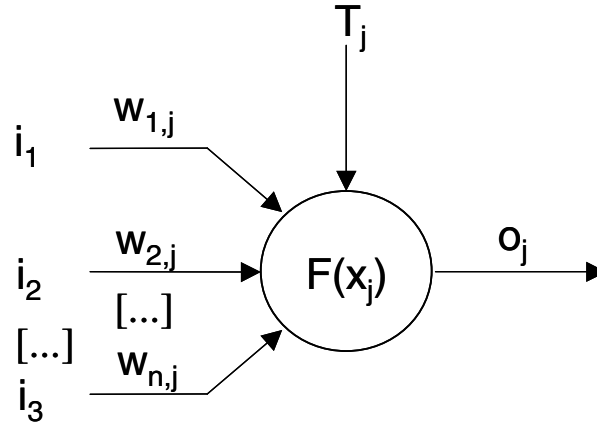


Figure 2-35 Node of the hidden layer

As transfer functions for neural nets, continuous and monotonic functions like the sigmoid function are chosen (Figure 2-35), (Equation 2-18).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2-18)$$

Learning in neural networks can be supervised or unsupervised. Commonly, supervised learning is used. During the training phase, the neural network is presented sets of known input/output patterns. The weights of the internodal connections are adjusted to match the desired output.

Adjustments may be made by propagating back the error, which has been evaluated between the output layer and the desired output. This technique is called backpropagation. The factor which governs the amount of change in each node of the network during the backpropagation is called the learning rate (η_j). A momentum factor can be introduced to avoid convergence to local minima during learning.

Since the sigmoid function is insensitive to input-values bigger than two, the input vector should be normalized to an interval of [0,1].

The number of nodes of the hidden layer can be determined after a formula given by Rogers (1994):

$$h = \frac{(p-1)m}{n+m+1} \quad (2-19)$$

- h number of nodes in the hidden layer
- n number of input variables
- m number of output variables
- p number of training pairs

More in-depth descriptions of neural networks are given in Haykin (1999) and Rojas (1996).

Artificial neural networks are useful in civil engineering (Garrett et al. 1997) and have multiple applications in the field. For example, they have been used to predict the deflections of beams which have been post strengthened by carbon fiber reinforced plastic (Flood et al. 2000) in order to aid engineers in the conceptual stage of the design process (Rafiq et al. 2000) and for structural optimization (Kaveh and Iranmanesh 1998). Rehak and Garrett (1992) envisioned the use of neural networks in structural control. Zagar and Delic (1993) studied neural-network control in the deflection of a bridge by predicting actuator commands. They are also used to predict the behavior of steelwork connections (Anderson et al. 1997). Chang et al. proposed online re-training for updating a finite-element model to real behavior of a bridge model tested under laboratory conditions (Chang et al. 2000).

Neural nets must be adjusted to the task that they are required to solve. This is done by choosing the network configuration, that is, the number of hidden layers as well as the number of nodes in the hidden layers. Training-data have to be prepared such that the network can fulfill its task after training. While choosing the correct configuration can objectively be judged by comparing the testing errors, the choice of the appropriate training data is a critical issue.

Generally, when more training data is used the results are better. Nevertheless, in most cases the number of data patterns is rather sparse. When the number of patterns appears to be sufficient, patterns might form groups, occupying only small regions of the solution space.

If too little data is presented, the network will give unsatisfactory results since the solution space is not adequately represented. If too much data is presented, the network will model too close to this data and, therefore, not be able to generalize. By specifying the ranges of the variables to be represented and searching for training data, which represent a hypercube (corners, center, midfaces etc.), training data will be evenly represented. If more points are needed, it would be possible to randomly choose points in this cube (Jenkins 1997). Jenkin's hypercube concept evaluates data to determine whether the solution space is covered to the greatest possible extent (Jenkins 1997). The minimal number of training data needed depends on the number of output nodes. For example, three output nodes are required for a three-dimensional hypercube. A 3D-cube can easily be visualized and consists of 27 significant points since there are corners, mid-sides, mid-faces and a center (Figure 2-36).

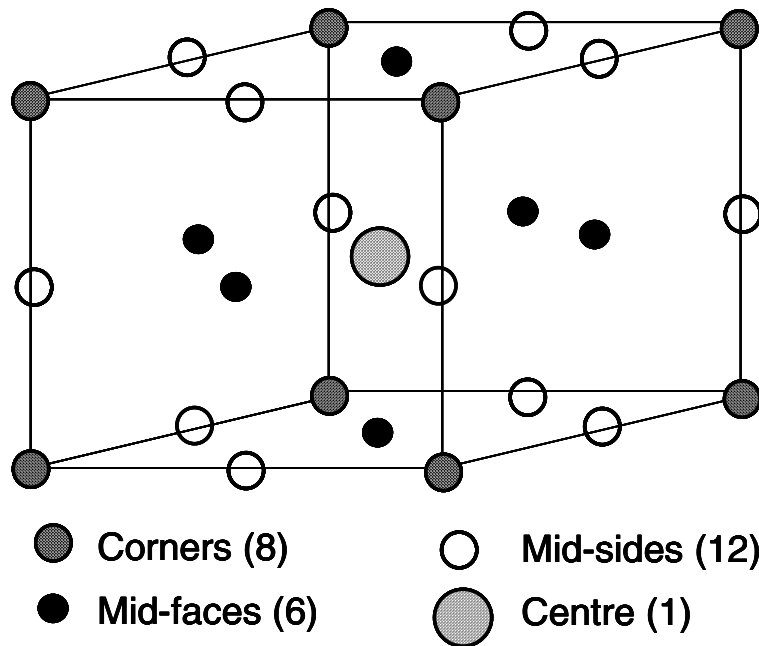


Figure 2-36 Hypercube (as presented in (Rafiq et al. 2000))

Rafiq (2000) adapted and tested the concept of a hypercube proposed by Jenkins. Another concept to choose training data for neural networks is presented by (Jingsheng Shi 2002). He proposes to detect clusters in data present and evaluate the network for each cluster.

2.4.2 Case-based reasoning

Case-based reasoning systems (CBR in the following) build on the observation that previous experience is useful. Humans solve new situations by first searching their memory for similar tasks they have successfully solved in the past. Retrieved solutions are then adapted (Kolodner 1993). The core component of case-based reasoning systems is the case-base, where past experience is stored. Case-base reasoning gave new energy to the application of artificial intelligence to design problems. An example is (Hua et al. 1996).

Cases are stored as pairs of task-solution descriptions. The interesting characteristic of CBR-systems is that by storing successful cases in the case-base, they improve performance over time. However, by storing more and more cases in the casebase, the administrative overhead increases and the system performance decreases without increasing competence. This issue is addressed in Section (2.4.2.2) “The utility problem”.

Leake (Leake and Wilson 1999) points out that the two basic prerequisites for the application of CBR are:

- Problem-solution regularity: solutions of prior problems are useful starting points for solving similar current problems
- Problem distribution regularity: the case library will contain cases similar to the new problem

Key tasks in the development of a case-based reasoning system can be divided into the following parts:

- Conception of the case-base
- Case retrieval, similarity measurement
- Case adaptation
- Case management

The conception of a case-base involves the choice of representation techniques and case memory organization. Several representation schemes are available. If cases are used for human browsing alone, text and image representations are sufficient. However, if cases need to be adapted automatically, representations that permit reasoning by the computer are needed. A simple representation involves a fixed set of attributes and values similar to that in a relational database. Object representations containing decompositions and abstraction hierarchies are also popular.

Case memory organization affects efficiency and ease of retrieval. A flat list organization is sufficient for relatively small case-bases. Hierarchical organizations improve the efficiency of retrieval when the size of a case-base increases. Clustering is another possibility to speed up retrieval. A more in-depth review of these techniques is provided by Kumar and Raphael (2001).

When the set of attributes is fixed and when a single value is possible for each attribute, cases could be stored in a relational database. This is advantageous because the RDBMS performs the task of efficient storage and retrieval.

2.4.2.1 Case retrieval and adaptation

The task description is used to discover similarities between present and already solved tasks stored in the case-base. The evaluation of efficient descriptive items can be considered as a knowledge-engineering effort. The retrieval procedure in a CBR system aims at selecting the most relevant cases to the current task through the assessment of similarity between the new situation and stored cases. Quality of solutions obtained depends on the retrieval procedure. Many methods exist, including k-nearest neighbor, user-defined retrieval rules and “case-based” case retrieval. The most widely used procedure is k-nearest neighbor retrieval according to which a distance between two cases X and Y is calculated:

$$Distance(X, Y) = \sqrt{\sum_{i=1}^d w_i^2 (x_i - y_i)^2} \quad (2-20)$$

W_i weight factor for the i^{th} attribute

Similarity is computed as a function of distance such that a distance of '0' corresponds to a similarity of 1.0. For example:

$$\text{Similarity}(X,Y) = 1 - \frac{\text{Distance}(X,Y)}{\text{Normalisationfactor}} \quad (2-21)$$

Recently, kernel functions are used to measure distances (Müller et al. 2001). Kernels have also been developed for applications involving non-numeric attributes such as image recognition and text matching. Currently, systematic procedures do not exist for choosing the best similarity metric for an application.

The adaptation process is one of the most challenging tasks in the development of a case-based reasoning system. When old cases have been retrieved from the case-base, they must be changed to meet the demands of the new solution.

Three adaptation methods are substitution, transformation and derivational analogy methods which have been summarized by Purvis and Pu (1995).

- *Substitution*: chooses and installs a replacement for some part of an old solution that does not fit in the current situation requirements.
- *Transformation*: uses heuristics to replace, delete or add old components to an old solution in order to make the old solution work in the new situation.
- *Derivational analogy*: uses the method of deriving the old solution in order to derive a solution in the new situation (Carbonell 1986).

The use of genetic algorithms to improve the task of case adaptation is proposed in (Gómez de Silva Garza and Maher 1999). Multiple similar cases are taken and treated with genetic operators such as crossover and mutation in order to generate solutions.

Smyth and Keane discuss adaptation-guided retrieval (AGR) (Smyth and Keane 1998). They state, that the presumption that the most similar case is the easiest one to adapt is sometimes wrong. Their proposition is to measure not only the similarity of the case, but also its adaptability.

They presented a system (Déjà Vu) that determines the adaptation requirements of cases during their retrieval. Adaptation specialists do the mapping between the retrieved case and the target problem. They are used to perform local modifications to cases and consist of two parts:

- The *capability part*: this part describes the parameters which can be adapted by this specialist
- The *action part*: this describes how the parameters should be adapted

Specialists are local and, therefore, ignorant of interactions. Adaptation strategies are used to solve conflicts between them. Each adaptation strategy repairs different types of intersection. This technique becomes relevant, when we take multiple conflicting control goals into account.

During the time of retrieval, several similar cases are presented to the adaptation algorithm as possible solutions. Cases where no adaptation specialist exists to adapt the nonmatching parameters to the target problem are filtered out of the space of possible solutions. This avoids the problem of unadaptable cases.

2.4.2.2 The utility-problem – case-based maintenance

At first sight the most attractive feature of case-based reasoning is its performance increases over time. There seems to be a linear relation between the size of the case-base and the competence of the system. The cost for the search for a case might dominate the usefulness of its adaptation. This effect has been named “utility problem” (Francis and Ram 1993). Theoretical considerations of the same authors lead to the conclusion that even massive parallel systems will not be able to increase efficacy.

Smyth and Keane (1995) established deletion policies, which should increase the system’s performance and preserve its competence at the same time. As they argue, classical deletion policies cannot be applied to case-based reasoning systems. The reason is that, in contrast to other systems, cases stored in a case-base are not equal. They define the concepts of coverage and reachability

- *Coverage* (of a case): the set of target problems it can be used to solve
- *Reachability* (of a target problem): the set of cases that can be used to solve this target problem.

These definitions are then used to divide cases into different categories:

- *Pivotal cases* provide coverage not provided by other cases in the case base. They empower the system with its basic competence and cannot be deleted without decreasing the quality of the case-base. A pivotal case is reachable by no other case but itself and should therefore not be deleted.
- *Auxiliary cases* only contribute to the performance of the system and may be therefore deleted if the saturation point of the case-base is reached. The coverage of an auxiliary case can be found in the coverage of one of its reachable cases, they can be deleted without affecting the competence of the system.
- *Spanning cases*: they link the problem regions that are independently covered by other cases. They do not affect competence directly, but may become important when auxiliary cases are deleted.
- *Support cases*: they exist in groups; the deletion of any one case does not affect the competence of the case-base at all. The deletion of the complete group, however, is analogous to the deletion of a pivotal case.

Categories are used to order cases by their competence: auxiliary cases, support cases, spanning cases and finally the pivotal cases that should be deleted as last cases. Deletion is started when efficacy of the case-base decreases. (Smyth and Cunningham 1996) shows an approach to measure system performance.

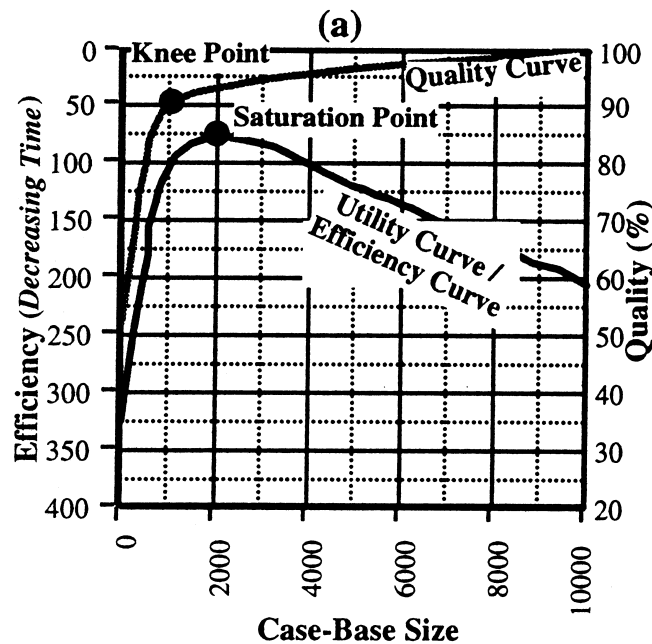


Figure 2-37 Case-base size vs. efficiency and quality (Smyth and Cunningham 1996)

Two distinct points are used in the figure above to characterize the performance of a CBR-system:

- The *Knee Point*: when the addition of new cases does not contribute to the solution quality of the case-base significantly
- The *Saturation Point*: when the addition of new cases decreases the efficiency of case retrieval

All this work led to an increased interest in how case-bases can be maintained such that utility problems are avoided. Leake and Wilson (1998) define case-base maintenance as follows:

“Case-base maintenance implements policies for revising the contents or organization of the case-base in order to facilitate future reasoning for a particular set of performance objectives.”

In the same paper, Leake and Wilson point out that case-deletion policies should also address changes in the problem-domain over time. Their strategy includes “snapshots” that trace changes in the case-base over time in order to discover trends. A framework for maintenance is presented in (Heister and Wilke 1998).

(McKenna and Smyth 1998; Smyth and McKenna 1998) provide means to rank the competence of case-bases. The contribution of competence of a single case in a dense group is lower than in a sparse group. They propose to compute case-density by the means of similarity. Competence groups are defined when coverage sets overlap. This means that the cases belong to the same competence group. It is stated that an extremely dense case-base could have one group that contains all cases. The other extreme is an extremely sparse case-base with one case per competence group. Experiments compared competence values calculated on a case-base basis with estimates obtained by the competence model presented in the paper Smyth/McKenna. The match is almost perfect. The model is even effective when case-bases contain redundant cases (competence of case-bases with non-redundant cases is almost linear to the case-base size).

Leake (2000) proposes to integrate performance criteria with case-addition and deletion policies. Aim is to assure competence and performance. Tasks may exist where such policies are not obvious to establish. Efficacy and competence of a case-base might also be assured by reducing the number of similarities calculated during retrieval. Wess (1993) decreases retrieval time by arranging cases in a tree structure (hierarchical case memory organization). Only relevant cases are considered during retrieval. A summary of recent developments in the field of case-based maintenance can be found in Leake et al. (2001).

Studies presented above concentrate mainly on one isolated part of the entire case-base reasoning process. This is problematic, since the process is highly interdependent. Challenges when transferring these results to create a complete system are numerous. It starts with selecting and ranking the importance of attributes used for case retrieval, to estimate how many cases are needed to build an operational system, which cases have to be stored in the memory, choosing a similarity metric such that cases which can be adapted are retrieved and involves the choice of an adaptation method.

2.4.3 Clustering

Classifying cases as proposed by Smyth and Keane (1995) to support case-based maintenance is not an easy task. The challenge begins with the choice of data used to classify or to group cases. Some parameters like the number of times a case has been retrieved as well as its success rate provide first criteria to what extent a case contributes to the overall knowledge of the system.

Clustering technologies provide useful algorithms for these tasks. A definition of clustering has been given by (Guha et al. 1998):

“Clustering problem is about partitioning a given data set into groups (clusters) such that the data points in a cluster are more similar to each other than points in different clusters.”

Clustering algorithms need a proximity *measure* to describe the similarity of two data points (or feature vectors) and a *clustering criterion*. The clustering criterion helps to decide which number of clusters, k , to choose for a given proximity measure and data sets, since k needs to be chosen by the user. k is fixed after testing multiple different values on the data set and evaluating the clustering criterion for each of them.

As clustering is of particular interest in computer science, a huge number of different algorithms have been developed. In the scope of this thesis, partitional clustering, which decomposes one data set into a set of disjoint clusters, will be used. For a broader overview of clustering techniques refer to (Halkidi et al. 2001). In particular, K-means is employed (Anderberg 1973). MacQueen used the notion of K-means to denote the process of assigning each data unit to that cluster (of K-clusters) with the nearest centroid (mean). The algorithm is described as pseudocode in Figure 2-38.

```

Take the first  $K$  data units in the data set containing  $m$  data sets as
clusters of one member each
Loop 1: Repeat for the remaining  $m-K$  data units
    Assign each data unit to the cluster with the nearest centroid
    Recompute the centroid of the gaining cluster
End of Loop 1:

Loop 2: Repeat for all  $m$  data units
    Take the computed centroids as fixed seed points and assign the
    data units now to the nearest seed point.

End of Loop 2:
    
```

Figure 2-38 K-means clustering algorithm

Since the number of clusters k is unknown in advance, a methodology is needed to provide a sort of quality measurement of the chosen number of K . (Legendre 2001) proposes the Calinski-Harabasz (C-H) criterion.

$$C - H = \frac{\left[R^2 / (K - 1) \right]}{\left[(1 - R^2) / (n - K) \right]} \quad R^2 = \frac{SST - SSE}{SST} \quad (2-22)$$

SST : total sum of squared distances to the overall centroid

SSE : sum of squared distances of the objects to their own centroids

The criterion has to be calculated for different values of K . Large values indicate a good clusterization.

2.5 Conclusions

This chapter intends to give a critical review of research activities in domains that are covered by the objectives of this work. Tensegrities provide interesting opportunities to advance research in active structural control. Although control proposals exist, they neglect non-linear behavior and have not been validated on full-scale structures. Models proposed risk to be not precise enough for some tasks. Potential increases in accuracy obtained by an over-conditioned simulation might reveal additional challenges when interdependencies between parameters do not improve model quality. Combining analysis with neural networks seems to be promising to provide a compromise between complexity and accuracy.

No mechanisms to adapt changing control objectives and environments are provided. Simulated annealing has proven to be successful in finding good control commands and equally addresses this issue (Fest 2002). Other techniques with different ranges of applicability are available. A comparative study of different algorithms in active structural control has not been done before. New techniques, like PGSL, might show advantages when compared with established ones.

It is now possible to build the foundations for the use of AI in structural control. More specifically, this work will serve to advance the areas of structural design, infrastructure maintenance, intelligent control, and more generally, computational techniques for civil engineering. Structures are often thought of as large static objects that are over-designed to withstand extreme environmental conditions. With the incorporation of continuous intelligent control, a structure becomes a dynamic object that can adapt to a complex environment.

3 Computational model

3.1 Tensegrity structure at IMAC

The following section starts with a description of the tensegrity structure designed and constructed at the Applied Computing and Mechanics Laboratory (Laboratoire d'informatique et de mécanique appliquées à la construction, IMAC) of the Swiss Federal Institute of Technology in Lausanne (EPFL). A computational framework for active structural control is presented. The computational model chosen (dynamic relaxation) is validated with measured results. A hybrid system accounting for inaccuracies by correcting results of dynamic relaxation with neural networks is tested. Since the active structure has been developed in the scope of another thesis (Fest 2002), descriptions given are limited to the amount needed to describe contributions of this work.

A summary of research issues that have been identified and addressed in this chapter are given below:

- How can accuracy of the computational model used be increased without adding complexity?
- Influence of methodology employed for the selection of training data on the quality of the hybrid model presented.
- Can structural behavior be modeled by neural networks alone?

3.1.1 Full scale tensegrity structure

The assembly of IMAC's tensegrity structure follows a modular concept inspired by an initial design of Passera and Pedretti, Lugano (Passera and Pedretti 2002). Each module consists of 24 cables and six bars. Bars meet in the center of a module at the central node. Cables are distinguished by their lengths:

- Cables forming the small triangle
- Cables forming the big triangle
- Lateral cables which link upper and lower layers of the structure (Figure 3-1)

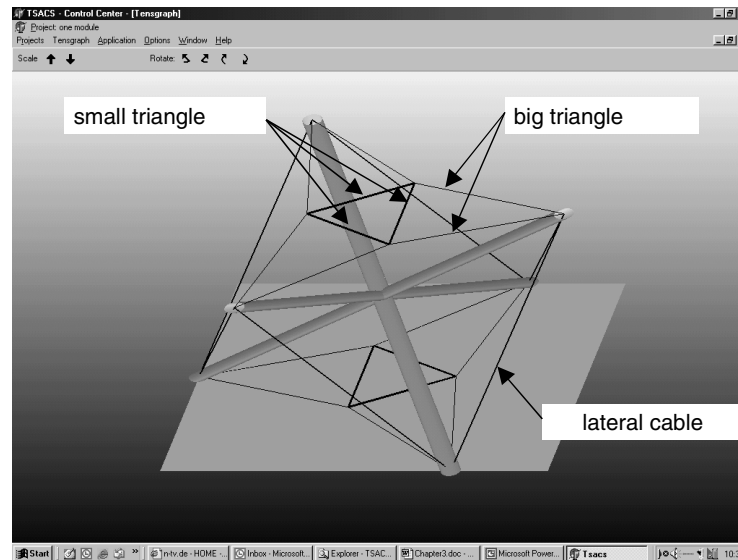


Figure 3-1 Tensegrity module (screenshot from the Tensegrity Structure Analysis and Control Software)

Materials chosen are:

- Stainless steel for cables (Table 3-1)
- Fiber reinforced polymer for bars (Table 3-2)

Ø cable [mm]	Area A [mm ²]	Modulus of elasticity E [Gpa]	Specific weight [Kg/100m]	Ultimate tensile force [kN]
6	13.85	115	13	17

Table 3-1 Material properties of cables (Fest 2002)

Specific weight [Kg/m]	Area A [mm ²]	Modulus of elasticity E [Gpa]	Ultimate force [kN]
1.7	703	23-38	42

Table 3-2 Material properties of bars (Fest 2002)

Although using a central node halves the buckling length and allows using a slender profile for the bars, it creates a complex joint. Tensegrity purists might argue that this detail might contradict the definition of a tensegrity structure. Nevertheless, bars between modules are not connected.

Designing nodal connections for tensile structures is challenging. For the type of tensegrity module presented in Figure 3-1, four elementary nodal types have been identified:

- Cable/cable connection
- Cable/strut connection (edge)
- Cable/strut connection (center)
- Central node

A modular node, which can be used for all possible connections of the first three types, has been designed (Figure 3-2). Every node fulfills the condition that links attached intersect in the same virtual point in 3D space (Figure 3-2a).

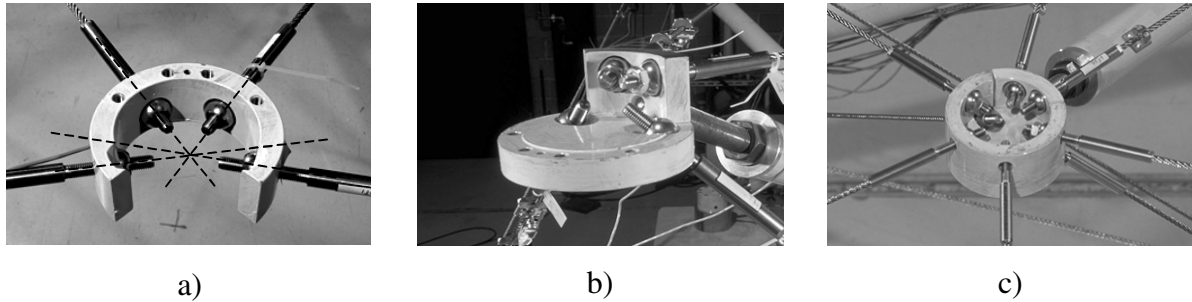


Figure 3-2 Modular nodal connector (Fest 2002)

The cable/cable connection (Arc 264° , Figure 3-2a) deforms with changing tension in the attached cables. As a result, tensile forces no longer intersect in the same virtual point. This deformation affects the accuracy of the behavioral model employed (Figure 3-3a).

In an enhanced version, rings tie the node together to limit this effect (Figure 3-3b).

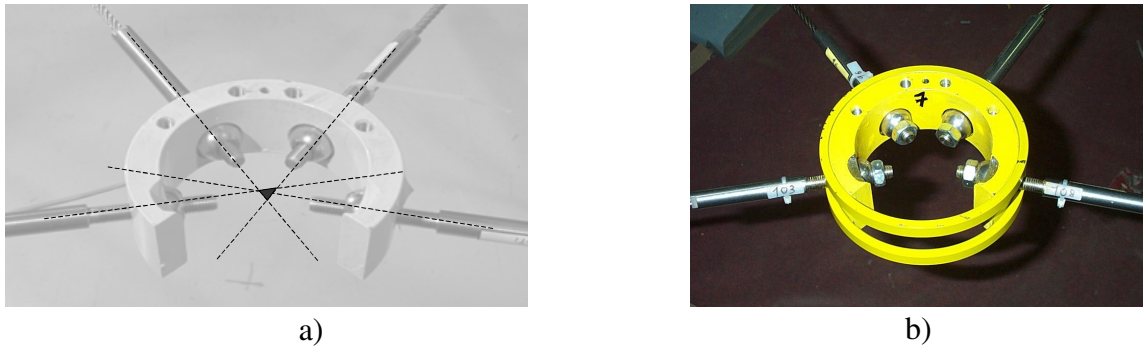


Figure 3-3 Open arc with rings (Fest 2002)

Within a period of several years, design of the central node changed multiple times (Figure 3-4). Development can be classified into three stages:

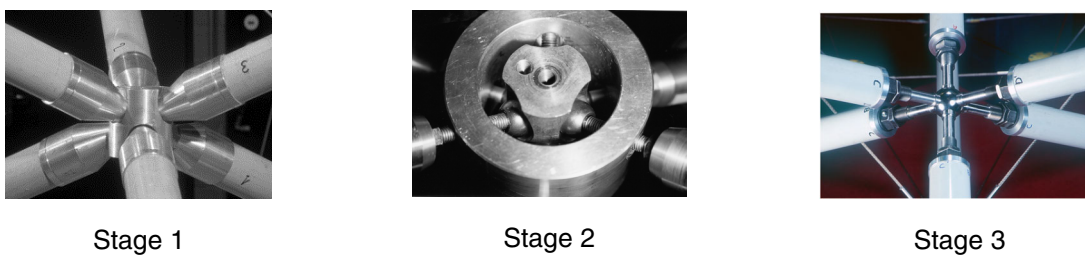


Figure 3-4 Three stages of nodal development (Fest 2002)

Stage 1 nodes had cone shaped openings where the ball pointed ends of the bars fitted in. This design was extremely sensitive to fabrication inaccuracies: compressive forces did not intersect in the same hinge but slightly eccentrically. Loading increased this effect. In the worst case, the structure became unstable and collapsed. The intermediate node version (Stage 2) refined the design. Eccentricities are less severe; nevertheless, they have not been eliminated. The stage 3 node design is entirely new. It reverses the principle applied for the former designs: it uses a sphere as a node and struts have cone-shaped ends. During module assembly, this system almost automatically centers the sphere. Under load, struts can slide on

the sphere and allow auto-centering. Instabilities, as they have been observed with two preceding designs, have been eliminated thereby.

Tensegrity modules may be combined to tensegrity structures. The first structure constructed at IMAC consisted of three modules (Figure 3-5).

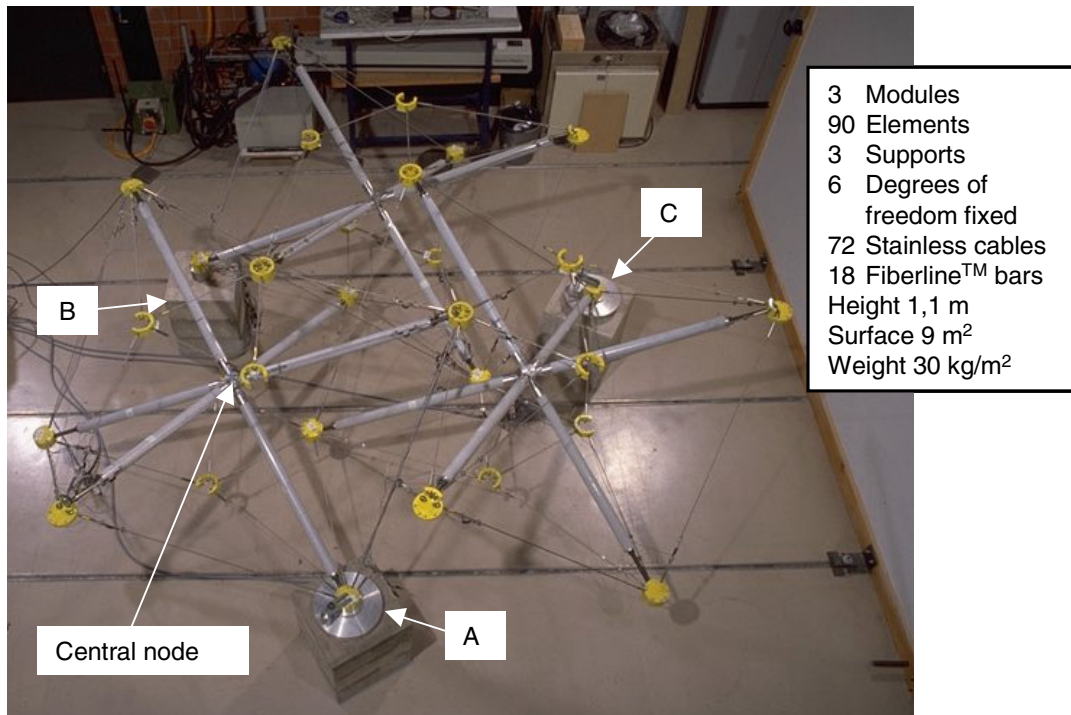


Figure 3-5 Three module tensegrity structure, A, B and C are supports (Fest 2002)

Bars of the modules are telescopic and can be extended or contracted. This changes the amount of self-stress and, thereby, the shape of the structure. First versions of the bars were controlled manually. The present configuration uses electrical jacks (Figure 3-6).

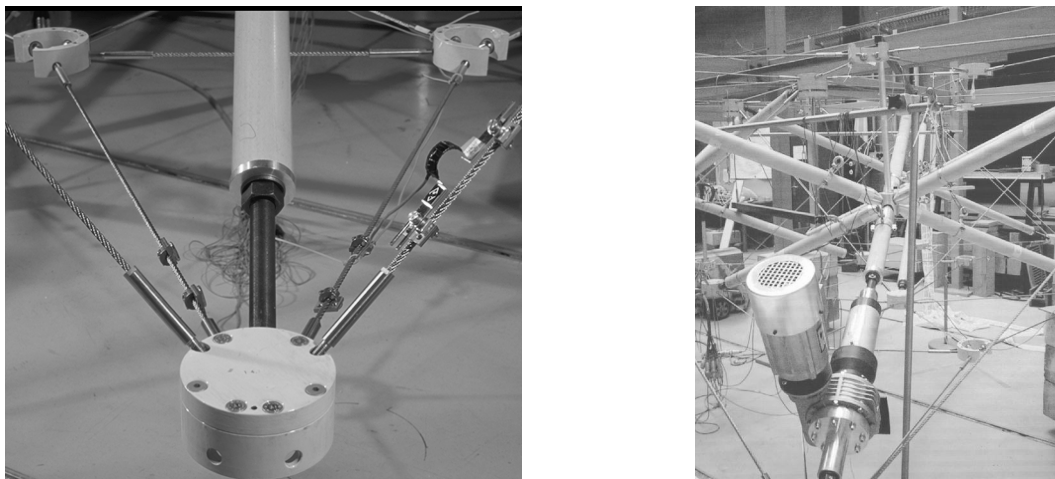


Figure 3-6 Telescopic bar (left: manual, right: electrical) (Fest 2002)

3.1.2 Five-module structure

After intensive testing, two more modules have been added to the three-module structure. These two additional modules are only supported by surrounding modules and provide a more realistic scenario (Figure 3-7).

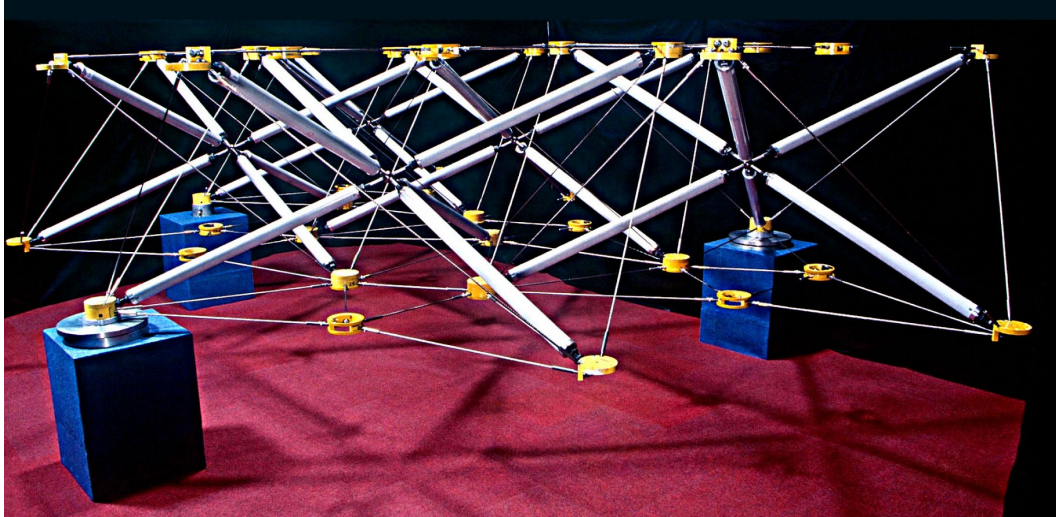


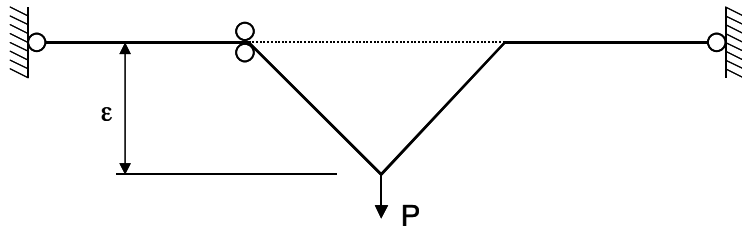
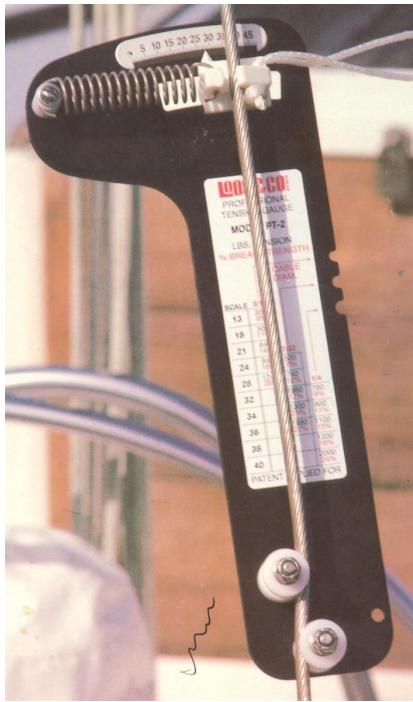
Figure 3-7 Five module tensegrity structure (Fest 2002)

Further modifications compared to the initial structure have been made:

- Central nodes have been upgraded to stage 3 type nodes
- Arc 264° type connectors have been tied together with rings

3.1.3 Measuring cable forces

Measuring cable forces has to account for initial tension. Mechanical devices used for this task have been discussed in the context of Section 2.1.1.1. Accuracies have been evaluated to 15% for the contact arm and 3% for the dial gauge. Although the dial gauge presented was used with small-scale models, the principle is also applied to thicker cables with a diameter of 5-7mm. The tension gauge (Figure 3-8) has its origin in sailing where it serves to adjust tension in shrouds and, thereby, trims the form of the sails. Figure 3-8 also explains how cable tension T and deflection ϵ caused by the integrated spring are related. Device accuracy is around 3%.



$$\text{Cable tension } T = A \cdot \epsilon \cdot E$$

A = cable section

ϵ = deformation as indicated by the gauge

$$E = 21.E^4 \text{ N/mm}^2$$

Figure 3-8 Tension gauge, mechanical principle

Baumann (2001) examined alternative methods, such as laser measurement. In practice, laser measurement is applied to:

- Suspension and cable stayed bridges
- Identifying eigenfrequencies of a cable

This technique has an error rate of approximately 15% and is mainly used to detect fatigue problems. An application for cables with diameters as they are used with the tensegrity structure is rare.

3.2 Geometry and topology

The shape of tensegrity structures is usually found in a separate process, called form finding. In general, the force density method (Section 2.1.1.2) is applied. In contrast to this, the geometry of IMAC's structure may be calculated beforehand, applying simple trigonometric functions. Descriptive parameters for the geometry of one module are:

- r_{gt} (radius of the “grand triangle”),
- r_{pt} (radius of the “petit triangle”) and
- h (height).

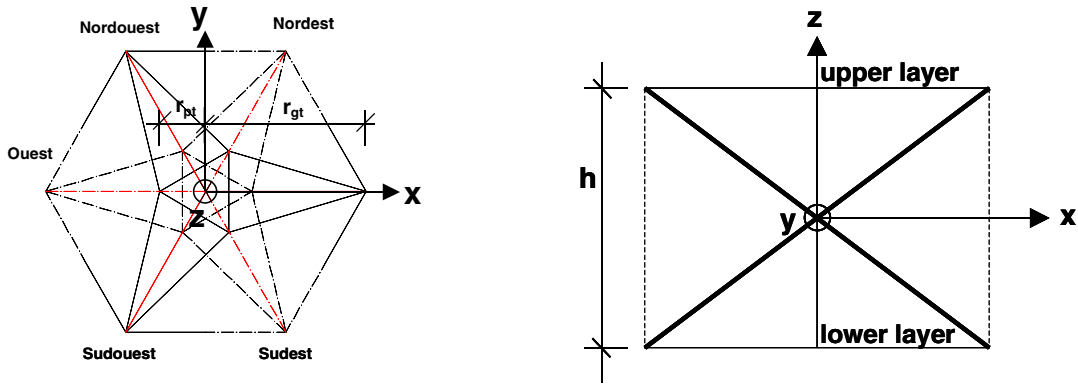


Figure 3-9 Basic parameters, directions and the coordinate system of the tensegrity module

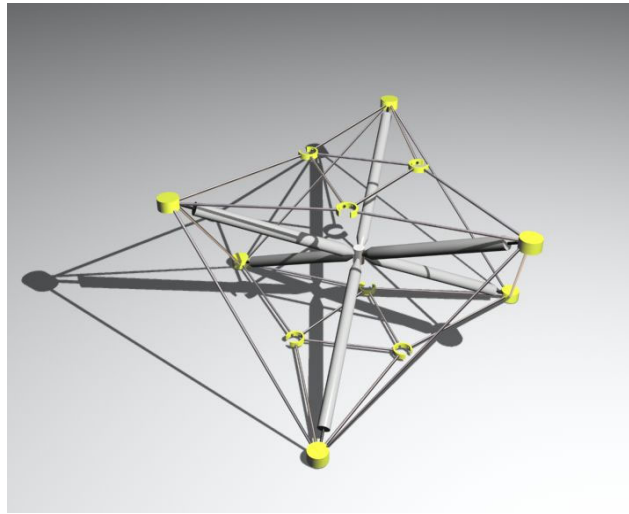


Figure 3-10 One module

Applying now equations presented in section 2.1.3 should give insight in mechanical behavior of IMAC's type of tensegrity structures. Geometric properties are sufficient for a first study of one module. According to Calladine, applying Maxwell's rule should result in a failure of the stability test (Calladine 1978) The rule states that

$$3 \cdot j - 6 \quad (3-1)$$

bars are needed for structural stability, where j = number of joints assuming a statically determinate structure in three dimensions. Applying this to one IMAC-module, we calculate that

$$3 \cdot 13 - 6 = 33 \quad (3-2)$$

bars are needed to obtain static stability with Maxwell's rule. In these modules, however, only 30 bars and 13 joints provide static stability.

Assembling the equilibrium matrix **A** results in a rank of 27. Thus, the number of independent self-stress states is calculated using Equation 2-11:

$$s = m - r = 30 - 27 = 3 \quad (3-3)$$

The number of mechanisms for one module is calculated using Equation 2-12:

$$q = n - r = 3 \cdot J - r = 30 - 27 = 3 \quad (3-4)$$

Since J is the number of non constrained joints. Initial pre-stress is applied to the structure when the telescopic bars are extended from their initial position. The notion of "2 mm pre-stress" means that all the bars have been uniformly extended by 2 mm more than their nominal initial length.

3.3 Tensegrity structure analysis and control software (TSACS)

The modular concept provides potential to construct large scale structures where the number of modules can vary from two or three to a couple of hundred. Generating and administrating data needed for analysis and control of such structures demands for computational support. This support is offered by database systems, which have clear advantages over other storage types. The architecture of the Tensegrity Structure Analysis and Control Software (TSACS) builds on this approach (Figure 3-11).

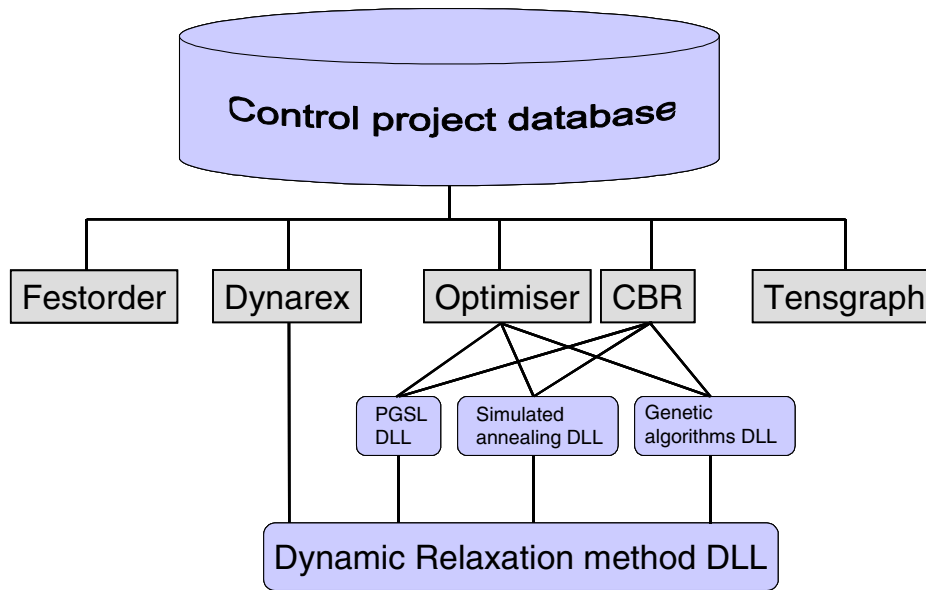


Figure 3-11 System architecture of TSACS; DLL: Dynamic link library

Five modules (rectangular boxes in the middle of the figure) form the core of the application. They communicate via the central database:

- Festorder: generation of geometry and topology data
- Tensgraph: visualization of the structure's shape
- Dynarex: form-finding and structural calculation of structures stored
- Optimiser: search for good control commands with stochastic search
- CBR: improving system's behavior over time with case-based reasoning

DLL's (boxes with round shaped edges in the lower part of the figure) are used whenever functionality has to be provided to more than one software module. They contain stochastic search algorithms (PGSL, simulated annealing, genetic algorithms) as well as the method used to calculate structural behavior (dynamic relaxation). Search employs dynamic relaxation for the evaluation of the objective function. Techniques have been discussed in Chapter 2. Modules Festorder, Dynarex and Tensgraph are closely related to the computational model and therefore presented in this chapter. Module Optimiser will be discussed together with the comparison of stochastic search techniques in chapter 4; the case-based reasoning module (CBR) will be introduced in Chapter 5 with tests related improving structural performance over time.

3.3.1 Module Festorder

Although calculating geometry and topology of a structure assembled with IMAC-type modules does not necessarily require computational support, generating these data “by hand” is lengthy and bears potential for mistakes. This was the motivation for implementing an algorithm for automatic geometry and topology data generation. Data generated are stored in the control project database of TSACS.

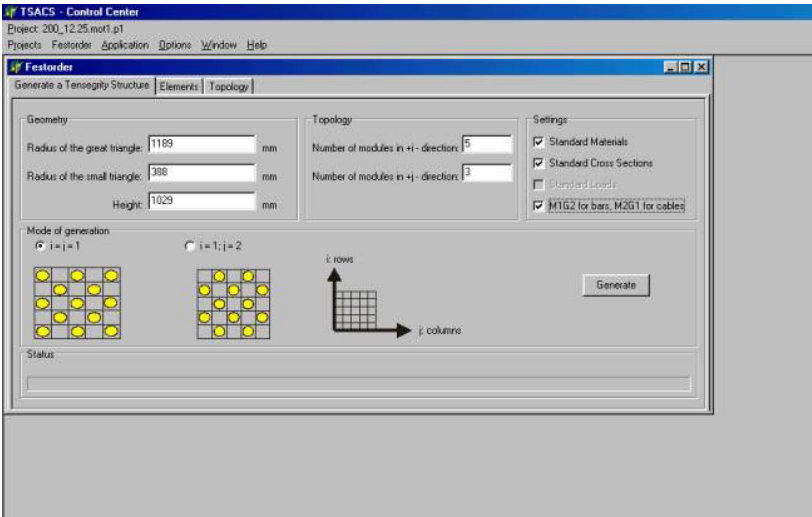


Figure 3-12 Screenshot “Festorder”

Besides the basic geometry parameters (r_{gt} , r_{pt} and height), the module arrangement has to be defined by introducing a generation mode as well as the number of modules in each direction (Figure 3-12). Information can be visualized. The graphical interface of Festorder lists node and link numbers as well as the module position within a multi-module assembly for a given module (Figure 3-13).

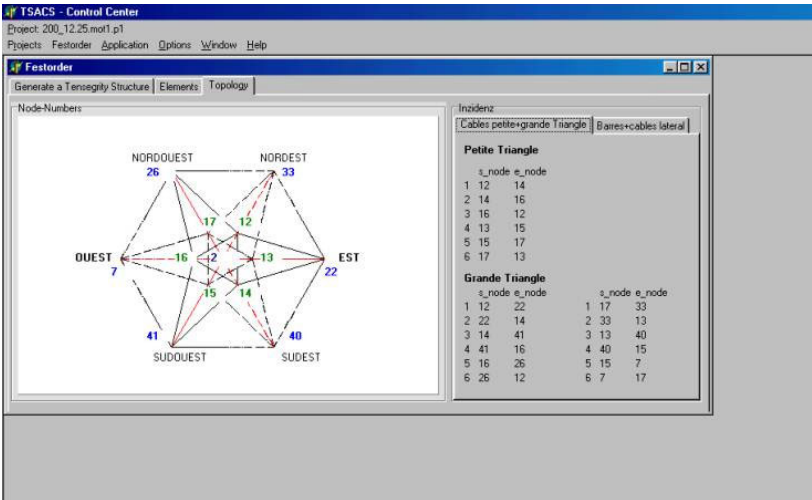


Figure 3-13 Screenshot of link and node numbering

3.3.2 Module Tensgraph

Tensgraph is a first three-dimensional visual control of data generated by Festorder. The model can be rotated or scaled. Colors and line-thickness can be chosen in a separate dialog-window. Lighting and shading are additional implemented features. The program employs OpenGL libraries.

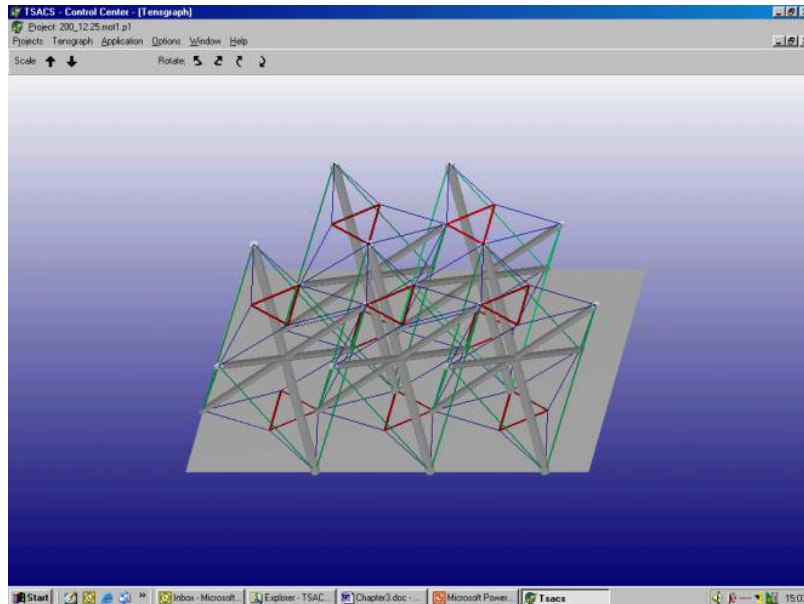


Figure 3-14 Screenshot of Tensgraph

3.3.3 Module Dynarex

“Dynarex” (**D**ynamic **R**elaxation) reads data prepared by module “Festorder”. The user interface allows entering additional data needed for structural analysis: nodal loads, materials and cross sections of cables and struts, etc. Dynarex generates an input file for structural analysis.

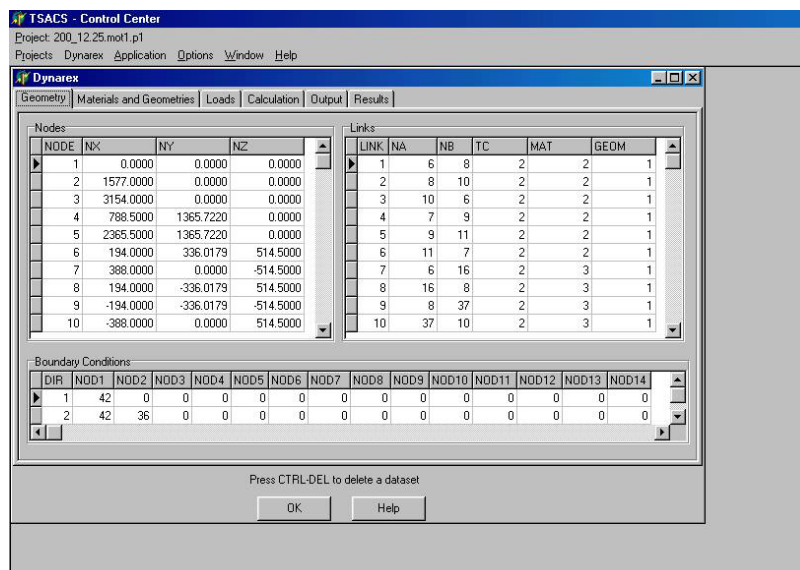


Figure 3-15 Screenshot of Dynarex

Analysis employs the dynamic relaxation method implemented by Rossier (1994). It has been converted to a dynamic link library (DLL). The resulting internal forces and stresses as well as nodal displacements are read into the database and can be accessed by the user interface of Dynarex.

3.4 Improvement of the computational model

3.4.1 Motivation

Perelli showed that the dynamic relaxation method correctly models the behavior of IMAC's tensegrity structure (Perelli 2000). Input parameters for the calculation have been determined according to the materials used through independent testing (Fest 1999). A discrepancy between theoretical calculations and measured behavior has been observed (Figure 3-16). Although this might be acceptable for isolated calculations, errors may accumulate throughout a sequence of control commands under active control.

Node friction, cable relaxation, node displacement and changing environmental conditions are possible causes for differences between measured and predicted behavior. Values of these parameters are difficult to determine accurately (Fest et al. 2002). Neural networks show potential to increase the accuracy of the dynamic relaxation method since they are able to model non-linear relationships and are known to be efficient when applied to time-variant systems (Garrett et al. 1997).

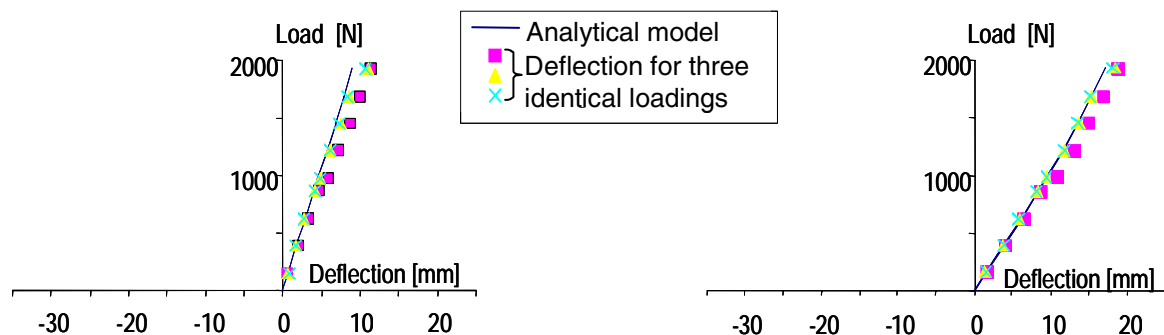


Figure 3-16 Example for accuracy discrepancy between simulated and measured results (left: node 52, right: node 63) (Fest 2002)

In contrast to other applications of artificial neural networks in civil engineering, the approach presented is hybrid. Instead of replacing analytical models completely, the network is used as an additional error correction step in the evaluation of nodal displacements under a given load.

3.4.2 Artificial neural network software applied

A wide choice of software exists for creating and testing neural networks. A particularly useful tool is the Stuttgart Neural Net Simulator (SNNS 2001). A graphical user interface helps design the network and offers a choice of learning methods and activation functions.

For more sophisticated tasks, a batch language may be used. The batch processor “batchman” processes small programs. Within the scope of this project, the graphical interface of SNNS

has been used to create network topologies. “Batchman” was employed for training and testing the networks.

3.4.3 Test description

The primary focus in testing was to determine whether using a neural network could increase the accuracy of the dynamic relaxation method when used alone. Internal complexity of a neural network (number of hidden layers, numbers of nodes in one hidden layer) as well as the amount of training data needed is linked to the number of input- and output variables (Equation 2-19). To be more precise, correcting all nodal deflections obtained by dynamic relaxation would require huge amounts of training data and a complex network geometry. This is not economical. The approach discussed here will correct deflections of nodes employed for the evaluation of the shape control objective.

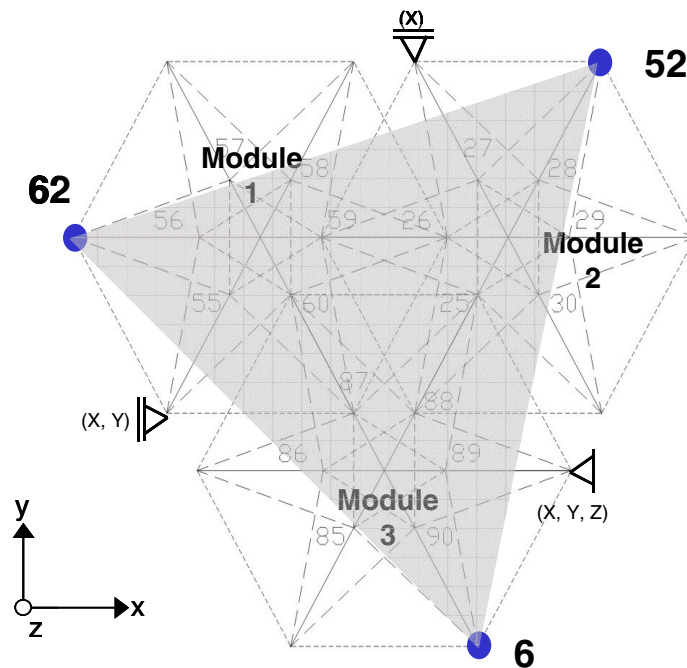


Figure 3-17 The control objective involved maintaining the slope constant between nodes 6, 52 and 62.

As a working control objective, maintaining the upper layer of the structure at a constant slope has been chosen as an initial research task (Figure 3-17). The slope is controlled by measuring the z-displacements of three nodes on the upper layer (nodes 6, 52 and 62 in Figure 3-17) and adjusting the telescopic bars such that they counteract displacements. Previous tests measured the displacement of these nodes. (Perelli 2000).

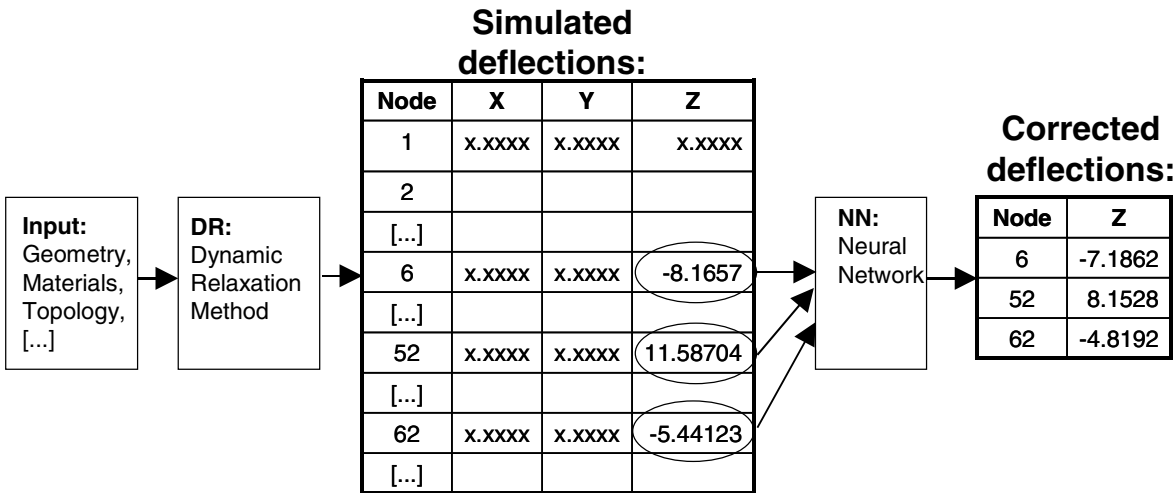


Figure 3-18 Hybrid approach

Ninety pattern sets consist of displacements calculated using the dynamic relaxation method and three measured displacements. Simulated displacements are input patterns; measured displacements are output patterns.

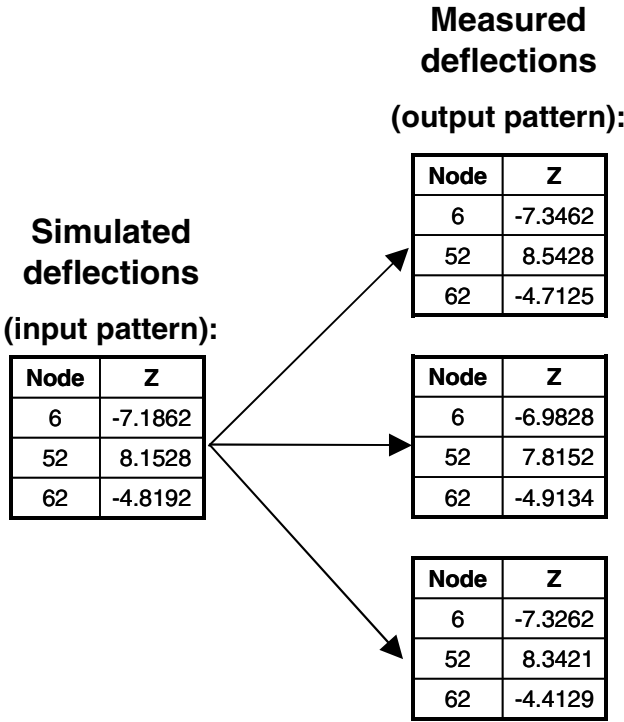


Figure 3-19 Pattern sets for one load case

Magnitudes and locations of the loading that lead to these pattern sets are presented in Table 3-3 and Figure 3-20. Load cases are subdivided into three classes that are named as follows: symmetric, asymmetric and central joint loading. Symmetric loading applies the same load magnitude to nodes whose combined center of gravity corresponds to the center of gravity of the structure. Asymmetric loading involves one edge node at a time and central joint loading means that the central joints of each module are loaded. Since the neural network is ultimately intended to be used for active control, results from a range of pre-stress levels were employed. Measurements were taken three times to reduce uncertainties. This results in a total number of 270 pattern sets.

Type	Node loaded	Pre-stress [mm]	Vertical loads applied [N]
Symmetric	6, 52, 62	2	152, 388, 623, 860
	5, 48, 61	2	152, 388, 623, 860
Asymmetric	6	2	152, 388, 623, 860
	52	2	152, 388, 623, 860
	62	2	152, 388, 623, 860
	5	2	152, 388, 623, 860
	48	2	152, 388, 623, 860
	61	2	152, 388, 623, 860
Central node	7	2	152, 388, 623, 860, 981, 1216, 1452, 1687, 1923
	54	2	152, 388, 623, 860, 981, 1216, 1452, 1687, 1923
	63	2	152, 388, 623, 860, 981, 1216, 1452, 1687, 1923
	7	3	388, 860, 1216, 1687
	54	3	388, 860, 1216, 1687
	63	3	388, 860, 1216, 1687
	7	4	388, 860, 1216, 1687
	54	4	388, 860, 1216, 1687
	63	4	388, 860, 1216, 1687
	7, 54, 63	2	152, 388, 623, 860, 981, 1216, 1452

Table 3-3 Magnitudes and location of the loading

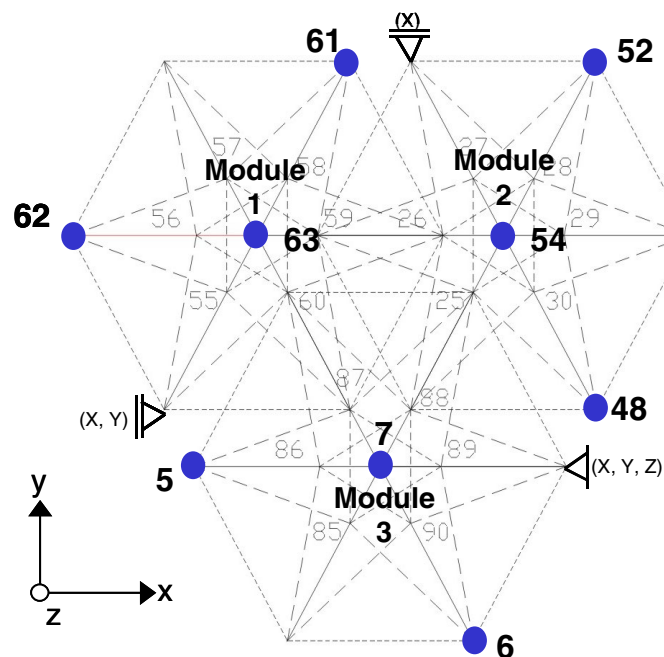


Figure 3-20 Node numbers of loaded joints, corresponding to Table 3-3

The goal of this work was to increase the accuracy of calculating deflections at three nodes. This refers to the control objective “slope control”. This goal thus fixed the number of input and the number of output nodes of candidate networks to three each. The number of nodes in the hidden layer was evaluated initially for twelve network topologies having zero, one and two hidden layers. The results of these tests indicated that four topologies had potential and they were subsequently employed for the main testing phase. These topologies, ranked by their test error, were

- 3-8-3
- 3-12-3
- 3-10-10-3
- 3-14-14-3

All networks used were feed forward and used the sigmoid transfer function. The learning rate was set to $\eta = 0.2$. No momentum factor has been introduced.

Throughout all tests, pattern sets were subdivided into training, testing and unseen patterns. As their name indicates, training patterns were used to train the network. The weights of the network were changed after each training cycle to minimize the training error. After every 100 training cycles, the test patterns were presented to the trained network and the test error was evaluated. Weights were only saved when the testing error decreased. It was observed that training errors decreased continually but test errors started to increase after some time. This is due to the fact that neural networks may over correlate training patterns when they have been subjected to too many training cycles. The test error was used to decide which network topologies should be tested with the unseen patterns. Unseen patterns were used after training and testing of the network to check overall generality.

A batch program was written using the Stuttgart Neural Net Simulator (SNNS) batch interpreter for training and testing the four network topologies. One-layer networks were trained for 500,000 cycles and the two layer networks for 1,500,000 cycles.

The error used is the sum of square error (SSE), which is equal to the sum of the square of the difference between normalized simulated and normalized targeted values.

In addition to primarily focusing on determining potential accuracy enhancement, two other applications of neural networks have been tested:

- Examining the possibilities of online-training to adapt the neural net to changing loads and environmental conditions.
- Compare correcting the dynamic relaxation method results with complete replacement by a neural network.

3.4.4 Enhancing the Accuracy of the Dynamic Relaxation Method

3.4.4.1 Using measured results

A total number of 270 patterns were available. The usual experimental procedure of eliminating unrealistic data resulted in the deletion of 30 patterns. The remaining 240 have been subdivided into three groups as follows:

- Training patterns: 150
- Test patterns: 39
- Unseen patterns: 17 (.3)

Training of the four network topologies and subsequent testing reveal test errors that are given in Table 3-4. Finally unseen patterns are tested against tripled targeted values allowing 17 comparisons. This has been carried for the 3-12-3 and the 3-14-14-3 topologies and results are shown in Figure 3-21. In this figure, deviations of simulated values from the targeted measured values are compared by calculating ratios of results when using neural networks (DR+NN) for an additional error correction step and when using only the dynamic relaxation method (DR) for simulation.

Network	Test Error (SSE)
3-8-3	8.10E-02
3-12-3	7.98E-02*
3-10-10-3	8.09E-02
3-14-14-3	7.87E-02*

Table 3-4 Comparison of four network topologies. The two best networks (3-12-3) and (3-14-14-3) are studied further in Figure 3-21

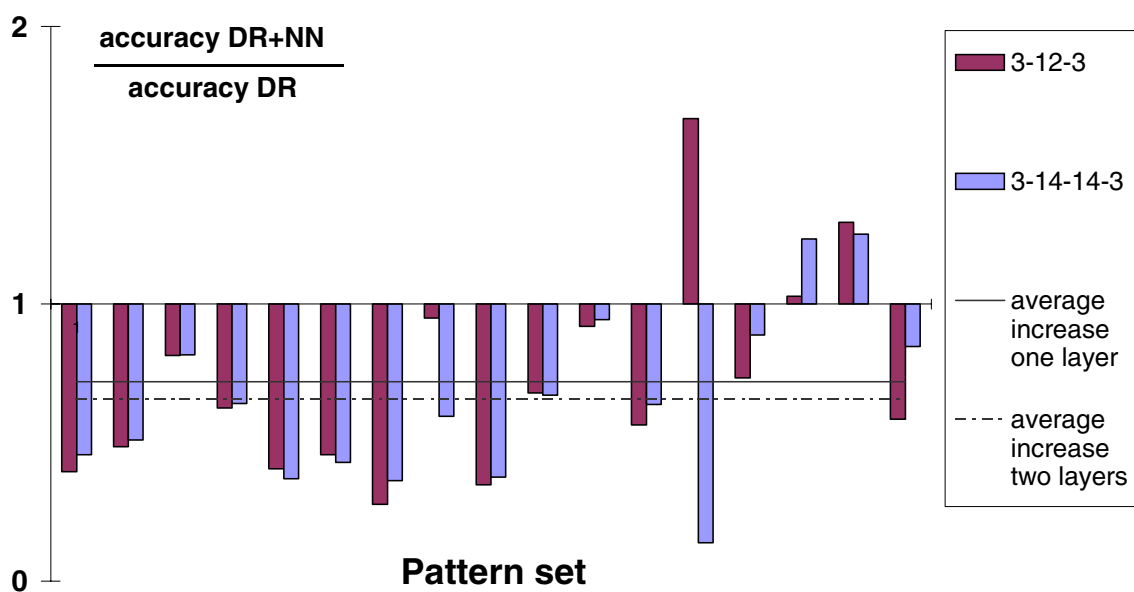


Figure 3-21: Accuracy loss for the 3-12-3 and the 3-14-14-3 topology

Values above “1” represent, therefore, an increase; values below “1” represent a decrease in accuracy when using a neural network.

The best network (3-12-3) only increases accuracy in 3 out of 17 cases (13,15,16). In all other cases, there are losses in accuracy. It can be concluded that training the networks with these data sets does not contribute to the overall accuracy of the simulation. This observation also is clear when values of the average increase in accuracy are calculated (continuous and dotted lines in Figure 9) since they are less than one.

3.4.4.2 Using the average of the measured values

Preceding tests employed pattern sets where one load case was represented by three patterns, since displacements have been measured three times for each loading. Simulated patterns did not change within the three measurements (Figure 3-19). It was assumed that the neural network averages these values during training. Figure 3-21 demonstrates that this approach did not give encouraging results. For the following test series, averages of the three measured values have been calculated before training.

This resulted in a total of 80 patterns and these were subdivided into 50 training, 13 test and 17 unseen patterns. The evaluation of the four network topologies identified the 3-8-3 and the 3-10-10-3 networks (Table 3-5).

Network	Test Error (SSE)
3-8-3	5.80E-03
3-12-3	6.05E-03
3-10-10-3	5.24E-03
3-14-14-3	5.79E-03

Table 3-5 Comparison of four network topologies trained with the average of data triples

Using the same schema as in Figure 3-21 to present the results, Figure 3-22 shows the accuracy enhancement for the 3-8-3 and the 3-10-10-3 topologies.

Figure 3-22 shows the evaluation of the unseen patterns. The situation has changed drastically: now there is only one data set out of 17 (in the 3-10-10-3 configuration) which shows a decrease in accuracy. The decrease of accuracy (15.75%) is a tolerable value, which is not expected to affect the stability of the structure in practice.

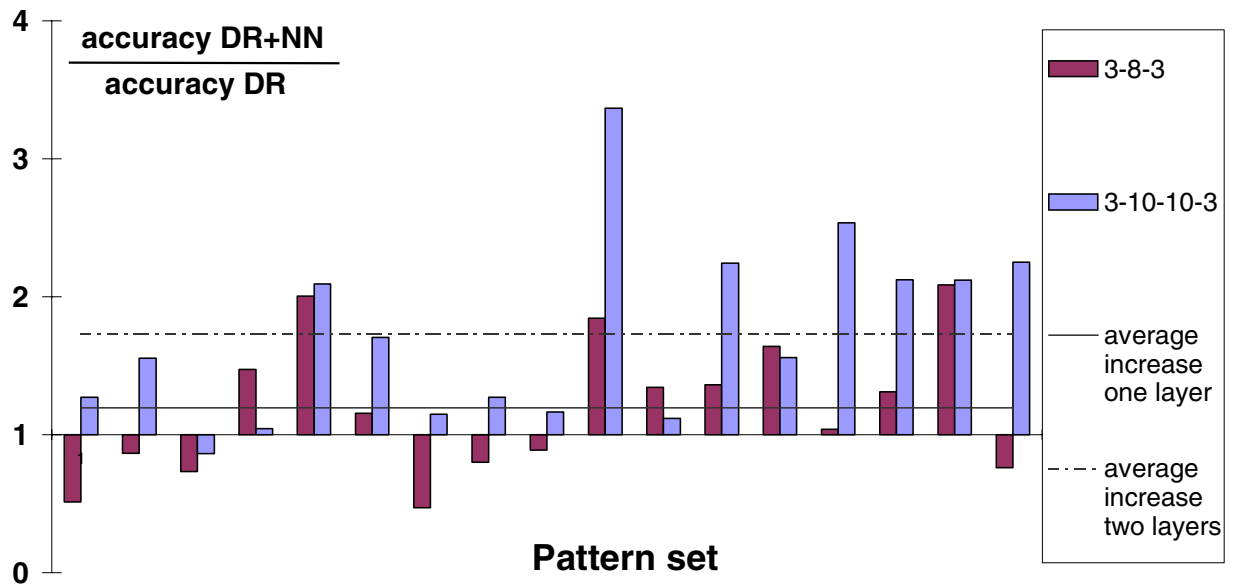


Figure 3-22 Accuracy enhancement for the 3-8-3 and the 3-10-10-3 topologies

Figure 3-22 demonstrates that training data characteristics affect the ability of the net to generalize. If too few data are used, the network will not be able to give a reasonable approximation. On the other hand, if too much data is presented, the network may model “too closely” these data (over conditioning) and thus not be able to generalize to other data.

Figure 3-23 is a plot of the ratio of improvements of accuracy in Figure 3-22 for each unseen pattern. Values above “1” indicate cases where the two layer network performed better than the one hidden layer network. The two hidden layer network is more advantageous than the one hidden layer network.

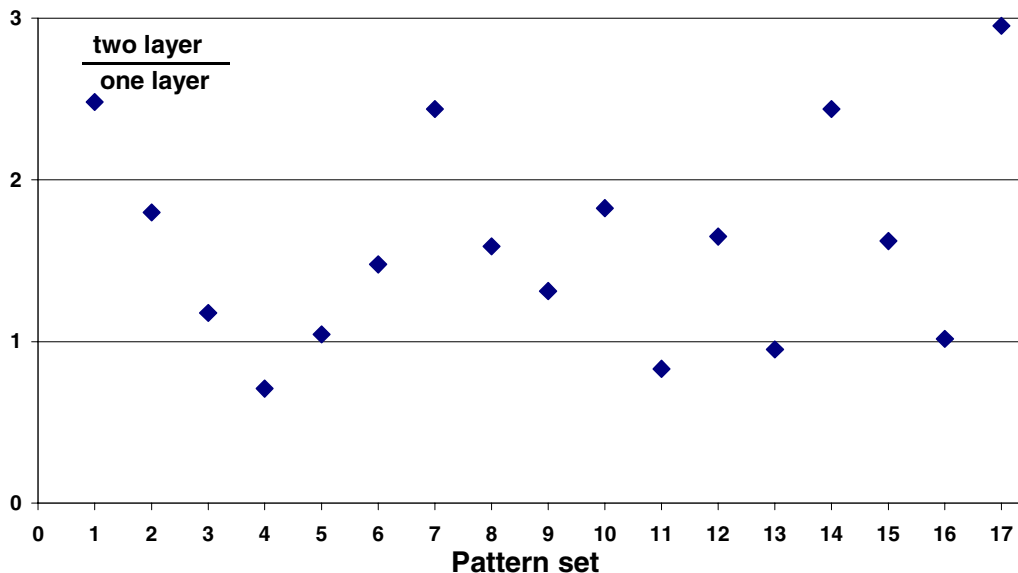


Figure 3-23 Comparison of the two layer/one layer configuration

3.4.4.3 Using Jenkin's hypercube for the selection of training data

Modeling relationships beyond the scope of the training patterns is difficult for statistical methods such as neural networks. Therefore, input-output training patterns should contain data that correspond to even distributions between the borders of spaces of possible values (Section 2.4.1).

Although its main focus is to indicate the best distribution of training patterns, the model can be used to reduce the number of pattern sets needed for training. The focus of this test was to determine the potential for further reductions in the number of training patterns needed. Therefore, two different testing and training pattern sets have been created: one by considering the hypercube concept (pattern set "A"), the other by randomly choosing patterns (pattern set "B"). The minimal number of 27 patterns has been used in both cases. The 3-8-3 and 3-10-10-3 network configurations have been trained with pattern set "A" as well as with pattern set "B". All networks have been tested using the same unseen patterns as in Section 3.4.4.1. Accuracy decreased to unacceptable values in both cases.

Unfortunately, the measurement data that is available in this study does not coincide with the optimal distribution described by Jenkin's hypercube. This is understandable since results are determined by specific loading configurations and the physical principles of structural behavior.

3.5 Substituting the dynamic relaxation method completely with artificial neural networks

The present model uses the dynamic relaxation method in combination with a neural network to calculate the nodal displacements of the tensegrity structure. It could be argued that one neural network might be able to replace the dynamic relaxation method completely. This would save much computational time. As a first step, it was checked whether the dynamic relaxation method could be replaced by a neural network. This has been tested using data taken from analysis results.

When all 33 nodes of the three-module structure are modeled, an extremely complex network results. Complexity increases exponentially with the addition of modules.

Test data were generated analytically for a network where the dead load is constant and the structure is loaded vertically at one point. Starting with three nodes in the input layer and one node in the output layer, the network has been trained for 60,000 cycles. A 3-8-8-1 configuration has been used. Although a sufficiently low training error was attained, the test patterns are further away from the desired output (Figure 3-24).

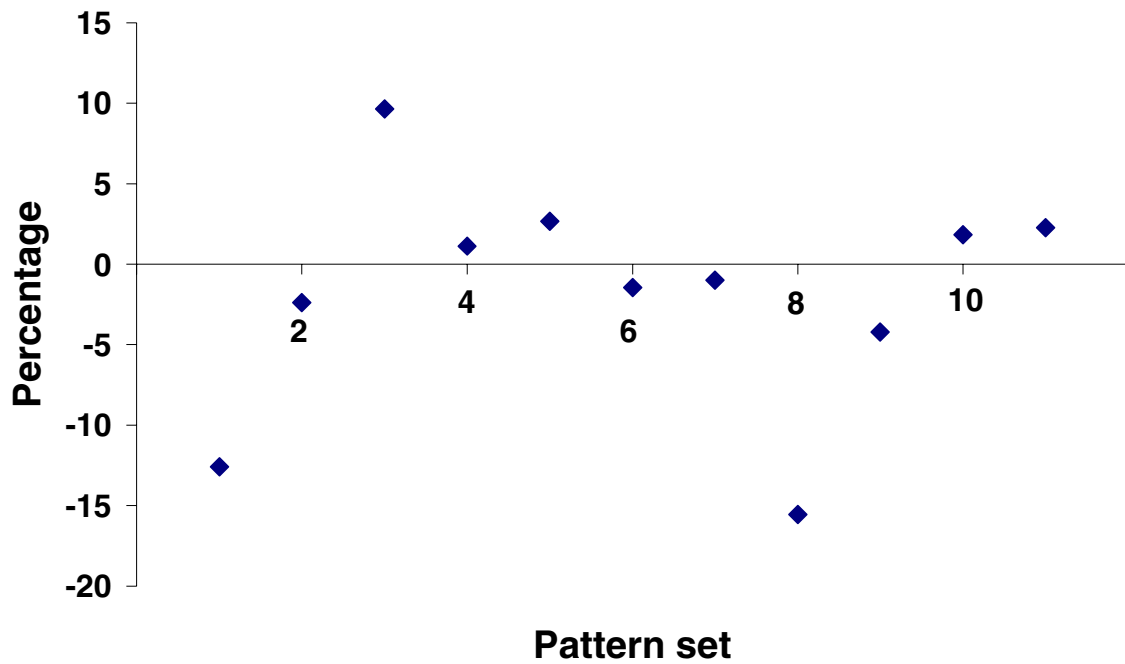


Figure 3-24 Percentage deviations for the test patterns (3-8-8-1 network, sigmoid activation function)

Until now, the sigmoid function has been used as activation function with normalized values between 0 and 1. The tan hyperbolic function has been chosen to enlarge the bandwidth and normalization between -1 and $+1$ and therefore allowing a more precise normalization. More training patterns than used for the previous test have been generated. As Figure 3-25 shows, the deviations increased dramatically. Further tests have been performed using another method of normalizing input data as well as with other network topologies (3-8-8-3 and 3-12-12-3). None revealed satisfactory results.

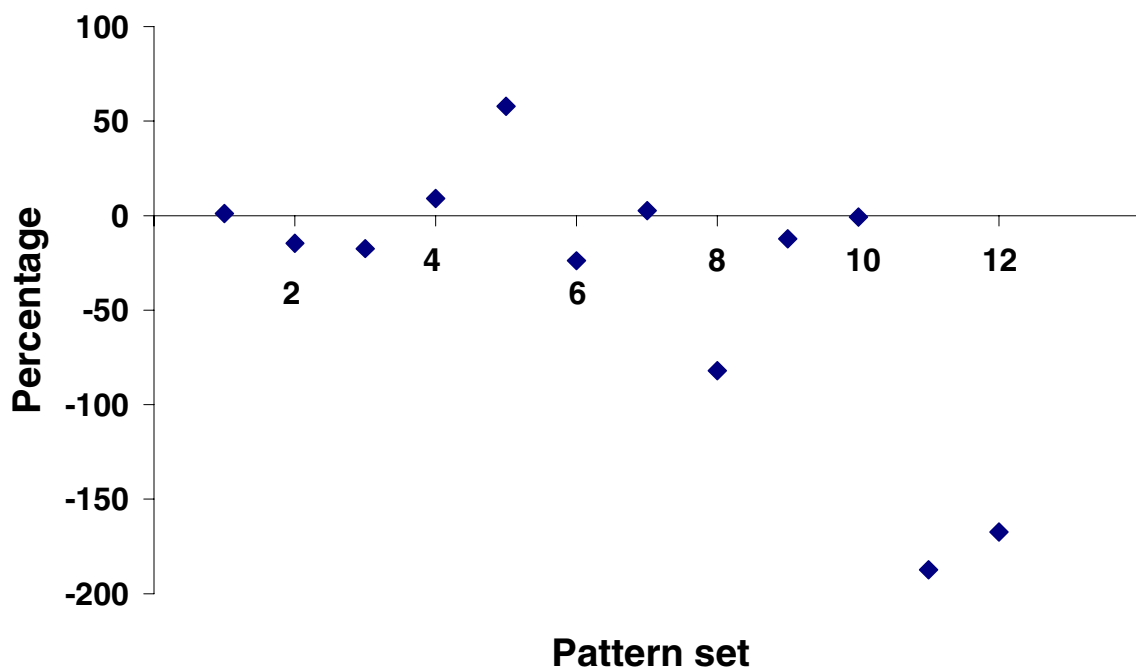


Figure 3-25 Percentage deviations for the test patterns (3-8-8-1 network, tanh activation function)

3.6 Testing the hybrid approach on the five-module structure

The three-module structure has been extended by two modules. With this extension, important details of modular design changed. Rings as shown in Figure 3-3b tie now the arc connector together. The central node has been changed from Stage 1 to Stage 3 design (Figure 3-4). Electrical jacks replaced manual telescopic bars (Figure 3-6). This changes structural behavior modeled by dynamic relaxation and, thus, influences the neural network used for error correction.

3.6.1 Obtaining new training data

Positions of the ten motorized telescopic bars as well as node-numbers of loaded and measured nodes can be found in Figure 3-26.

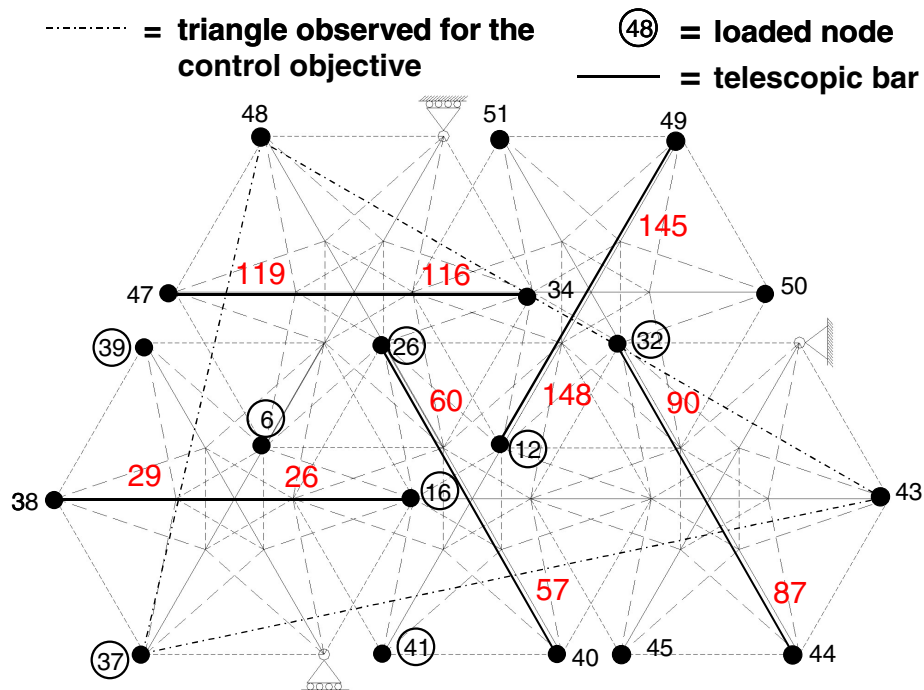


Figure 3-26 Five module structure with loaded nodes and controlled struts

Measurements have been made with the following objectives:

- To calibrate the numerical model employed
- To obtain data for neural network re-training

Loads used can be found in Table 3-6 (one node) and Table 3-7 (two nodes).

Node loaded	Vertical loads applied [N]
6	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211
12	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211
16	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211
26	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211
32	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211
37	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211
39	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211
41	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211
48	152, 270, 387, 505, 623, 741, 858, 976, 1094, 1211

Table 3-6 Single node loading

Nodes loaded	Vertical load applied [N]
37 & 48	157, 196, 313, 430, 547, 664, 781, 898, 1015
48 & 45	157, 196, 313, 430, 547, 664, 781, 898, 1015
37 & 45	157, 196, 313, 430, 547, 664, 781, 898, 1015
41 & 50	196, 313, 430, 547, 664, 781, 898, 1015, 1249
16 & 34	157, 196, 313, 430, 547, 664, 781, 898, 1015
39 & 48	157, 196, 313, 430, 547, 664, 781, 898, 1015
37 & 50	196, 313, 430, 547, 664, 781, 898, 1015, 1249

Table 3-7 Loading at two nodes simultaneously

Loading has been applied as demonstrated in Figure 3-27. Load steps were realized by adding a lead bar weighing 117N for each step to the basic weight. For simultaneous loading of two nodes, the load platform carrying the lead bars has been connected to a traverse that distributes load evenly to two nodes. To get every second loading close to the steps chosen for single loading, small weights have been added to the basic setup. The distance between nodes 37 and 50 demanded for a longer traverse than the one normally used. Decreasing the amount of small weights added has compensated differences in weight.

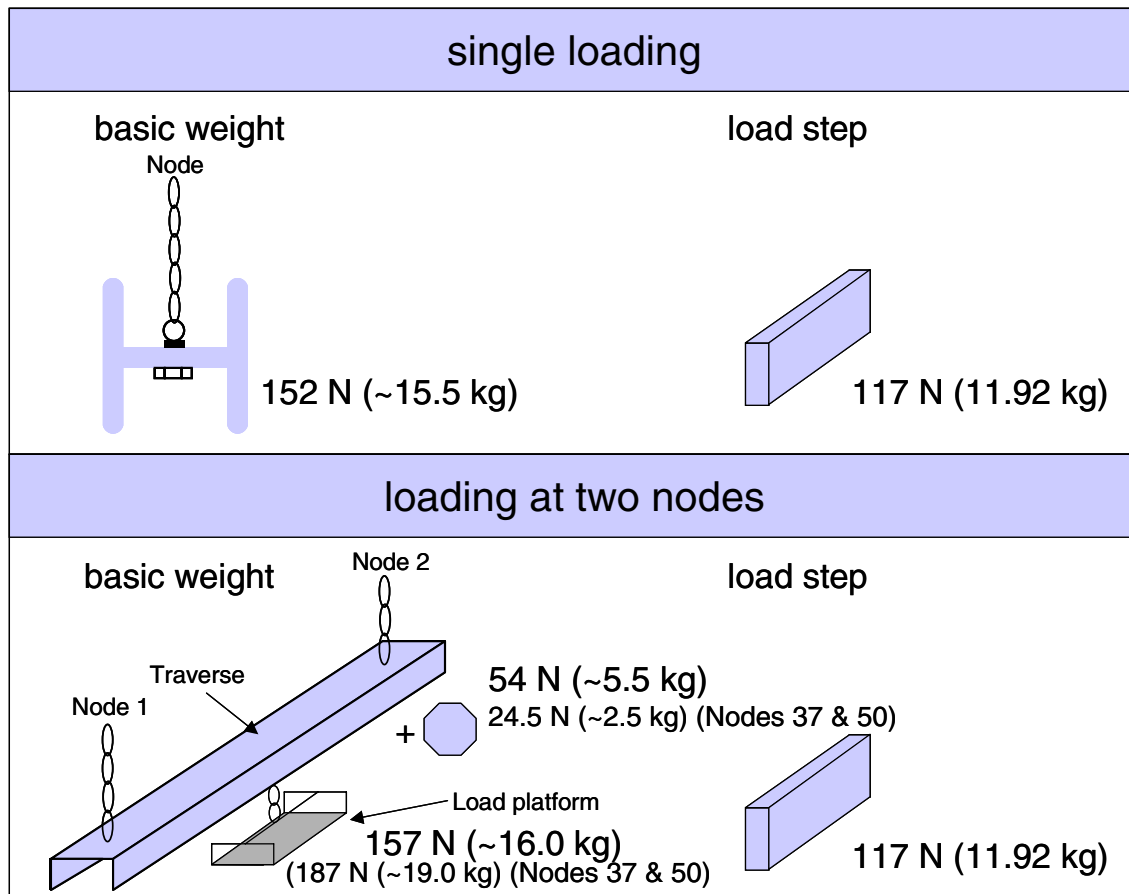


Figure 3-27 Assembly of the loading

Although deflections calculated by dynamic relaxation still described structural behavior correctly, distance observed to measured values were higher than those observed for the three-module structure. As a first measure, elasticity modules have been adjusted using measured results.

3.6.2 Training of a new neural network

The initially trained network for accuracy enhancement cannot be re-used, since

- Nodes to be corrected have changed according to the new control objective presented in Chapter 5.
- Material parameter changed significantly (adding of motors, upgrade of central node, etc.)

Nevertheless, the nature of the network task remains the same: to correct for inaccuracies of the dynamic relaxation method that cannot further be calibrated by adjusting model parameters. This led to the re-use of the 3-10-10-3 network topology that performed best with the preceding structure. Load cases for training have been introduced in Table 3-6 and Table 3-7, a total number of 160 pattern sets has been subdivided into

- 113 training
- 31 testing and
- 16 unseen patterns

As before, data for the three input nodes has been generated using dynamic relaxation. Data for the output nodes were measured on the structure. Measurements have been taken three times each, the average used to compose output patterns. The three nodes network represents the z-deflections of structural nodes 37, 43, 48.

Results obtained are encouraging: in contrast to the network trained for the three module structure, where the average increase in accuracy has been ~ 1.8 , the new average observed is ~ 3.0 .

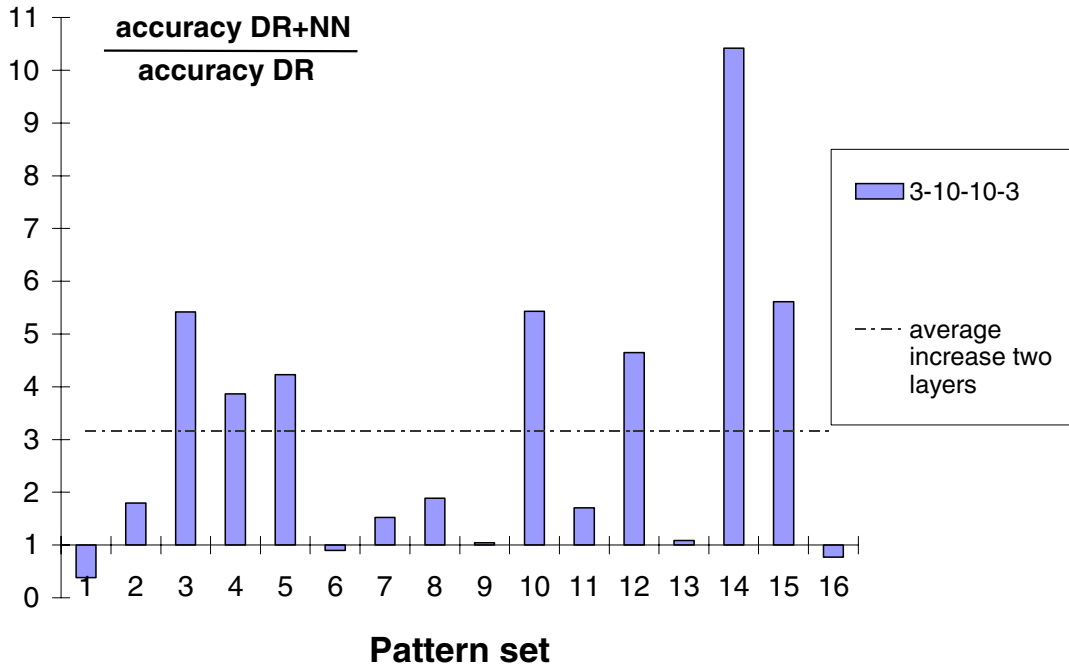


Figure 3-28 Improvement of accuracy

Two patterns with decreases in accuracy were found (1 and 16). This is due to the fact that for pattern 1, accuracy of dynamic relaxation is already excellent and small changes result in accuracy decreases. In the case of unseen pattern 16, values to be corrected were near zero. Correcting values which differed less than 5% from the measured ones in other tests also reduced accuracy. This behavior might justify ignoring the correction in such cases.

3.6.3 Testing robustness of the network

Neural networks are reported to be robust against noise. For testing purposes, a neural network with artificially perturbed input patterns has been generated. The perturbation has been introduced by changing the modulus of elasticity during simulation. This resulted in changed input patterns. Results of this network obtained with perturbed unseen patterns are presented in Figure 3-29.

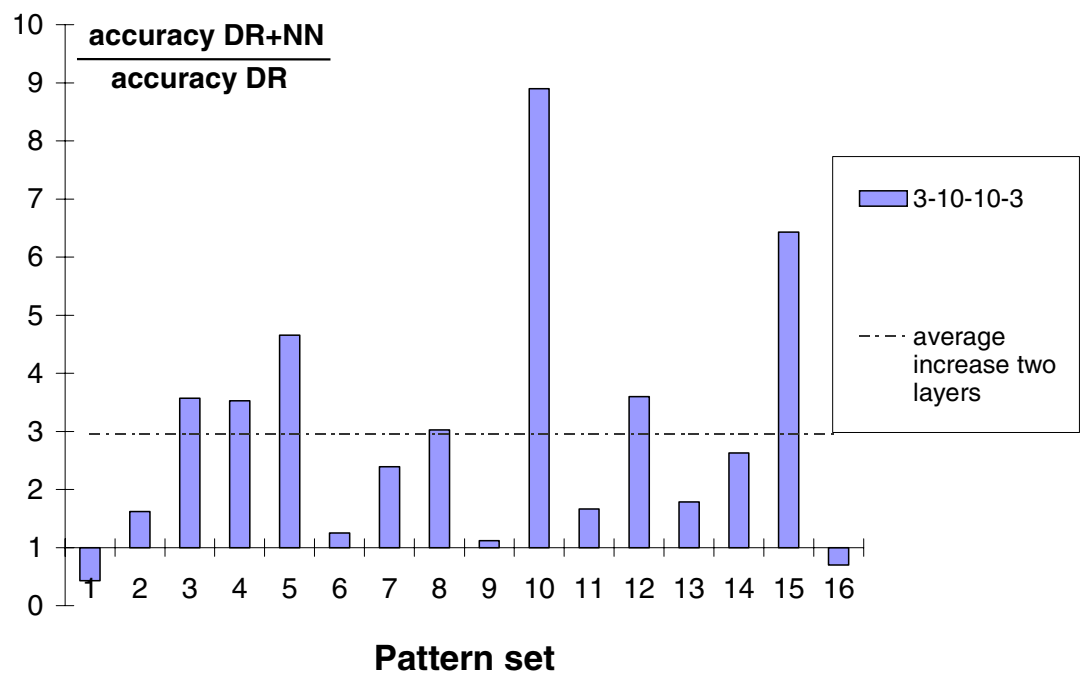


Figure 3-29 Increases in accuracy of the perturbed input patterns

The network has then been evaluated with corrected unseen patterns and without re-training (Figure 3-30). This evaluation showed proof of stability: results show almost the same level of precision as before.

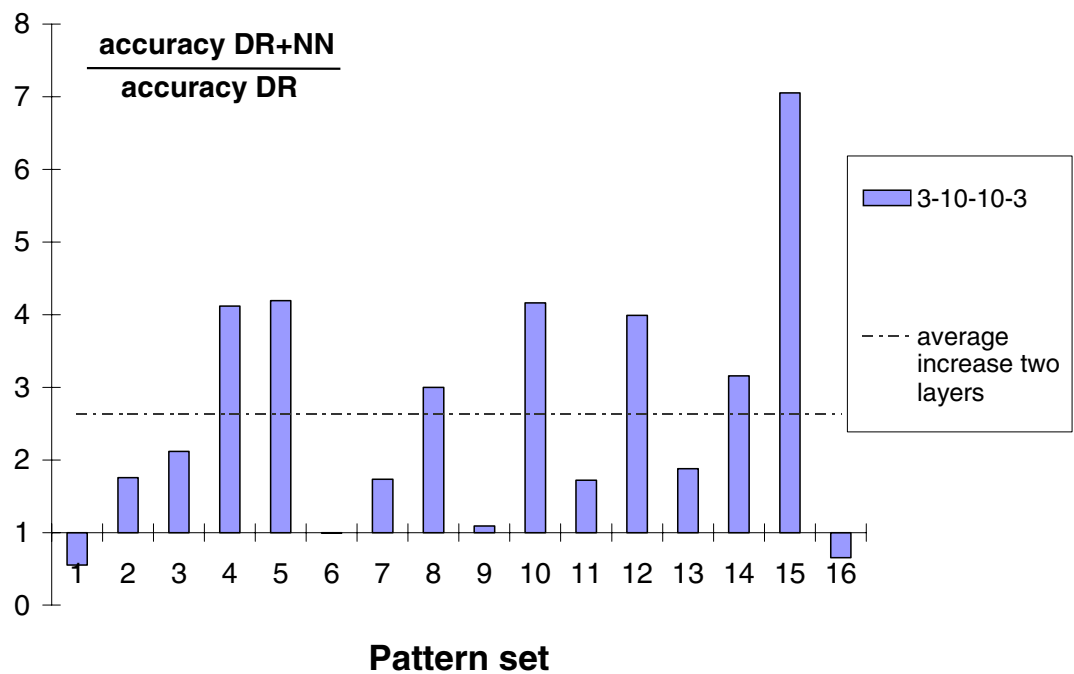


Figure 3-30 Improvement of accuracy, obtained with corrected unseen patterns,

3.7 Discussion of results

Artificial neural nets provide a useful complement to simulation using the dynamic relaxation method. Even when used with a relatively sparse number of training data, they lead to increased model precision while maintaining explicit structural knowledge. Such a combination is clearly better than using neural nets alone.

In exceptional cases where the neural net is not able to increase accuracy, the decrease in accuracy is not expected to affect the reliability of the model for iterative active control.

Although the concept of choosing the training data according to a hypercube has been used successfully with other applications, its implementation is complicated by the characteristics of the space of the measurement data. It might, nevertheless, be a useful filtering tool when the structure is controlled actively, since large numbers of measurements will be available.

Several network configurations using various activation functions and methods for normalization of input values have been studied. No justification for complete replacement of the dynamic relaxation method with a neural network could be found.

Application of the hybrid approach to the five-module structure with changed structural characteristics gave further confidence in the hybrid approach. Testing robustness against noise by using artificially changed input patterns showed stability.

Increases in accuracy justified employing neural networks. Tests underlined network stability in case of small errors.

4 Stochastic search algorithms for active structural control

The concept of active structural control has been introduced in chapter 2. The schematic (Figure 4-1) is repeated here, since it will serve as reference for proposed extensions. Building on work of Shea and Smith (1998), three stochastic search techniques are tested and compared for active structural control. After investigating the complexity of the control task, stochastic search algorithms are proposed for an intelligent sampling of the solution space. Possible solutions are tested against the behavioral model of the structure provided by the dynamic relaxation method. More precisely, a structural analysis is performed during each iterative step of the search.

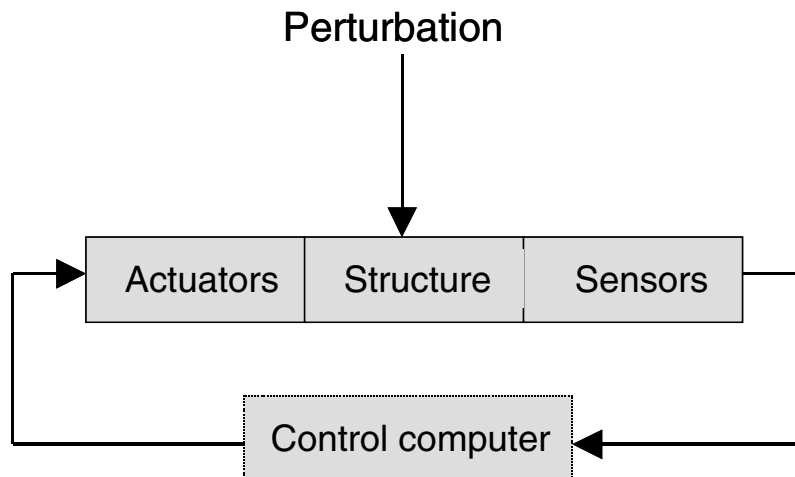


Figure 4-1 Schematic of active structural control

This chapter discusses the following issues:

- Complexity of control tasks
- Can stochastic search be used to find good control commands?
- Comparison of different methods tested

4.1 Complexity of the control task

Active structural members of the tensegrity structure are telescopic struts. They can be extended or contracted, controlling the self-stress and, thereby, the shape of the structure. A general control objective of the tensegrity structure assembly might be formulated as follows:

Find a set of strut positions that minimizes the cost of the objective function under a given loading of the structure.

In the following, a set of strut positions minimizing the objective function will be called a set of good control commands.

Since the behavioral model is geometrically non-linear, such a set cannot be determined directly by matrix inversion. One strategy to find a good set is to generate and test all possible solutions. This has been used in (Kobori et al. 1993). However, with this example, complexity was low. The number of possible solutions in this case was 3. In the case of the active tensegrity structure, the number and discrete positions of the telescopic strut define the space of possible solutions.

Assuming that a telescopic strut can move ± 21 mm from its initial position, further that precision of application of control movements is 0.25 mm, the total number of possible strut positions evaluates to

$$\frac{21}{0.25} \cdot 2 = 168 \quad (4-1)$$

Assuming a tensegrity structure consisting of three-module, each with 6 telescopic struts as a basis to calculate all possible combinations of bar positions *num* results in the following equation:

$$num = P^n \text{ with} \quad (4-2)$$

n: number of telescopic bars (3 modules \times 6 telescopic bars = 18)

P: number of possible bar positions ($21/0.25 \times 2 = 168$)

As an estimate, each calculation of the objective function takes 0.3 seconds. The time needed to test all possible solutions evaluates to 1.08×10^{32} years.

Increasing processor speed and the use of massive parallel systems might decrease this time significantly. Table 4-1 shows the application of Moore's law, which states that processor speed doubles every 18 months, to the same control task.

Number of modules	Time needed to evaluate all possible solutions		
	Today [years]	In 5 years 10 times faster [years]	In 20 years 10000 times faster [years]
3	1.08E+32	1.08E+31	1.08E+28
7	2.76E+85	2.76E+84	2.76E+81
15	1.80E+192	1.80E+191	1.80E+188
20	1.04E+259	1.04E+258	1.04E+255

Table 4-1 Moore's law applied to the control problem

Note, that complexity increases by adding more modules. Increases in processor speed cannot counteract a task that is

- *Exponential* with respect to the number of struts
- *Factorial*, when the sequence of control commands is important

It is obvious that a more sophisticated approach, sampling possible bar movements is needed. In contrast to “brute force” approaches, which generate and test every possible solution to find a global minimum, stochastic methods sample the solution space using special strategies. Although they provide strategies to avoid local minima, solutions found might not represent the global minimum. Nevertheless, near optimal solutions can already successfully be applied for control applications. Techniques tested are:

- Simulated annealing (SA)
- Probabilistic global search Lausanne (PGSL)
- Genetic algorithms (GA).

4.2 Control objective and constraints

The working objective of a constant slope has been inspired by the possible application of the tensegrity structure as a roof or an antenna structure. The fabric attached to the upper layer should stay on a constant slope to avoid ponding (Figure 4-2).

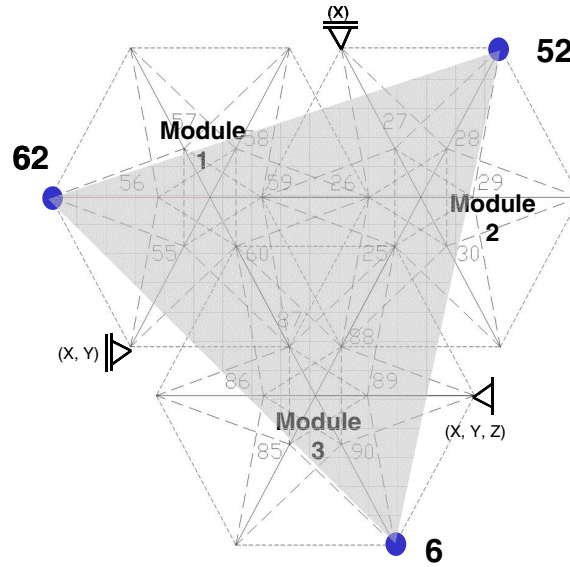


Figure 4-2 Control objective constant slope, nodes used for slope control

The objective function used compares nodal positions of three nodes on the upper layer in the z-direction. A near ‘0’ value indicates good solutions.

(4-3)

$$\text{cost} = \sqrt{\frac{(Node_{6,z} - Node_{52,z})^2 + (Node_{6,z} - Node_{62,z})^2 + (Node_{52,z} - Node_{62,z})^2}{3}} (mm)$$

The constant slope objective does not necessarily require that the upper layer of the structure returns to its initial position after loading. Possible solutions can also be found well above or well below this position (Figure 4-3).

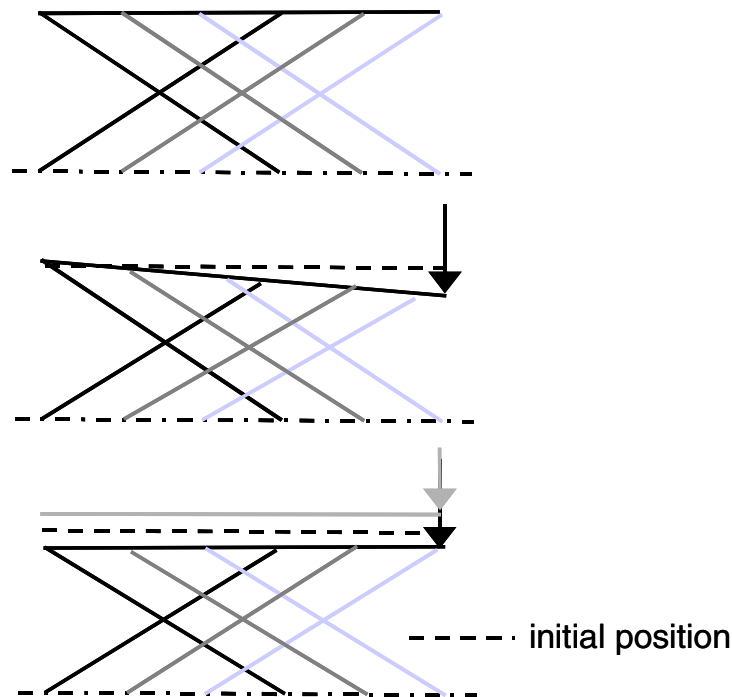


Figure 4-3 Multiple valid solutions of the control objective slope control

The constraints used were

- Maximum bar stress [28.5 Mpa
- Maximum cable stress [901 Mpa (70% of the maximum cable stress allowed by the manufacturer)
- Maximum buckling force [20 kN.

4.3 Control system architecture

Section 2.2.2 introduced the concept of active structural control where building displacements are constantly measured and actuator movements are calculated by the control computer. The algorithm for tensegrity slope control employs stochastic search. As a first technique, simulated annealing has been used. Validating the objective function is computationally costly, since each time a full structural analysis has to be performed.

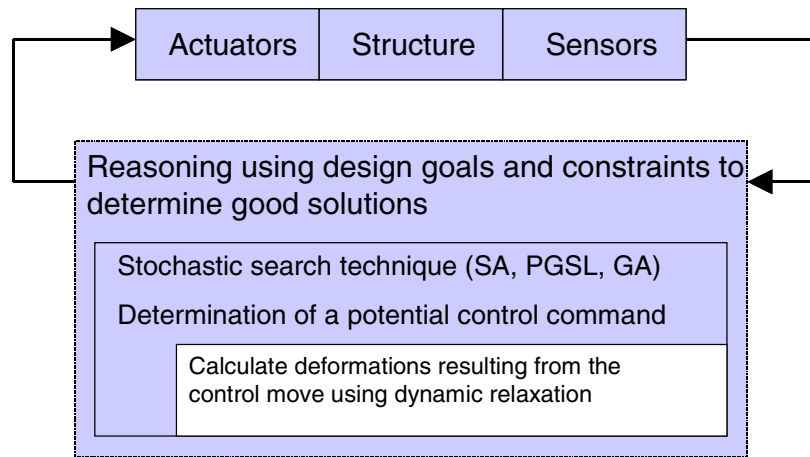
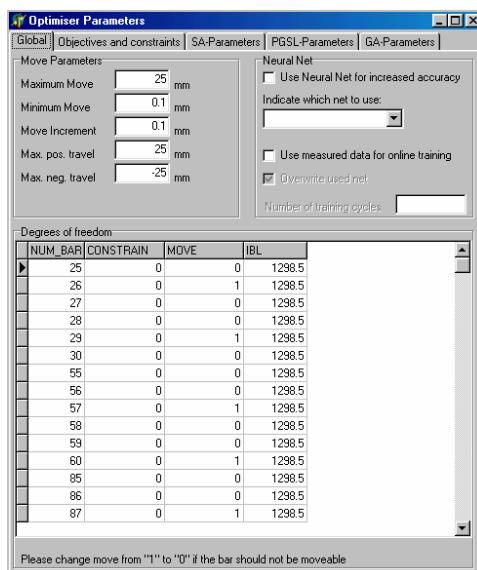
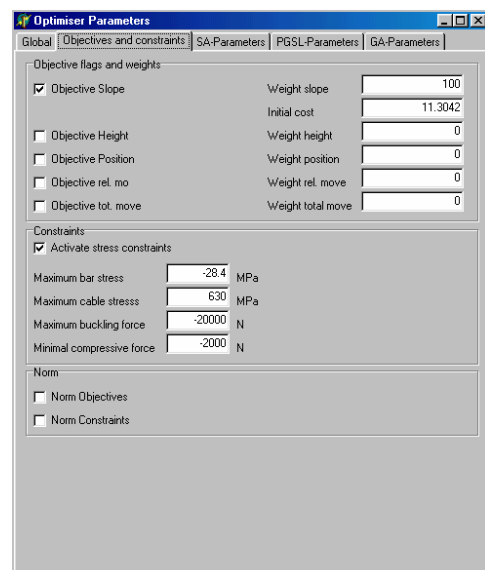


Figure 4-4 Extended basic scheme of structural control

All stochastic search techniques are tested in the same framework provided by the Tensegrity structure analysis and control software (TSACS). The application used (Optimiser) provides a generic interface for all search techniques employed. In the “Global settings” menu, maximum and minimum bar strokes, the bars that are telescopic, move increments, etc. can be defined. The “Objectives and constraints” menu proposes different applicable control objectives, weighting factors related to those and constraints to be checked during optimization.



a) Global settings



b) Objectives and constraints

Figure 4-5 Screen-shots of general settings

The main window of Optimiser (Figure 4-6) asks for nodes to be used with the control objective as well as the search algorithm. All additional data needed are taken from the database.

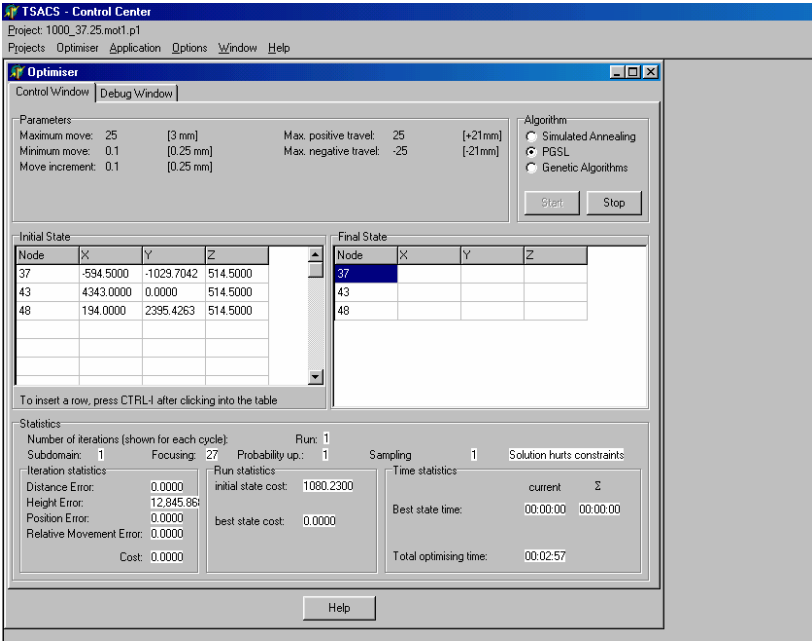


Figure 4-6 Screen shot “Optimiser”

4.4 Test description

4.4.1 Test configuration

Good sets of control commands found by simulated annealing have been tested on the real structure (Fest 2002) with the constant slope objective discussed in Section 4.2. Figure 4-7 shows the result of applying such a control command to the three-module structure with five telescopic bars per module. The y-axis represents the value of the constant-slope objective, once calculated with the dynamic relaxation method (Analytical model) and once measured on the structure (Tests). Bar strokes are applied manually, one bar at a time. The initial slope difference has been almost entirely compensated when the last telescopic bar is adjusted. Similar observations have been made for seven load cases tested

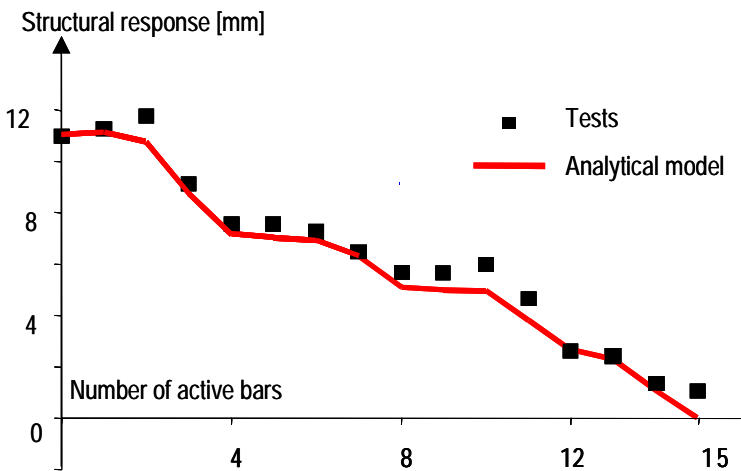


Figure 4-7 Structural response as function of modified bars (Fest 2002)

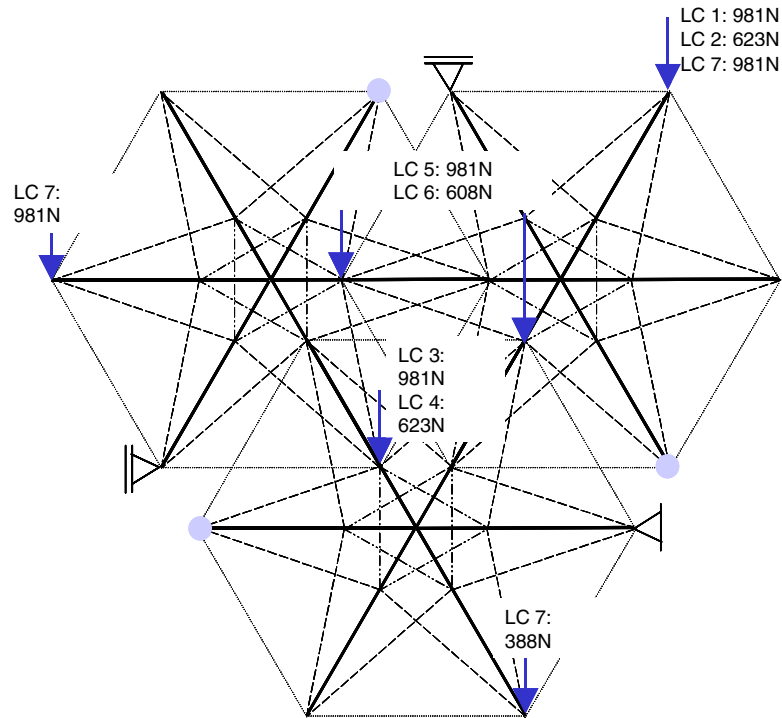


Figure 4-8 The seven basic load cases, nodes controlled by the objective function are shaded in gray

Tests have been conducted for a two and five telescopic bar per module configuration. The two telescopic struts varied for each load case tested in order to test different arrangements.

Configuration 1	Configuration 2	Configuration 3
	Configuration	Loadcase
	1	1
	2	2, 4
	3	3, 5
	4	6
	5	7
Configuration 4		Configuration 5

Figure 4-9 Configuration for two telescopic struts per module, thick lines represent telescopic struts
In the five strut configuration; one strut per module has been fixed to avoid instabilities.

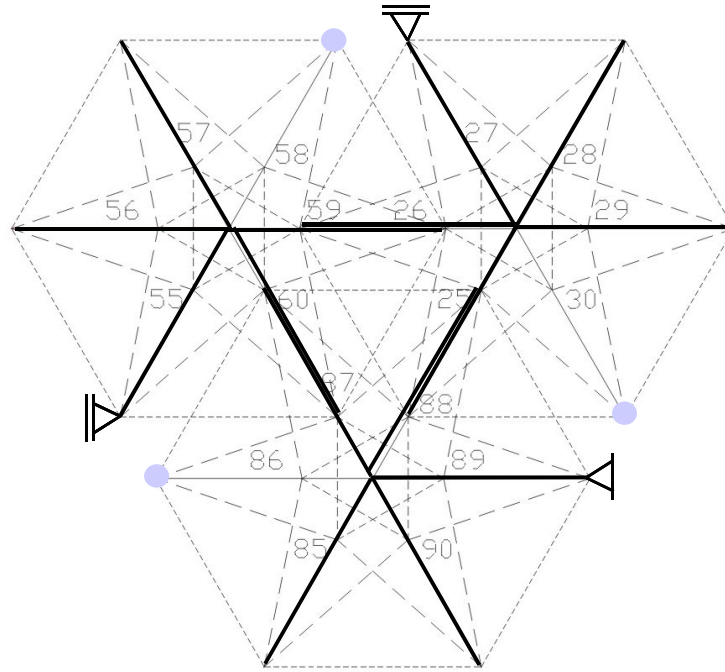


Figure 4-10 Configuration for five telescopic struts per module, thick lines represent telescopic struts

These test configurations were re-used, first to calibrate the simulated annealing algorithm which has been integrated into the TSACS environment, then to test two additional techniques (PGSL and Genetic Algorithms).

Three test series are studied:

Series 1: Constrained scenario

Two bars per module were assumed to be telescopic and could be moved by $\pm 3\text{mm}$. Seven different load cases are optimized. (Tested algorithms: simulated annealing, PGSL and GA's)

Series 2: Search progression

Five bars per module were assumed to be telescopic and could be moved by $\pm 3\text{mm}$. Seven different load cases are optimized. (Tested algorithms: SA, PGSL, GA's)

Series 3: Effect of the number of search runs

Five bars per module were assumed to be telescopic and could be moved by $\pm 3\text{mm}$. Two different load cases are optimized. (Tested algorithms: simulated annealing and PGSL)

4.5 Comparison of stochastic search techniques

4.5.1 Simulated annealing

Simulated annealing has been ported from the implementation already used for the preliminary tests described in (Fest 2002).

Schedule parameters have been set according to explanations given in Section 2.3.1.2. The number of iterations has been set to 60 for the two-strut configuration and 150 for the five-strut configuration. Each iteration (move) consisted of 100 candidate solutions. Six sub-

ranges for variable changes were used to define the move set. As simulated annealing works best for incremental small changes of solutions, the first four ranges cover 50% of the total range, 3 mm, while the remaining two are set at 70% and 100% of the maximum range.

A guideline given in (Swartz and Sechen 1990) for the number of moves per iteration is

$$10 \times n_{\text{variables}}^{\frac{4}{3}} \quad (4-4)$$

with $n_{\text{variables}}$ = number of variables. While this was used as a starting point for setting the number of moves it was found that good convergence was achieved in far fewer moves for this particular task.

A further 40 “freeze” iterations are performed at the end of the process where only better solutions are accepted. The initial temperature was set to 200, based on the numeric range of the cost function, and updated every ten moves.

Additional parameters were then set to allow the process to stop if absolute and relative convergence criteria were met. In the results presented, all processes converged within the first two iterations of the “freeze” process. All search parameters were held constant across all load cases.

4.5.2 Probabilistic global search Lausanne (PGSL)

Parameters for PGSL are adjusted following empirical guidelines provided in Section 2.3.2.2. Table 4-2 presents the parameter sets tested. Sets 2-4 used the empirical procedure for choosing parameters whereas set 1 tested arbitrary chosen ones. All sets have been evaluated for two load cases.

Parameter	Set			
	1	2	3	4
Number of sampling cycles (NSC)	5	2	2	2
Number of probability updating cycles (NPUC)	3	1	1	1
Number of focusing cycles (NFC)	150	150	300	150
Number of subdomain cycles (NSDC)	8	10	15	15

Table 4-2 Test cases for parameter adjustment of PGSL

Results are plotted in best-so-far curves. X-axis represents the number of iterations, the y-axis the costs.

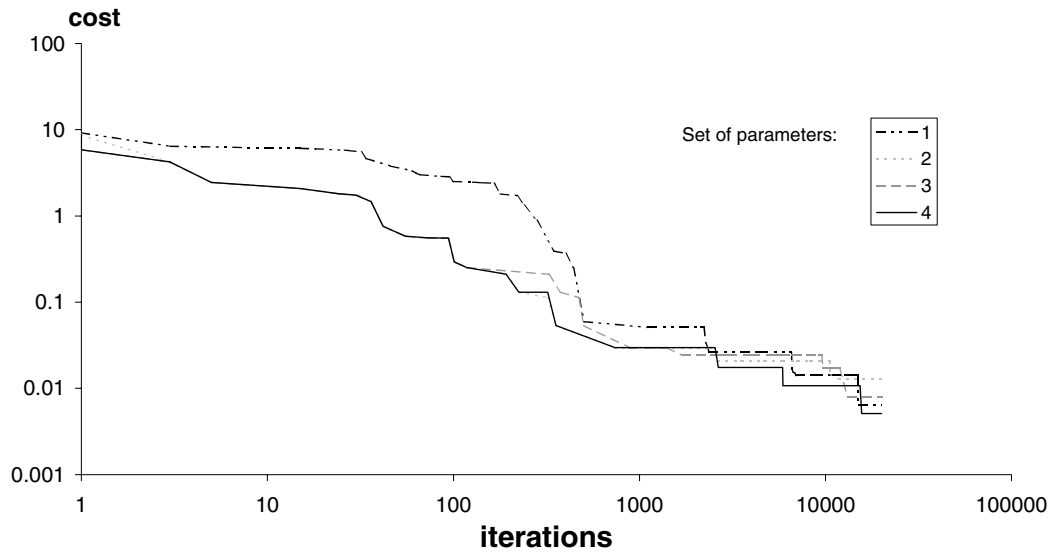


Figure 4-11 Parameter study

Parameter set 4 has been used with the tests. The parameter study reveals that the procedure followed for sets 2, 3 and 4 results in the best solutions. Values of sampling and probability updating cycles stayed constant for parameter sets 2, 3 and 4. Therefore, only the values of focusing cycles (NFC) and subdomain cycles (NSDC) need to be adjusted to fix the total number of evaluations of the objective function. This underlines the ease and simplicity of fixing PGSL parameters.

4.5.3 Genetic algorithms

GA's are often complicated because of the encoding and decoding used for the chromosome. Our application encodes only bar length that is rather simple. Parameters sets have been adjusted to the control task as described in Section 2.3.3.2. Best results were obtained by applying parameters presented in Table 4-3. Test series 1 and test series 2 employed different sets.

	2 bars per module	5 bars per module
Width	30	75
Population size	30	75
Pcross	0.9	0.9
Pmut	0.015	0.0066
Num_gen	60	300
Selection	Tournament selection	Tournament selection
Encoding	Gray scale	Gray scale
Elitism	1 member copied	1 member copied

Table 4-3 Parameters used with genetic algorithms

4.5.4 Results

All results have been obtained on a Pentium PC with a processor speed of 600 Mhz. The PC had no additional network load.

4.5.4.1 Series 1

The time needed to find the best solution is presented in Table 4-4.

Load Case	Minimum Cost [mm] Equation (4-3)	Time [mm:ss]		
		SA	PGSL	GA
1	9.8096	25:14	06:16	07:40
2	3.3600	17:51	04:44	04:51
3	6.1594	15:57	07:44	05:24
4	1.1591	13:28	05:32	06:05
5	6.3735	08:12	06:26	04:31
6	1.4448	13:07	07:37	06:09
7	3.1963	13:06	07:29	05:22

Table 4-4 Time needed to find the best solution for 2 moveable bars per module

Although all algorithms required approximately 3 to 4 hours to complete search, the final solution for the two bar task was found after several minutes as noted in Table 4-4. Minima attained were the same with all stochastic search techniques. Solutions were identical regarding cost and proposed bar strokes. An optimal solution near a cost value of zero could not be obtained in this case since:

- Only two telescopic bars per module were allowed to move
- Movement was limited to $\pm 3\text{mm}$
- Possible bar positions were only in steps of 0.25mm

Upon examining the proposed bar strokes for the final solution it was observed that these tests identified solutions that are on the edge of the solution space since every bar was moved by its maximum stroke, either $+3\text{mm}$ or -3mm . This signifies that the iterations did not provide better solutions after each bar was extended to its limit. Nevertheless, GA and PGSL identified the best solution more rapidly than simulated annealing.

4.5.4.2 Series 2

Load case	Initial cost Equation 4-3	SA		PGSL		GA	
		Final cost	Iterations	Final cost	Iterations	Final cost	Iterations
1-edge	16.9356	1.4101	16,287	1.4101	1,700	1.4101	4,638
2-edge	10.7783	0.0073	23,781	0.0085	32,428	0.0065	15,816
3-top/center	13.2787	0.0106	51,888	0.0054	30,000	0.0029	5,068
4-top/center	8.6432	0.0024	11,779	0.0075	16,316	0.0028	15,696
5-top/center	12.5542	0.0171	21,000	0.0050	21,628	0.0036	15,216
6-top/center	7.8998	0.0071	1,670	0.0050	37,842	0.0008	14,870
7-edge	9.9886	0.0051	45,781	0.0029	37,811	0.0008	6,913

Table 4-5 Costs and iterations for 5 moveable bars per module

Table 4-5 presents the number of iterations needed to attain the best state for each combination of load case and search technique. During each iteration the objective function is evaluated.

The development of the costs during an optimization has been plotted in best-so-far curves as introduced in Section 4.5.2. The curve of the simulated annealing process shows three peaks. As a point-to-point search technique, it is launched three times from the initial conditions in order to allow three different paths to converge to a near optimal solution. For PGSL and genetic algorithms in contrast, this is less advantageous since they are inherently parallel techniques.

Figure 4-12 to Figure 4-18 provide results from seven different load cases that now can be compared with respect to minimum cost and speed of convergence. PGSL and GA show faster convergence in load case 1 (Figure 4-12) than simulated annealing. Nevertheless, all algorithms converge to the same best cost. This leads to the conclusion that this solution is most likely the global minimum.

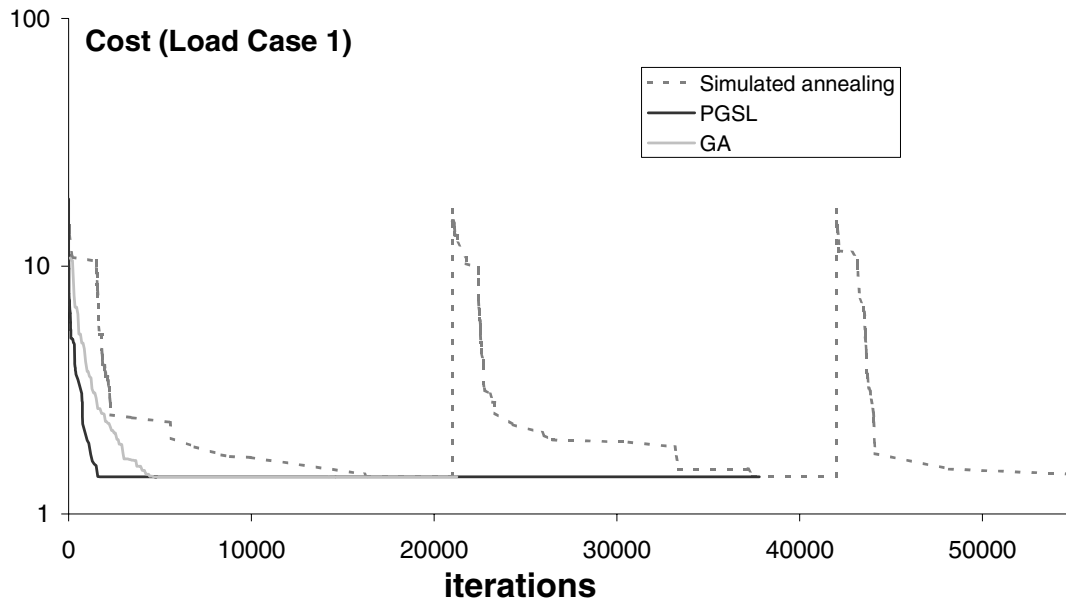


Figure 4-12 Best-so-far curve load case 1

The final results of load case 2 (Figure 4-13) are close to each other. GA's converge to the best result. Convergence behavior is better compared to simulated annealing. Although PGSL is outperformed regarding the end result and number of iterations, in the first 2,500 iterations, PGSL converges faster.

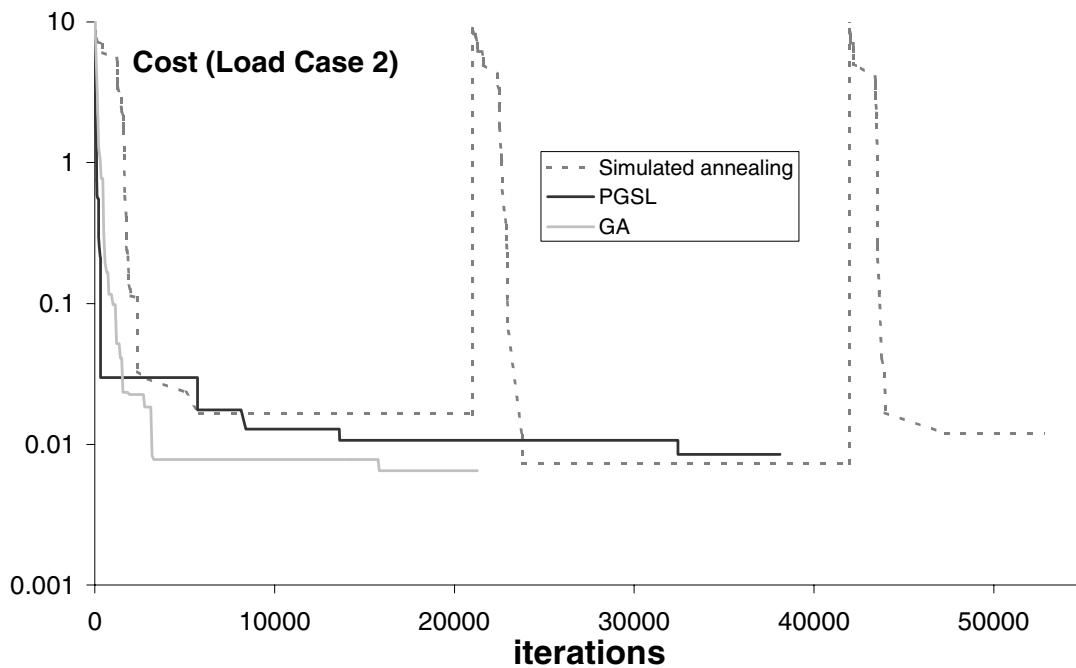


Figure 4-13 Best-so-far curve load case 2

For load case 3 (Figure 4-14), PGSL performs better than simulated annealing in terms of best cost and speed of convergence but is left behind by GA's. Again, there is a short period in the beginning of the search where PGSL converges quicker than GA's.

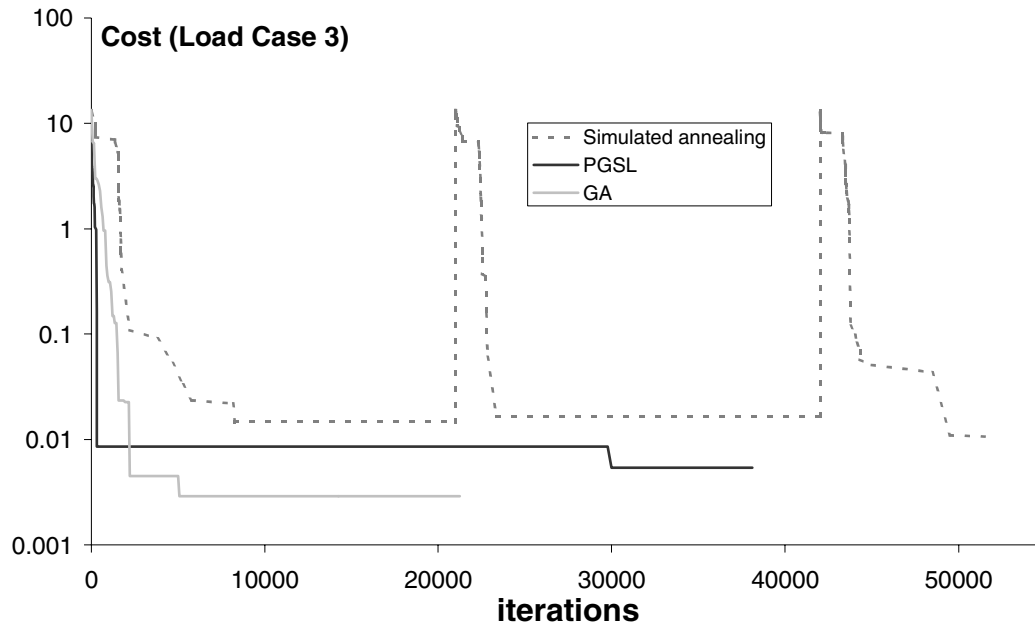


Figure 4-14 Best-so-far curve load case 3

Simulated annealing converges to the best solution in load case 4 (Figure 4-15). Genetic algorithms are close to SA and converge faster.

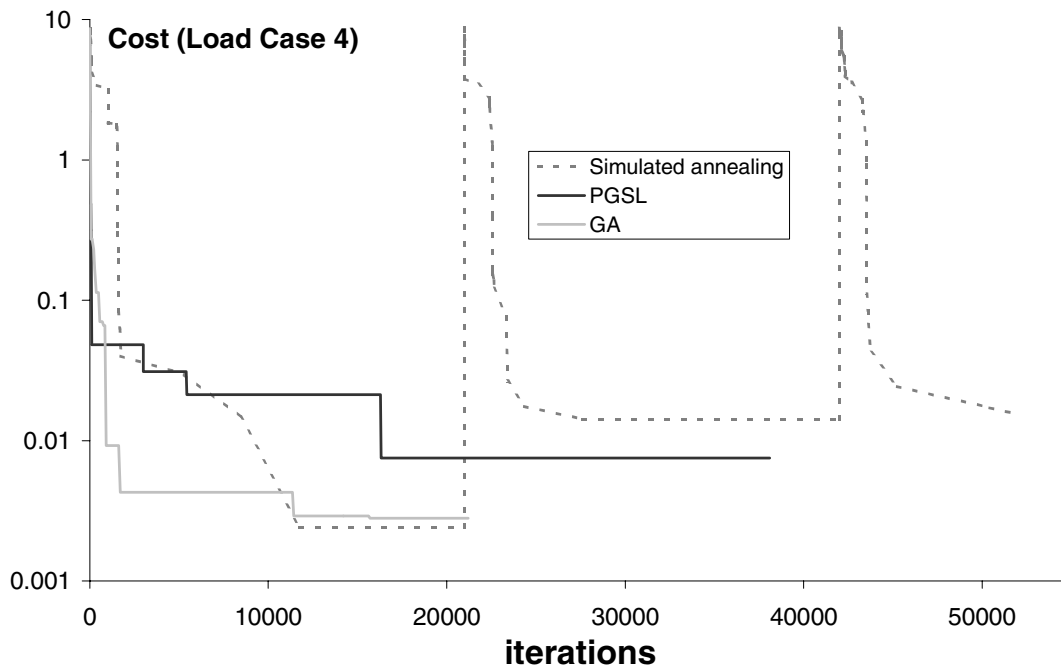


Figure 4-15 Best-so-far curve load case 4

Observations for load case 5 (Figure 4-16) are similar to those made for load case 3. However, a zone between approximately 3000 and 21000 iterations is present, where simulated annealing outperforms PGSL.

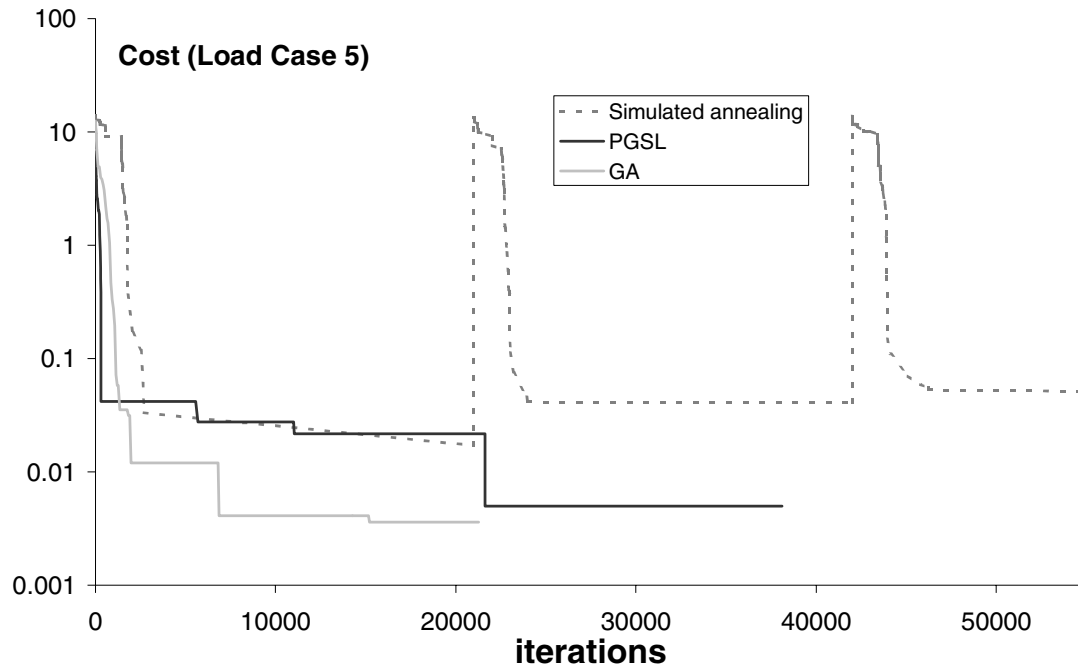


Figure 4-16 Best-so-far curve load case 5

Load cases 6 and 7 (Figure 4-17 and Figure 4-18) may be discussed together since these results are analogous. PGSL converges faster for the first approximately 2000 iterations of the objective function. In the middle of the iteration, simulated annealing provided better solutions than PGSL. Although PGSL found the set of best bar movements, the differences in cost are negligible. Once more, GA's outperformed simulated annealing and PGSL.

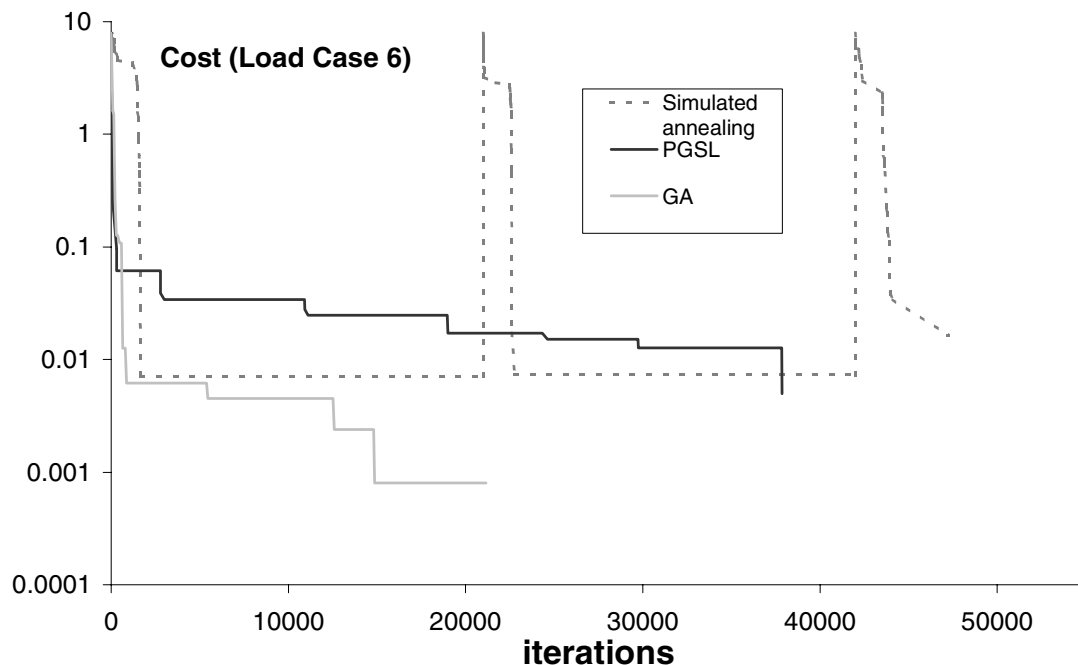


Figure 4-17 Best-so far curve load case 6

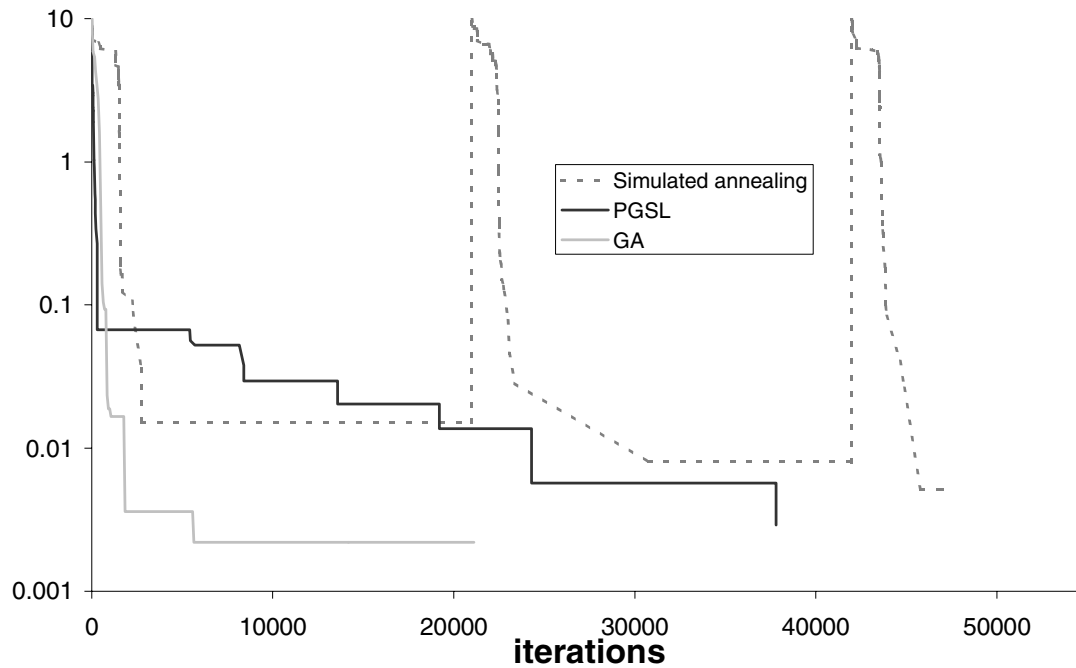


Figure 4-18 Best-so-far curve load case 7

4.5.4.3 Effect of the number of runs

Stochastic search techniques do not necessarily converge to the same solution when started multiple times for the same objective and initial state. Therefore, multiple runs have been executed for two load cases (load case 2 and load case 6) to evaluate the effect of the number of runs.

Figure 4-19 shows the results for 25 runs for load case 2. After two runs, a solution in the region of the best solution has already been found. Both algorithms converge to the same solution after eleven runs.

Load case 6 (Figure 4-20) is a more complex problem. The lowest cost curve for simulated annealing shows a more “staircase-like” behavior. Although after 6 runs, no further changes in the lowest cost of simulated annealing can be observed; PGSL finds a better solution close to the best state of simulated annealing after 15 runs. In all cases, acceptable costs were achieved after two to four runs, considering practical aspects of applying solutions to the structure. This is discussed further in the next section.

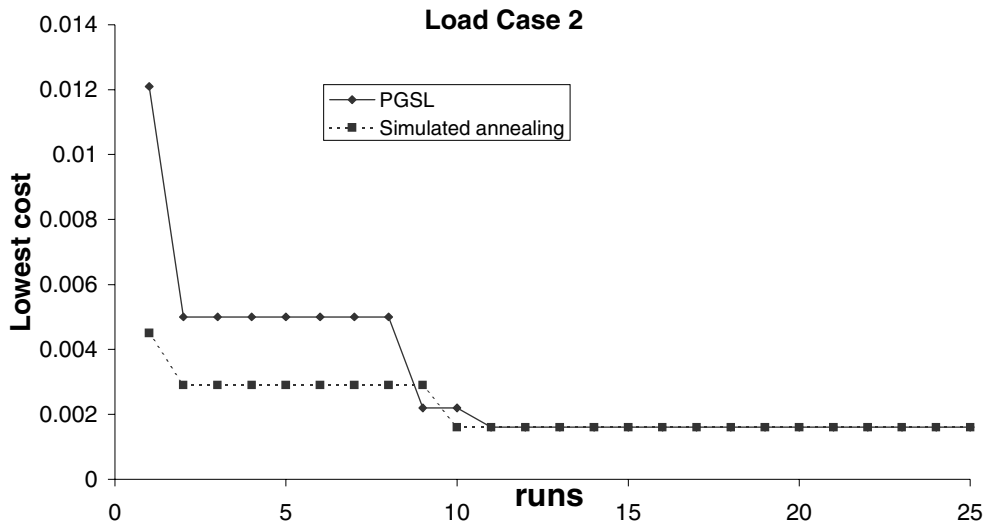


Figure 4-19 Best-so-far curves for multiple runs, load case 2

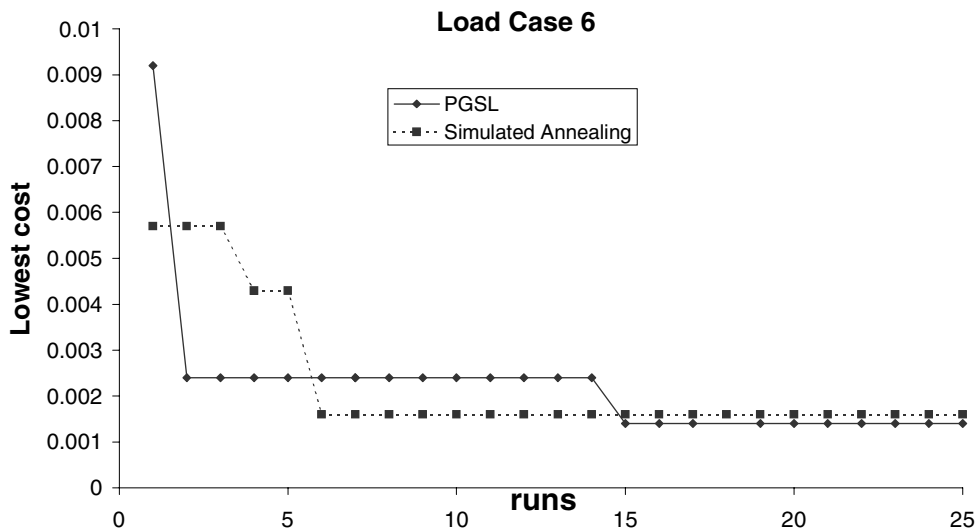


Figure 4-20 Best-so-far curves for multiple runs, load case 6

4.6 Discussion of results

The first two test series showed the ability of all stochastic search methods to find good control solutions. As shown in 4.5.4.2, there was little evidence that one method provides more accurate results than the other. No best algorithm for all test load cases can be identified. These results thus support the “no-free-lunch” theorem (Wolpert and Mcready 1997). PGSL converges faster in the approximately first 1000 iterations of the objective function. During that period, simulated annealing accepts worse solutions to avoid local minima. This behavior leads to a better end result in some cases. Nevertheless, from a

practical viewpoint, all algorithms provide good results. GA's provided a more stable behavior. Convergence was good in all cases and a best cost was always found.

The success of this search does not necessarily require solutions that are near to 0. As was observed in Section 4.5.4.1, it may not be possible to counteract completely all deflections within the constraints of this task. Furthermore, the usual inaccuracies between behavioral models and real behavior often do not justify the computational cost of a theoretically better solution. Such tradeoffs help determine the most appropriate values required for accuracy.

Evidence of the stochastic nature of PGSL and SA has been given with test series 3, presented in section 4.5.4.3. Although costs were similar, command characteristics were different for almost each run. The time necessary to evaluate multiple runs for determination of good control moves inhibits the practical use of search methods even for quasi-static control. Solutions with similar values of the objective function result in significantly different bar movements. This has an impact on their applicability, since current actuator positions might lead to different choices.

PGSL and GA have advantages in terms of the number of parameters to be fixed before each optimization process. Simple guidelines lead to rapid parameter adaptation for other applications.

As a step-wise method, simulated annealing allows movements to control solutions that violate constraints, which are currently rejected. SA may include soft constraints in order to find an optimum by stepping through a region of invalid solutions. Since PGSL and GA's are methods that focus on good solutions in parallel, they do not iterate from invalid solutions. The approach used for finding good solutions thus differs between methods. This difference may determine which algorithm is best suited for a given situation.

Employing stochastic search to find good control solutions provides flexibility regarding control objectives and constraints since they can easily be changed. Nevertheless, calculation and search time is still in the region of hours and too long even for applications in quasi-static control. Although calculation and search time decreases with more powerful computers, Section 4.1 shows that complexity of the control task precludes avoiding excessive times through increases in computer speed. Adding modules, telescopic bars, new or multiple control objectives increases demand for computational power faster than expected processor speed increases.

5 Improvement of structural performance over time

Civil engineering structures are designed within the limitations of hazard scenarios and serviceability requirements that are governed by their intended use and location. Adaptation to changes in these specifications is difficult and might necessitate structural modifications. This statement is valid for conventional buildings as well as for structures that are equipped with active control systems. Although such installations may increase load-carrying range, they have mainly been conceived to protect structures against earthquakes and to enhance occupant comfort under high wind loading.

Results of the preceding chapters demonstrated that inner self-stress states of tensegrities could be varied to account for changing control objectives. Challenges include their geometric non-linear behavior, inaccuracies of the simulation and finding good control solutions when no closed form formulation of the system is available. These results form the modules to create a system that improves performance over time. Figure 5-1 proposes further extensions to the scheme presented in Chapter 4, Figure 4-1.

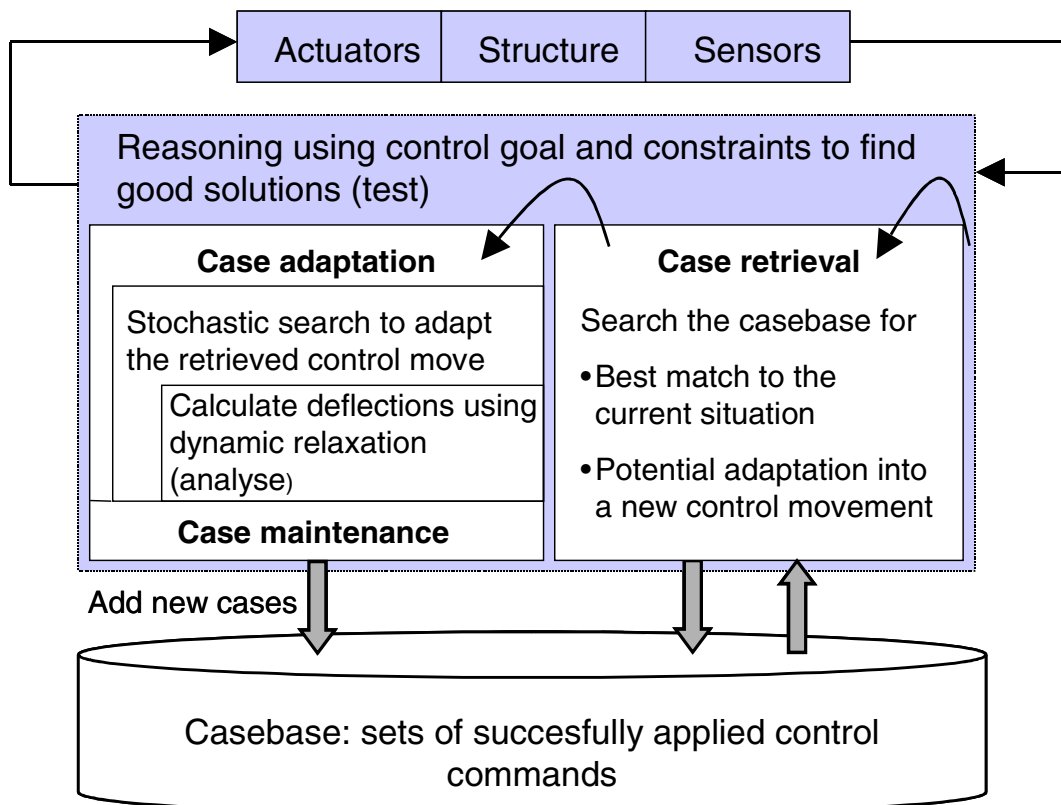


Figure 5-1 Schematic of intelligent structural control, integrating neural networks, stochastic search and case-based reasoning (system architecture)

Techniques for accuracy enhancement (neural networks, Chapter 3) and the search for good control solutions (stochastic search techniques, Chapter 4) are integrated into the reasoning process. In a practical scenario constant monitoring of the structure determines if a shape correction is necessary. If the answer is yes, the case-base retrieves a set of control commands (a case) that has already been successfully applied in the past and is close to the current situation. Stochastic search is employed to adapt the case.

The chapter is divided into three sections. The first section presents the CBR-system. The second section discusses results obtained by testing a prototype of this system on the three-module structure without active control components. In the third section, results are transferred to an actively controlled tensegrity structure consisting of five modules. Active control components employed are presented in Figure 5-2.

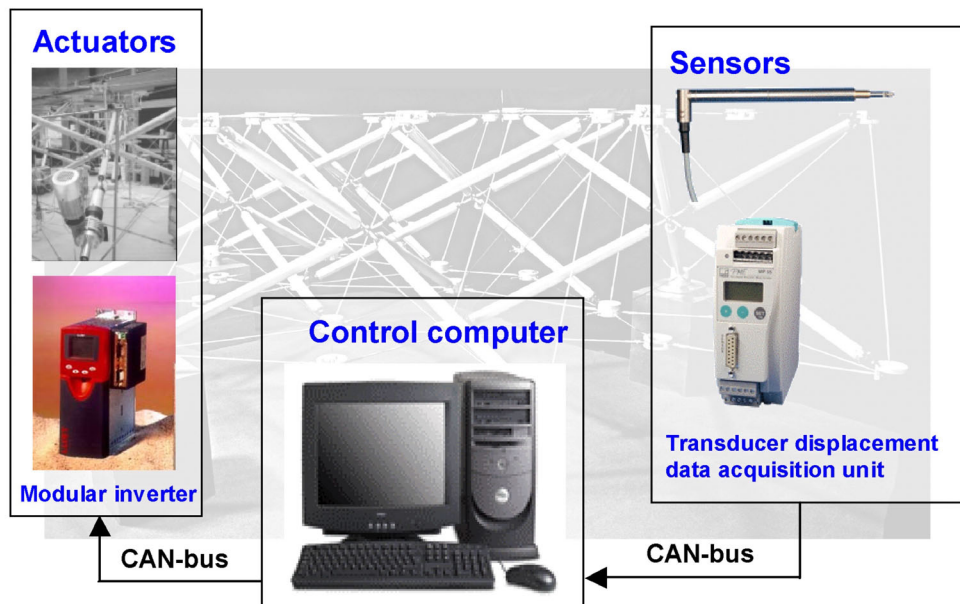


Figure 5-2 Active control components

Details of the control-system are given in (Fest 2002).

Topics that are addressed in this chapter are:

- A prototype system to increase control performance using case-based reasoning (CBR)
- Techniques for case adaptation
- Testing CBR on tensegrity systems with different control objectives and structural characteristics

5.1 CBR-system

Building on program modules that were developed and tested as described in previous chapters, a case-based reasoning system has been developed. The following modules were implemented:

- Retrieval (Figure 5-3)
- Adaptation (Figure 5-4)
- Application (Figure 5-5)
- Maintenance (Figure 5-6)

During retrieval, successfully solved control-tasks are compared to the current task. A set of cases ranked by their degree of similarity is proposed for adaptation (Figure 5-3).

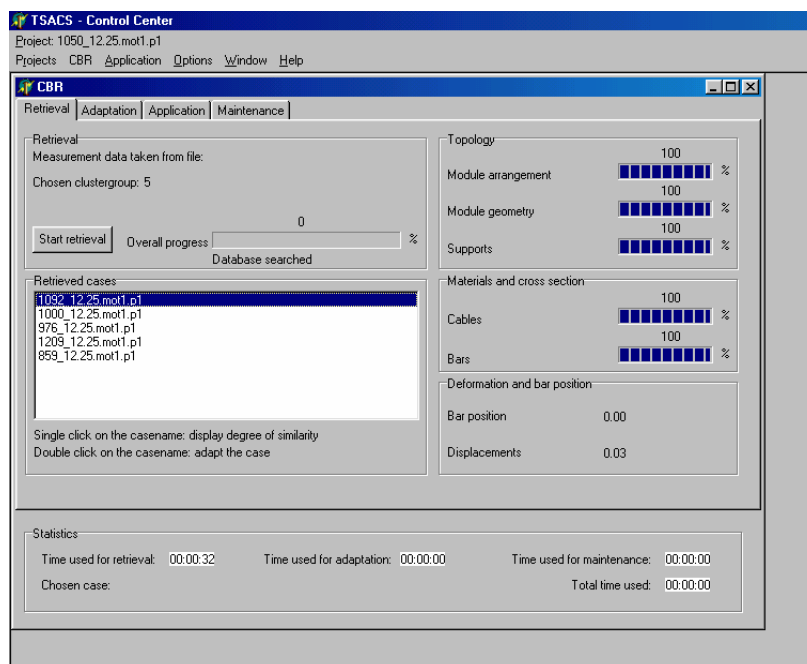


Figure 5-3 Screenshot of the Retrieval module

After choosing the case to adapt, Adaptation offers a choice of three stochastic search techniques to converge to a new control-command (Figure 5-4).

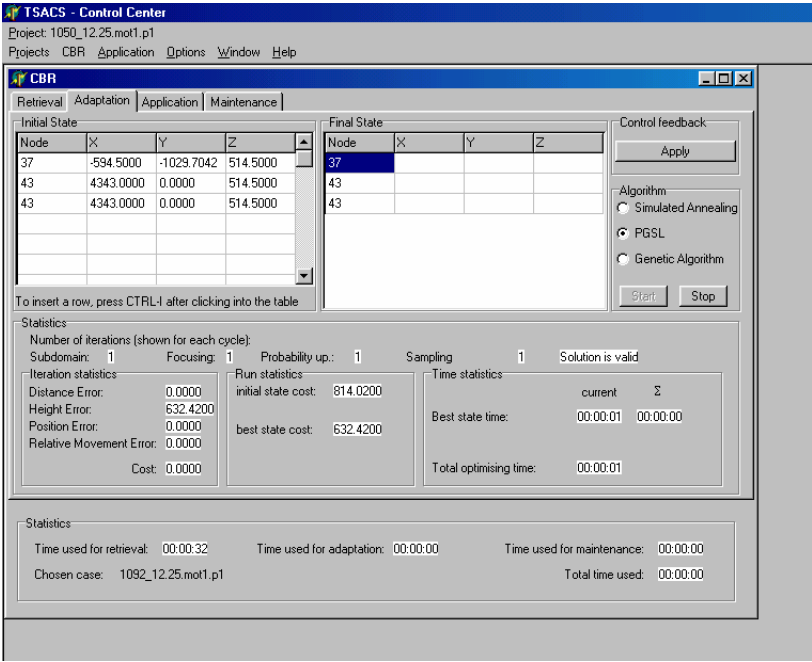


Figure 5-4 Screenshot of the Adaptation module

For case-adaptation, a parameter has been added to PGSL. This parameter limits the starting interval of allowable values around each optimization variable by a given percentage. A value of 100% represents a full-range starting interval. As a result, search is intensified in the region that is close to the solution to be adapted. Whereas no performance improvements are expected with sparse case-bases, this parameter may lead to improved performance when dense case-bases are available.

Characterizing a case-base as “sparse” or “dense” is difficult, since there is no relation between the total number of cases stored and their distribution with respect to the control-tasks. It might be convenient to define a *sparse* case-base as a case-base where stored cases are mainly pivotal (Section 2.4.2.2). Performance increases are possible by adding auxiliary and spanning cases. A *dense* case-base in contrast, contains already several of such cases. Adaptation success does not only depend on a single pivotal case. Good solutions might be obtained by starting adaptation from other cases as well.

Applying control commands to the structure in a single stroke by moving the telescopic bars at once all together might cause instabilities. Fest (2002) applied the following strategy:

- 1) Divide control movements into smaller commands.
- 2) Apply commands on a bar-by-bar basis, starting with contracting movements and finishing with extensive movements, until the final stroke is attained.

This strategy may lead to relaxing the structure first and re-tensioning it afterwards.

Commands are divided by a factor defined by the user. Since during each step only one telescopic bar is allowed to move, the total number of checks to be made can be easily calculated:

$$\text{Number of checks} = \text{Division factor} \times \text{number of telescopic bars} \quad (5-1)$$

Due to geometrically non-linear behavior, intermediate stages might cause instabilities. A sequence-check evaluates the entire path from the initial state to the entirely applied control

command by simulating each single step to avoid such problems. Commands are then transferred to the control software for application (Figure 5-5), (Fest 2002).

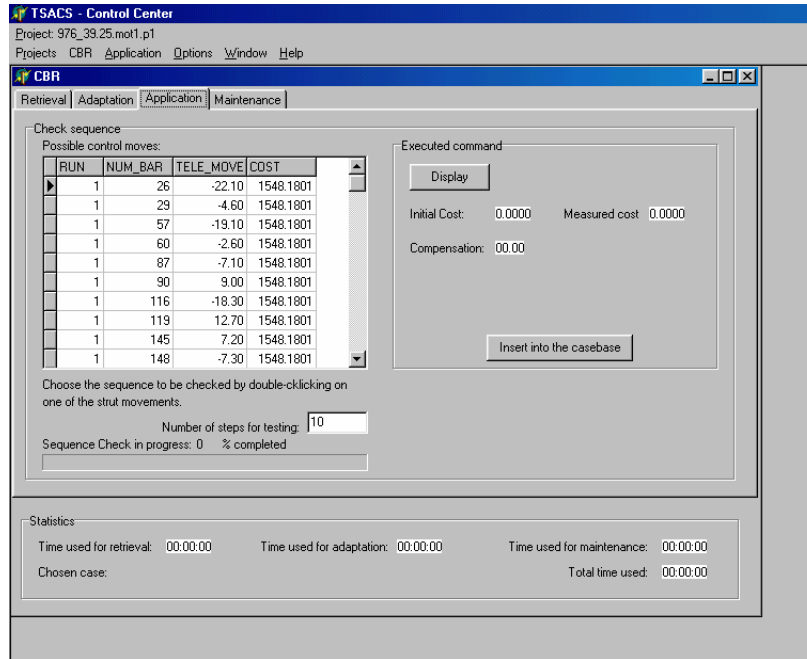


Figure 5-5 Screenshot of the Application module (sequence check)

Maintenance is linked with retrieval of similar cases. With a growing case-base, the number of cases to be compared with the current task increases. Each similarity calculation slows case retrieval time. Comparing only relevant cases reduces the total number of comparisons, thereby increasing performance. The maintenance module groups cases in clusters, employing k-means clustering (Section 2.4.3). After calculating the distance between the current control task and the centroid of each cluster, only cases around the nearest cluster are considered during retrieval (Figure 5-6).

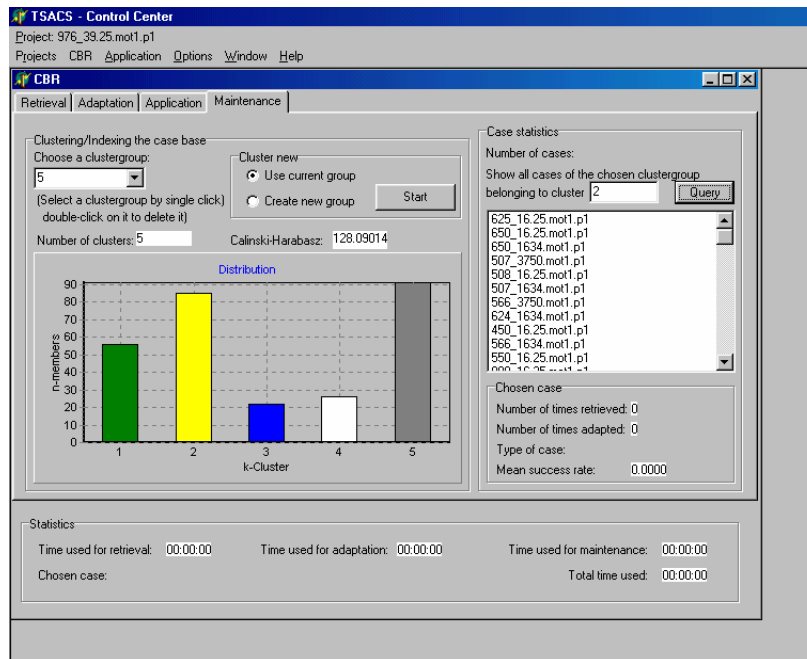


Figure 5-6 Screenshot of the Maintenance module

5.2 Prototype system for the three module structure

5.2.1 Case-bases for testing

Each control solution consists of descriptive case data (node positions, materials, topology, etc.) and a solution part (a set of telescopic bar commands). Data structures containing this information have been created for testing different stochastic search techniques for active structural control. After successful search, the solution part containing actuator movements is stored in the database.

Preceding tests (Chapter 4) provided solutions for seven control tasks. This number is insufficient for testing CBR-systems. Cases were designed to have variations in loading of these seven kernel cases. They have been optimized and added to the case-base. Each new task is affiliated to at least one kernel case.

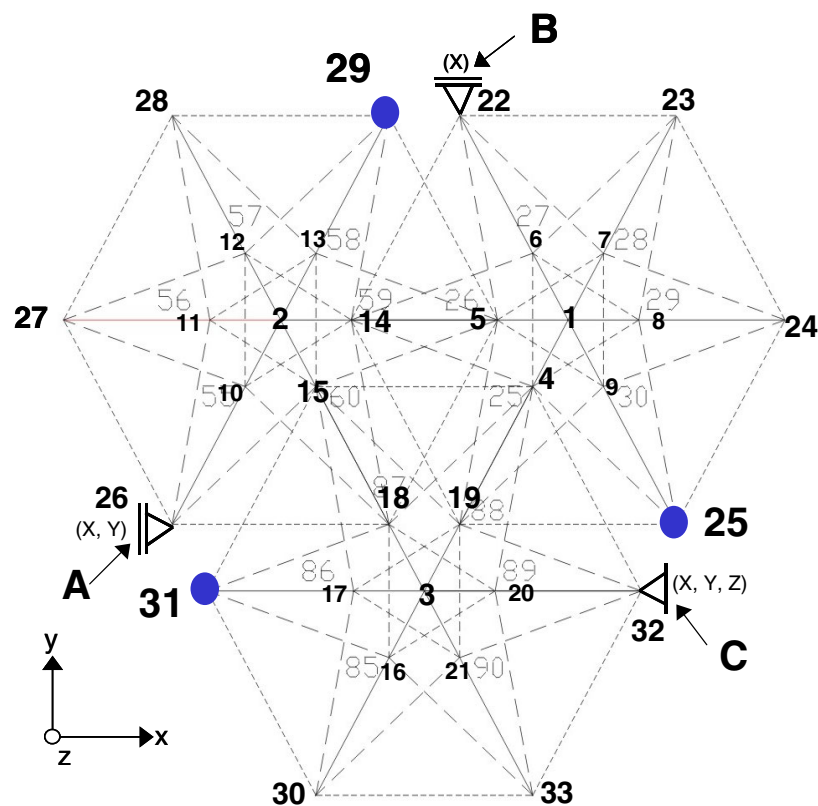


Figure 5-7 Three-module structure with loaded nodes; A, B and C are supports

Node loaded	ID	Loads applied [N]
23	1.1	310, 623 , 776, 981 , 1291
18	1.2	310, 623 , 776, 981 , 1291
27	1.3	310, 623, 776, 981, 1291
33	1.4	310, 623, 776, 981, 1291
5	1.5	310, 623, 776, 981, 1291
4/14	2.1	310, 608 , 776, 981 , 1291
5/18	2.2	310, 608, 776, 981, 1291
15/19	2.3	310, 608, 776, 981, 1291
23/27/33	3.1	130/327/327, 260/654/654, 388/981/981 , 518/1308/1308
24/28/30	3.2	130/327/327, 260/654/654, 388/981/981, 518/1308/1308

Table 5-1 Kernel cases and new load cases, kernel load cases are printed in bold letters, node numbers correspond to Figure 5-7.

In total, 48 control tasks have been created and stored in the database. The reference case-base (Case-base IV) contains all tasks. For testing purposes, three smaller case-bases have been derived by deleting 10 cases in each step while descending from case-base IV to case-base I.

Case base	I	II	III	IV
Number of cases	18	28	38	48

Table 5-2 Test case bases created

A general overview of the content of each case-base can be obtained from Table 5-3. Numeration of tasks might be deduced from Table 5-1:

- 1.1.310 Task with ID 1.1; magnitude of loading 310 N
- 2.3.981 Task with ID 2.3; magnitude of loading 981 N

The ID-tag describes the load-position. Control tasks 1-7 correspond to the seven kernel load cases and are, thus, named differently.

Nr.	Load Case	Database			
		I	II	III	IV
1	5b23n981	X	X	X	X
2	5b23n623	X	X	X	X
3	5b18n981	X	X	X	X
4	5b18n623	X	X	X	X
5	5b4e14n981	X	X	X	X
6	5b4e14n608	X	X	X	X
7	5b33e23e27	X	X	X	X
8	1.1.310	-	-	X	X
9	1.1.776	-	-	-	X
10	1.1.1291	-	X	X	X
11	1.2.310	-	-	X	X
12	1.2.776	-	-	-	X
13	1.2.1291	-	X	X	X
14	1.3.310	X	X	X	X
15	1.3.623	-	-	X	X
16	1.3.776	-	-	-	X
17	1.3.981	-	X	X	X
18	1.3.1291	X	X	X	X
19	1.4.310	X	X	X	X
20	1.4.623	-	X	X	X
21	1.4.776	-	-	-	X
22	1.4.981	-	-	X	X
23	1.4.1291	X	X	X	X
24	1.5.310	X	X	X	X
25	1.5.623	X	X	X	X
26	1.5.776	-	-	-	X
27	1.5.981	-	-	X	X
28	1.5.1291	-	X	X	X
29	2.1.310	-	-	X	X
30	2.1.776	-	-	-	X
31	2.1.1291	-	X	X	X
32	2.2.310	-	X	X	X
33	2.2.608	-	-	X	X
34	2.2.776	-	-	-	X
35	2.2.981	X	X	X	X
36	2.2.1291	X	X	X	X
37	2.3.310	X	X	X	X
38	2.3.608	X	X	X	X
39	2.3.776	-	-	-	X
40	2.3.981	-	-	X	X
41	2.3.1291	-	X	X	X
42	3.1.130	-	-	X	X
43	3.1.260	-	-	-	X
44	3.1.518	-	X	X	X
45	3.2.130	X	X	X	X
46	3.2.260	-	-	-	X
47	3.2.388	-	-	X	X
48	3.2.518	-	X	X	X

Table 5-3 Overview of cases, cases 1-7 are kernel cases

5.2.2 Similarity measurement and retrieval

The first phase in the use of a CBR-system consists of finding cases close to the current situation. Since the probability to retrieve an exact match is usually low, the solution part of a similar case has to be adapted. The challenge of designing procedures for similarity measurement involves

- Identifying the case features/properties which are essential for similarity
- Selecting a similarity metric to be employed.

Sometimes, case-descriptions have to be mapped to numerical values for retrieval. In the scope of this work, a mapping is not necessary, since numerical values of the same type are being compared. Attributes chosen for comparison are

- Nodal displacements
- Strut positions.

An assumption of this kind of retrieval is that a successful set of control commands can be re-used for the same state and load case of the structure. This corresponds to the first basic prerequisite of Leake and Wilson (1999):

- Problem-solution regularity: solutions of prior problems are useful starting points for solving current problems.

The second prerequisite

- Problem-distribution regularity: the case library will contain cases similar to the new problem

can be fulfilled within the scope of the tests.

To avoid additional complexity, structural properties of the task description, which have to correspond exactly to the stored task, are

- Number and arrangement of tensegrity modules
- Geometry of one tensegrity module
- Place and type of supports
- Materials used.

Only exact matches to the above attributes are proposed for case adaptation.

As the exact detection of place and magnitude of loads might be too complex in practical situations, measured nodal displacements are used to identify similar cases. Comparison of displacements employs a nearest neighbor metric, which has been discussed in Section 2.4.2.1:

$$\text{Distance}(X, Y) = \sqrt{\sum_{i=1}^d w_i^2 (x_i - y_i)^2} \quad (5-2)$$

W_i weight factor for the i^{th} attribute, set to 1 for all attributes

For most applications the number of measurements is limited. Selecting nodes to be used in the distance metric (Equation 5-2) involved simulating structural behavior for multiple load cases and choosing the nodes with the most significant overall displacements.

Selected nodes are identified by a circle in Figure 5-8. A verification process that checks whether the retrieval process with the chosen nodes identifies one of the affiliated seven basic cases as the most similar one is necessary. The comparison of telescopic bar positions evaluates always to '0' for performed tests, since stored solutions have been obtained by starting the optimization from the initial '0' bar position. As a result, only nodal displacements are relevant during distance calculation.

Exemplary for all intermediate load cases, results of testing retrieval for load case 1.1-776 are shown. The distance metric should retrieve the basic load case 2 as the most similar one (Figure 5-8).

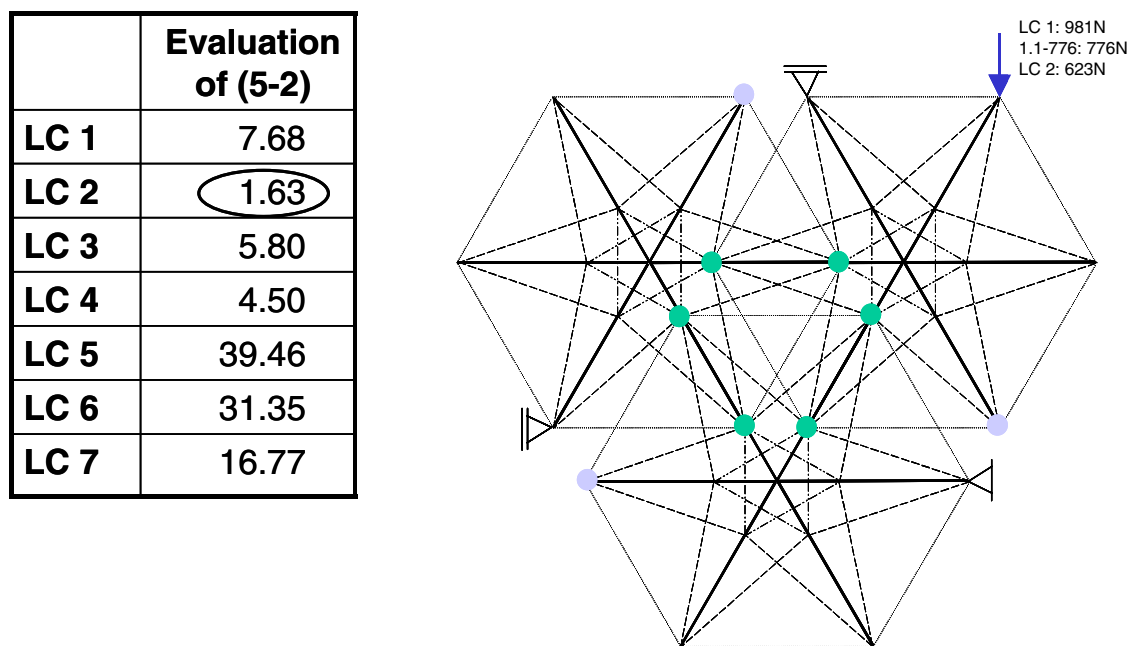


Figure 5-8 Distance metric for load case 1.1-776; circled nodes are used in the distance metric

Usually, degrees of similarity are normalized and given in terms of values between 0 and 1, where '1' presents identical and '0' opposite cases. However, as the maximum value for deflection to be used for normalization could not be fixed at this stage of the project, values are not normalized and transformed into a degree of similarity. Using equation (5-2), low values indicate good matches.

5.2.3 Tests performed on case bases

The different case-bases (Table 5-2) are used to:

- Compare results obtained by case-based reasoning to find control solutions for the “missing cases” in CBI-CBIII with results obtained using “pure” optimization. Algorithms had no fixed threshold and continued until the maximum number of iterations has been attained.
- Test which search algorithm (PGSL, simulated annealing, genetic algorithm) is most suited for case adaptation. For these tests, a threshold of 0.1 has been chosen for the objective function. Explanations regarding the threshold are given in the following section.
- Determine the influence of the case-base size on time and quality when searching for good control commands. A threshold of 0.1 has been chosen.

5.2.4 Comparing case-based reasoning with pure optimization

The number of iterations needed to attain the best solution is compared when

1. The optimization process is started with the initial strut length (Optimiser)
2. The optimization process is started with strut lengths obtained from a case that is the most similar to the current situation.

In both cases, the total number of iterations is fixed to 22,000 and PGSL is used. Case-base III is used for this test. The ten deleted control tasks are adapted (compare Table 5-3). Two control tasks where the CBR process outperformed the optimization regarding iterations needed and the best state cost are identified: 3.1-260 and 2.3-776. Other tasks finished with values slightly above the ones attained with pure optimization. Nevertheless, CBR needed less iteration: 1.3-776, 1.4-776, 2.1-776 and 3.2-260. For the remaining cases, no clear message was obtained.

Observations become clearer when cost development over iteration is compared in a graph for the two techniques used (Figure 5-9). Although control task 1.1-776 needs more iterations for the best state, it starts with significantly lower costs and converges faster to cost-function values around 0.1. This behavior can be generalized for other cases tested. Analyzing the measurement precision as well as the control system has motivated the choice of 0.1 as threshold value. Since the case-base is rather sparse, best results have been found with a wide starting interval of 90% (Section 5.1).

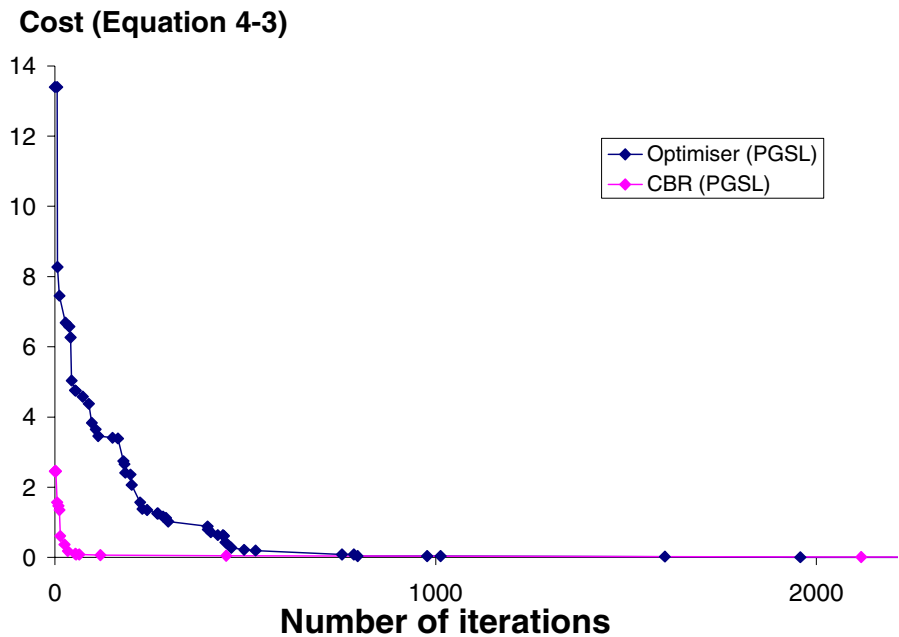


Figure 5-9 Gradient behavior for load case 1.1-776, database III

Gradient behavior has been found similar to Figure 5-9 with all other cases tested.

5.2.5 Size of the case base

Theoretically, performance of case-based reasoning systems should increase with the size of the case base. Case bases with different sizes (CBI-CBIII) have been compared to see how they influence search. The curve for control-task 1.3-776 is representative for other tasks. Case base III performed best and, thus, reinforces this assumption.

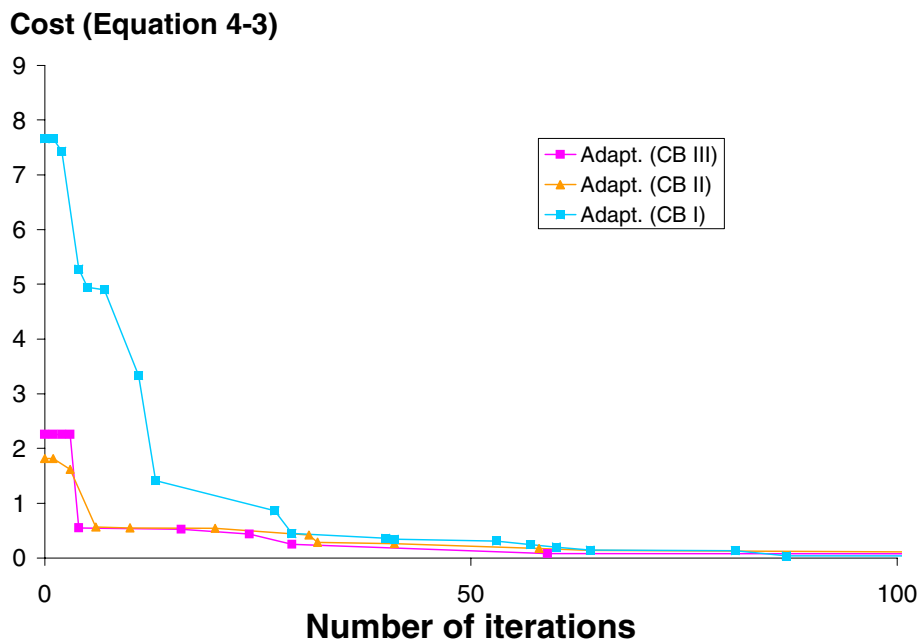


Figure 5-10 Results obtained using different case bases; adapted control task 1.3-776

5.2.6 Different search strategies for adaptation

Adaptation uses stochastic search. Methods tested are simulated annealing (SA), PGSL, genetic algorithms (GA) and a search (DH) that consists of the freezing part of simulated annealing only (Section 2.3.1.1). From a theoretical point of view, applying stochastic search should avoid retrieving cases that cannot be adapted at all. Case base III has been used throughout the comparison. A threshold of 0.1 is fixed.

Control Task	PGSL Iterations	SA Iterations	DH Iterations	GA Iterations	Optimiser Iterations
1.1-776	54	3,216	413	428	754
1.2-776	50	2,025	45	146	127
1.3-776	59	1,805	217	287	435
1.4-776	56	3,120	11	148	14,027
2.1-776	63	1,862	170	491	163
2.3-776	24	1,890	2,218	573	n.a.
3.1-260	41	1,884	25	146	13,557
3.2-250	50	1,846	1,920	495	10,747

Table 5-4 Case base III, Threshold: 0.1; n.a. = Threshold not attained after 22,000 iterations

Table 5-4 shows that PGSL performed best concerning the number of iterations needed to attain the threshold in most cases. The performance of undirected search (DH) can be considered good as well. Nevertheless, the risk of becoming trapped in local minima is greater than with other techniques. Search times (iterations needed) differ from case to case. Genetic algorithms showed stable behavior; they delivered good solutions at a constant rate of iteration. Simulated annealing did not decrease the number of iterations needed. The behavior of techniques is also illustrated in Figure 5-11.

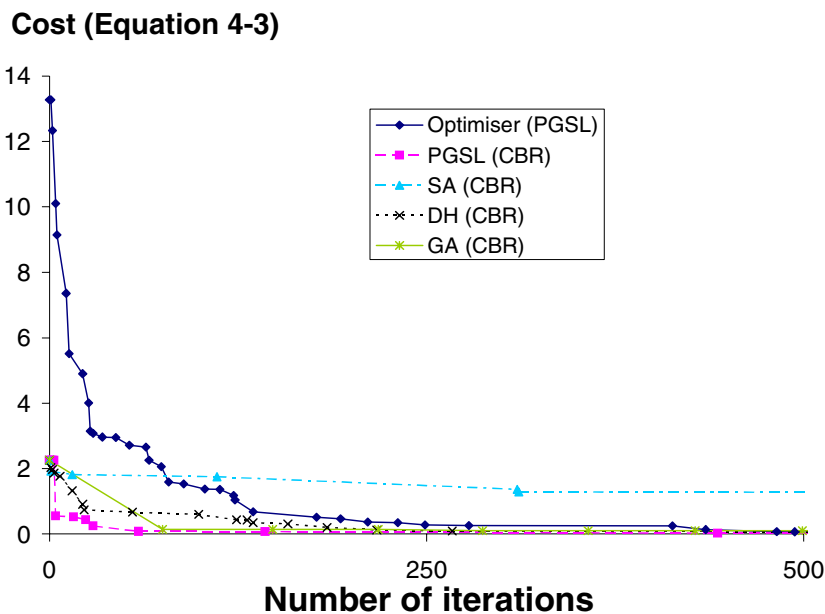


Figure 5-11 Gradient behavior for different search techniques and control-task 1.3-776

5.2.7 Discussion of results

Stochastic search is useful for case-adaptation. Regarding search techniques used, simulated annealing performed worst. This behavior can be explained with the “cooling schedule” which accepts worse solutions over a long period of iterations. Shortening this interval reduces the likelihood of the algorithm finding good solutions in some cases. Cutting down this period to ‘0’ limits the process to a kind of undirected search that has been tested as well (DH in Table 5-4). Although this strategy provides good results, there is a greater risk of getting stuck in local minima than with other methods. Both techniques will not be used in the following tests. PGSL and GA’s performed well, where GA’s showed a more stable convergence behavior.

In general, employing a case-based reasoning system to detect and adapt successful control commands speeds up the process compared to pure optimization by a factor varying between 10 and 200. The size of the case-base influences the search process. A saturation point exists, where adding cases does not result in significant increases in accuracy. Instead, the retrieval process slows down. Strategies need to be developed to counteract performance decreases.

5.3 Applying results to the five module structure

5.3.1 Changing the control objective

When extending the three-module structure by two modules (Section 3.1.2), the initial control objective of keeping a 0-slope has been changed (Fest 2002). A new serviceability criterion, which ensures that water drains from the region of node 43 to the edge marked by the nodes 37 and 48, has been chosen. The initial slope under dead load of the system, named $\text{slope}_{\text{initial}}$ in Figure 5-12, should be kept constant. The new cost-function (5-3) uses this value. The implementation in the control software was straightforward and consisted of an additional objective function available through the settings menu.

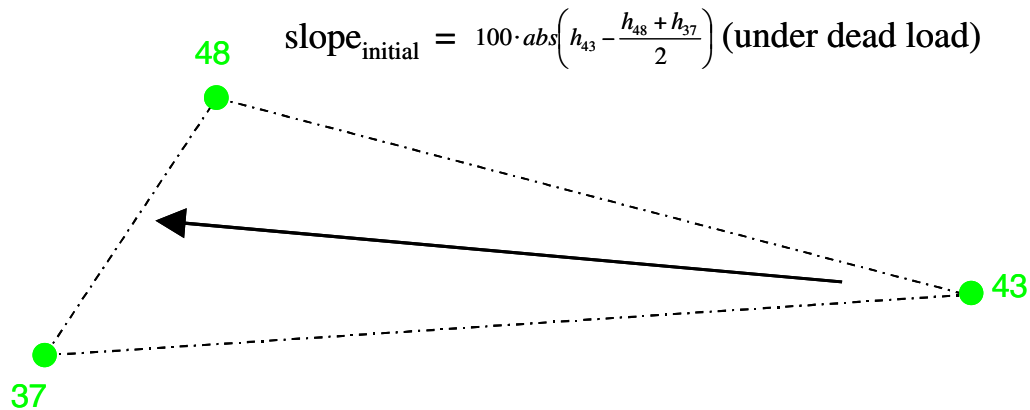


Figure 5-12 Control objective

$$\text{Cost} = \text{slope}_{\text{initial}} - 100 \cdot \text{abs}\left(h_{43} - \frac{h_{48} + h_{37}}{2}\right) \quad (5-3)$$

With the new control-objective, a constraint has been added. It insures a minimum bar compressive force to avoid tension in bars. The value has been fixed to 2000 N.

Figure 5-13 shows a planar view of the tested structure. The triangle observed for the new control objective is indicated by a dotted line. Telescopic bars are emphasized with thick continuous lines. Circled node numbers are loaded nodes, boxed node numbers are nodes used for the distance metric.

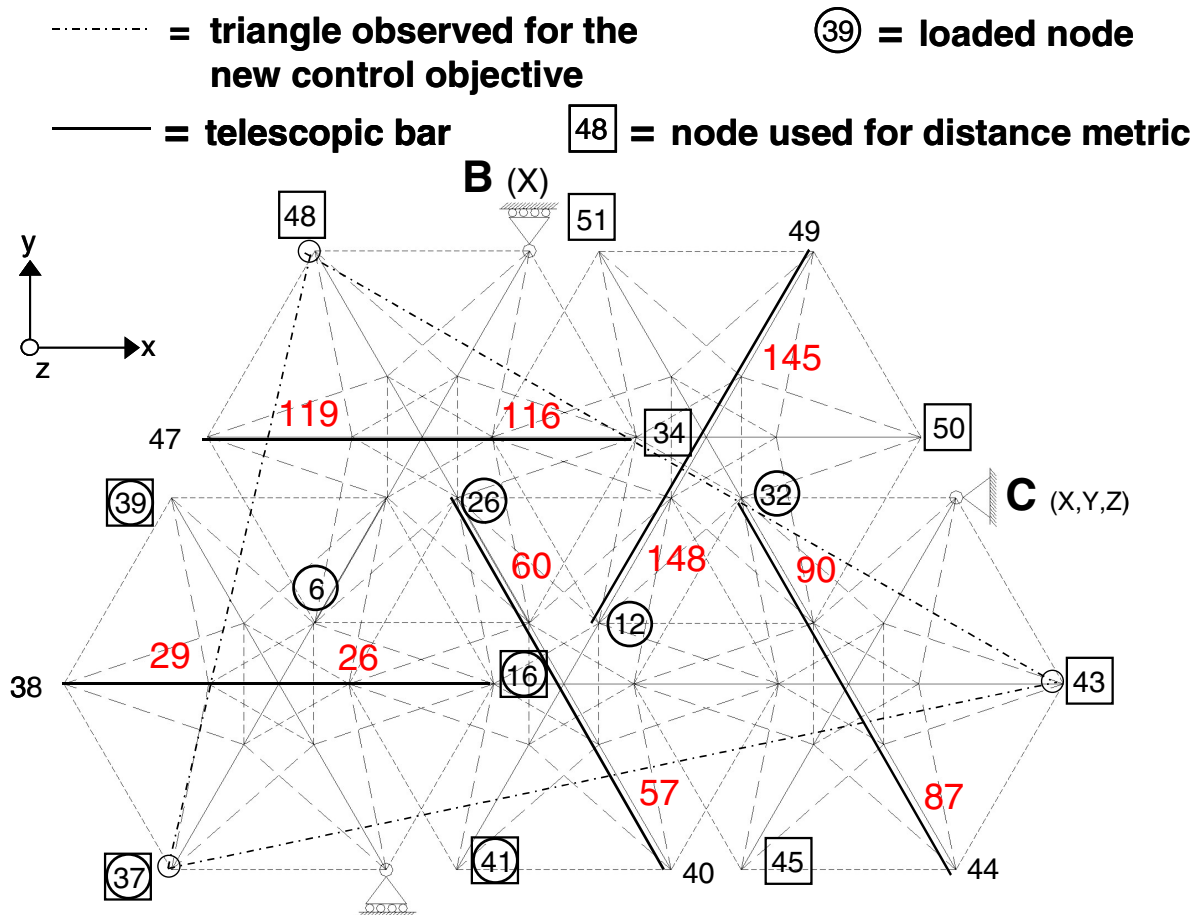


Figure 5-13 Five-module structure with loaded nodes and controlled struts; A, B and C are supports

5.3.2 Creation of a reference case-base

To start testing case-based reasoning in conjunction with the real structure and the control setup, a new reference case-base has been created. The process is similar to the one described for the three module structure. For the one node task, rows with load magnitudes 200, 300, 350, 450, 550, 650, 700, 800, 900, 1000 and 1050 have been added to load cases used in Section 3.6. Loads 300, 650 and 700 have extended the two-node task. Compare Table 5-5 and Table 5-6, which also list the initial cost calculated with Equation 5-3 for each task.

Load [N]	Nodes								
	6	12	16	26	32	37	39	41	48
150	55.73	8.52	19.97	103.30	106.18	159.28	242.53	2.02	163.63
200	74.43	11.73	26.62	137.08	141.48	212.63	323.28	2.82	218.38
274	102.18	15.77	36.37	186.58	193.38	291.78	442.63	4.17	299.43
300	112.03	17.32	39.82	203.73	211.58	319.53	484.53	4.72	327.98
350	130.98	20.22	46.32	236.68	246.48	373.18	565.18	5.77	382.88
391	146.48	22.72	51.77	263.53	275.13	417.23	631.00	6.67	427.93
450	168.93	26.27	59.47	301.73	316.13	480.78	726.28	8.22	492.98
508	191.18	29.82	67.07	338.88	356.33	543.38	819.73	9.72	556.93
550	207.28	32.32	72.57	365.53	385.33	588.78	887.43	10.87	603.33
625	236.08	36.92	82.27	412.73	436.88	670.08	1008.38	13.02	686.33
650	245.78	38.52	85.62	428.23	454.08	697.23	1048.58	13.87	714.08
700	265.18	41.57	92.07	459.28	488.38	751.53	1129.48	15.52	769.48
742	277.05	44.27	97.52	485.03	517.03	797.38	1197.53	16.87	816.13
800	303.63	47.77	104.97	520.38	596.53	860.68	1291.63	18.88	880.63
859	325.98	51.52	112.62	412.68	596.55	928.23	1387.73	21.09	976.46
900	341.73	54.17	117.87	580.48	624.23	970.28	1454.63	22.65	992.28
976	370.83	59.07	127.67	625.53	675.53	1053.83	1579.03	25.76	1077.18
1000	380.08	60.57	130.72	639.58	691.68	1080.23	1618.38	26.72	1104.13
1050	399.33	63.72	137.07	668.83	725.18	1135.38	1700.58	28.83	1160.18
1092	414.13	66.37	142.47	693.18	753.30	1192.93	1786.34	31.14	1286.68
1209	461.24	73.97	157.37	760.18	831.13	1311.58	1963.60	36.01	1339.01

Table 5-5 Single node loading with initial costs, node numbers corresponding to Figure 5-13

Load [N]	Nodes						
	37 & 48	48 & 45	37 & 45	41 & 50	16 & 34	39 & 48	37 & 50
98	211.13	13.72	97.13	103.27	87.57	265.43	0.58
157	338.88	160.03	155.28	164.17	132.12	425.48	0.98
215	464.93	218.68	212.33	223.17	180.72	583.03	1.53
274	593.98	278.18	270.13	282.32	230.12	743.43	2.28
300	650.43	304.33	295.53	308.17	251.28	814.28	2.53
332	720.63	336.48	326.68	339.67	278.62	901.48	3.03
391	850.18	395.58	383.93	397.07	327.72	1062.43	3.93
449	978.23	455.33	440.03	452.67	375.92	1221.18	4.93
507	1106.68	510.93	495.93	507.37	424.12	1380.08	5.93
566	1237.88	569.13	552.43	562.32	472.92	1542.73	7.18
624	1367.38	626.23	607.78	615.37	520.72	1703.33	8.48
650	1425.53	651.63	632.38	638.92	524.27	1775.63	9.18
700	1537.73	700.48	679.78	683.67	583.32	1915.08	10.43

Table 5-6 Loading on two nodes with initial costs, node numbers corresponding to Figure 5-13

Solutions are found through pure optimization employing PGSL as algorithm. Threshold was set to 1. This is a trade-off between measurement accuracy, which has been evaluated to 20 (Fest 2002) and the objective to have a sufficiently long iterative period of search for algorithm comparison. Three runs have been launched, each time providing the system with three different solutions for each control task.

Cells shaded in gray represent control tasks that have been evaluated on the structure . These cases will be re-used for testing the CBR system; they have been chosen on the basis that initial costs should be ≥ 400 .

5.4 Evaluation of case-based reasoning

5.4.1 Similarity measurement

Analogous to the distance metric employed for the three-module structure, a nearest neighbor approach is used for the new configuration. Complete similarity (100%) evaluates the function to 0. Complete similarity is required for

- Number and arrangement of the modules
- Geometry of each module
- Materials used.

Distance is checked for nodal deflections and position of telescopic bars. As with the three-module structure, control tasks tested as well as cases stored in the case-base do not contain initial telescopic bar movements different from '0'. This means that the obtained value by the distance function represents the differences in nodal deflections. Nodes chosen are a tradeoff between the ones representing the most important deflections and measurement constraints: 16, 34, 39, 41, 45, 50, and 51. Nodes included in the control objective, 37, 43, 48 have been added. As discussed in Section 5.2.2, comparison of initial bar positions evaluates to '0'.

Testing checked if the appropriate control solution (case) is retrieved. An appropriate control solution is defined as the case with slightly different loads on the same node or nodes as the current control task. For example, searching for solutions that are similar to control task 550_37 (a load of magnitude 550 N at node 37), should result in proposing case 508_37 (load of magnitude of 508 N at node 37). Cases that are shaded in Table 5-5 and Table 5-6 have been tested. For single loading, appropriate solutions have been retrieved in all situations. For double loading, tasks with small loads (157-215 N) did not succeed in identifying appropriate cases. Nevertheless, retrieval for all tasks having load magnitudes above 215N resulted in appropriate cases.

5.4.2 Performance increases through case-based reasoning

5.4.2.1 Test description

Control commands obtained employing stochastic search (PGSL) have been validated on the real structure (Fest 2002). Slope compensation was $\langle 78\%$ for single loading and $\langle 72\%$ for double loading. Results presented here are intended to evaluate performance increases when a case-base is used. Setup is similar to tests conducted on the three-module structure. Differences are

- Techniques employed and compared have been reduced to the two most advantageous ones: PGSL and GA's
- The case-base used for retrieval represents a much broader scope of the space of possible solutions and should avoid similarity check difficulties as encountered with the first prototype.

- Tests have been performed for all cases marked in gray in Table 5-5 and Table 5-6. Each adaptation process has been carried out 3 times, the results of the medium best solution have been chosen and plotted in graphs.

After testing, parameters for the search techniques have been kept constant throughout testing (Table 5-7, Table 5-8). Parameters and their choice adaptation to the task to be optimized have been described in Section 2.3.2.2 (PGSL) and Section 2.3.3.2 (GA's).

Parameter	Value
Number of sampling cycles (NSC)	2
Number of probability updating cycles (NPUC)	1
Number of focusing cycles (NFC)	100
Number of subdomain cycles (NSDC)	20

Table 5-7 Parameters employed with PGSL

Parameter	Value
Population size	40
P_{cross}	0.9
P_{mut}	0.01
Num_gen	50
Selection	Tournament selection
Encoding	Gray scale
Elitism	1 member copied

Table 5-8 Parameters employed with GA's

Test series are described below:

Series 1: Testing potential of CBR to improve performance

This test is similar to those made on the three-module structure. Performance of “pure” optimization is compared with case-based reasoning, using PGSL and GA's for adaptation.

Series 2: Comparison of GA's for “pure” optimization with GA's for adaptation

In Series 1, the impact of starting genetic algorithms from a similar task is only compared with pure optimization using PGSL. An objective comparison needs results from pure optimization employing GA's.

Series 3: Testing clustering to overcome maintenance problems

The way cases are organized in a case-base influences efficacy of retrieval. K-means clustering is tested as methodology to decrease retrieval time.

Series 4: Performance enhancement over time

By constantly adding good cases to the case-base, adaptation performance should increase. The influence of case-base size has been tested.

5.4.2.2 Series 1: Potential of CBR for performance improvement

As in Section 5.2.4, best-so-far curves are used to compare performance. In general, adaptation employing genetic algorithms converged faster than pure optimization. Three different starting-interval limitations have been used when testing PGSL for adaptation: 100, 90 and 30%. The example in Figure 5-14 shows one case where PGSL with no starting interval limitation (100%) converges faster than pure optimization.

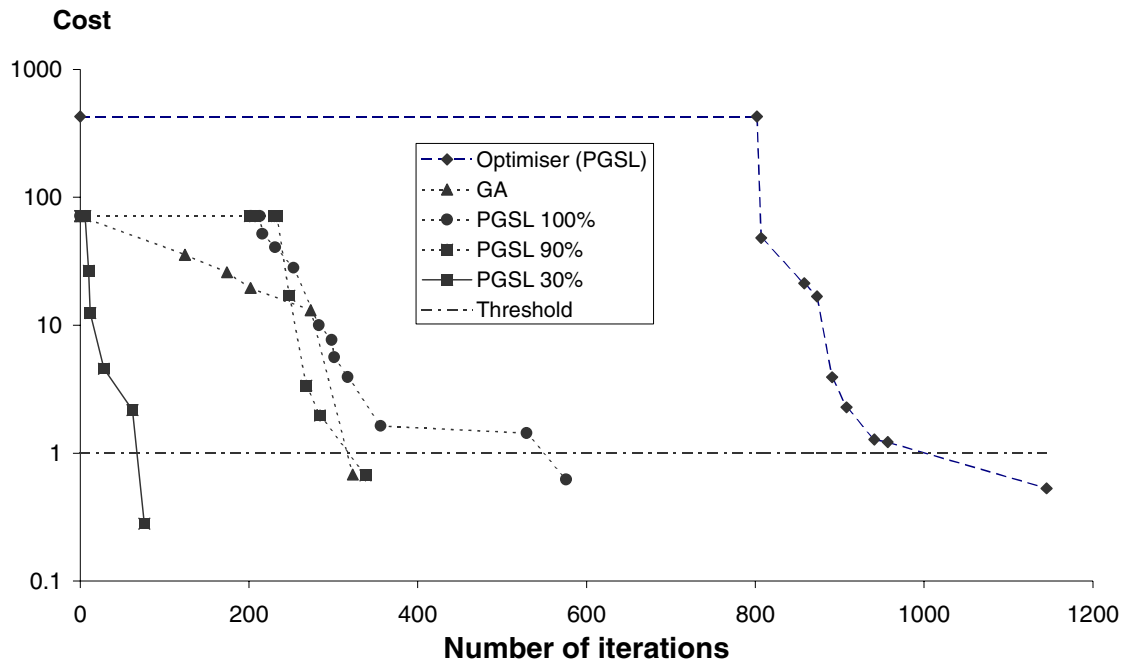


Figure 5-14 Control-task 391_48 adapted with results from control-task 350_48

Not every case converged faster when adapted without starting interval limitations (Table 5-9).

Faster	Almost the same	Slower
391_37	625_32	550_48
391_48	700_37	700_48
550_37	449_1634	859_32
625_26	624_4150	900_26
1209_26		1092_32
157_3948		215_3948
391_3745		274_3948
391_4150		624_4845
391_4845		624_3745
507_1634		
700_4845		
700_3745		
700_1634		

Table 5-9 Performance of adaptation using PGSL compared with pure optimization without limiting the starting interval

There is a weak tendency for PGSL to perform better than pure optimization when started from a good case. Nevertheless, no clear conclusions can be drawn. Experience during simulations of the three-module structure indicated that limiting the starting interval to 90%

improved performance. Table 5-10 compares adaptation with a starting interval of 90% with adaptation employing a starting interval of 100%.

Faster	Almost the same	Slower
391_48	550_37	391_37
550_48	1209_26	900_26
650_26	274_3948	157_3948
650_32	624_4845	391_3745
700_37	700_3745	507_1634
700_48		391_4150
859_32		700_4845
1092_32		700_1634
215_3948		
391_4845		
449_1634		
624_3745		
624_4150		
700_4150		

Table 5-10 Performance of adaptation using a starting interval limited to 90% compared with a starting interval of 100 %

Although limiting the starting interval to 90% did not significantly increase the performance of adaptation compared to pure optimization with PGSL, a tendency to converge faster than an unlimited starting interval is observed. In addition to that, the case-base used is rather dense. Task-solutions might be very close to the case chosen to adapt. This led to further restricting the starting interval to 30 %. Performance increased significantly. **With this setting, PGSL performed better than any other adaptation technique in every tested case** (Table 5-11).

Case	Iterations	
	Pure optimization	PGSL
391_37	540	61
391_48	1145	76
550_37	762	70
550_48	956	330
625_26	1102	270
625_32	746	300
700_37	984	96
700_48	1122	241
859_32	388	322
900_26	350	12
1092_32	738	88
1209_26	1172	74
157_3948	386	114

215_3948	1120	114
274_3948	490	62
391_4845	940	102
391_3745	682	300
391_4150	878	242
449_1634	380	224
507_1634	736	228
624_4845	602	74
624_3745	380	26
624_4150	2344	336
700_4845	1472	112
700_3745	930	68
700_4150	n. a.	348
650_1634	478	302

Table 5-11 Comparison of pure optimization and PGSL with a starting interval limitation of 30%

A representative comparison of pure optimization with optimization employing different starting interval limitations and GA's is shown in the figure below.

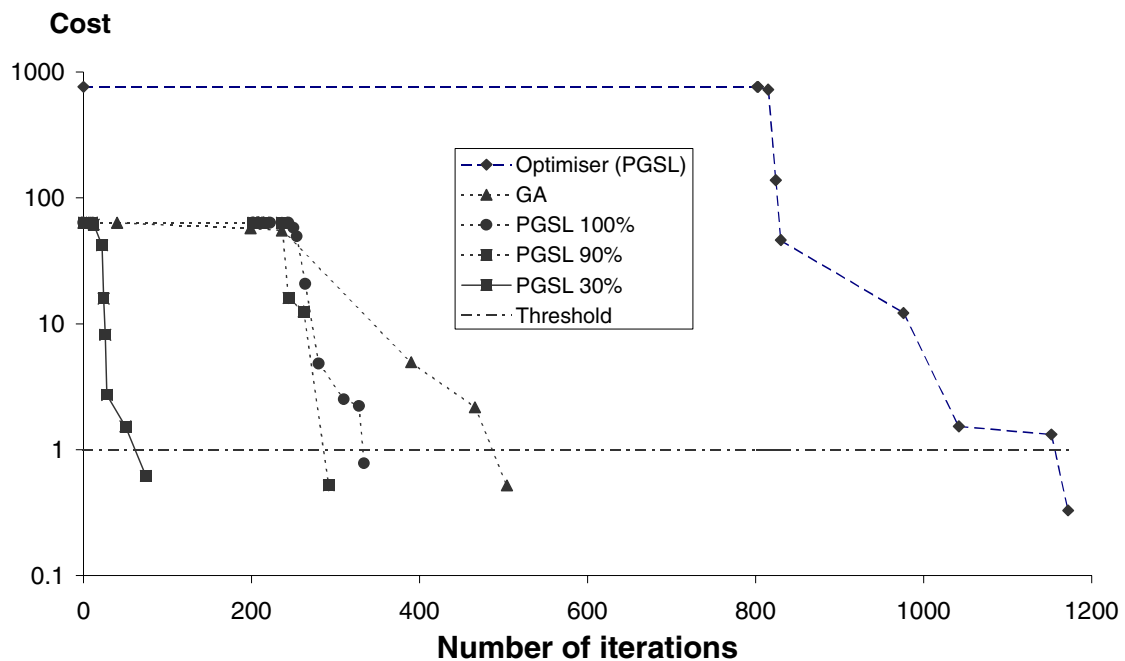


Figure 5-15 Comparison of pure optimization (Optimiser) with adaptation for control-task 1209_26, adapting case 1092_26

5.4.2.3 Series 2: Take GA's advantage from being launched from a good control task

Pure optimization employed PGSL to find good control commands. Although genetic algorithm converged faster when started from an already solved situation, this does not indicate that they performed better than pure optimization employing genetic algorithms as well. For comparison, control tasks have been optimized with genetic algorithms.

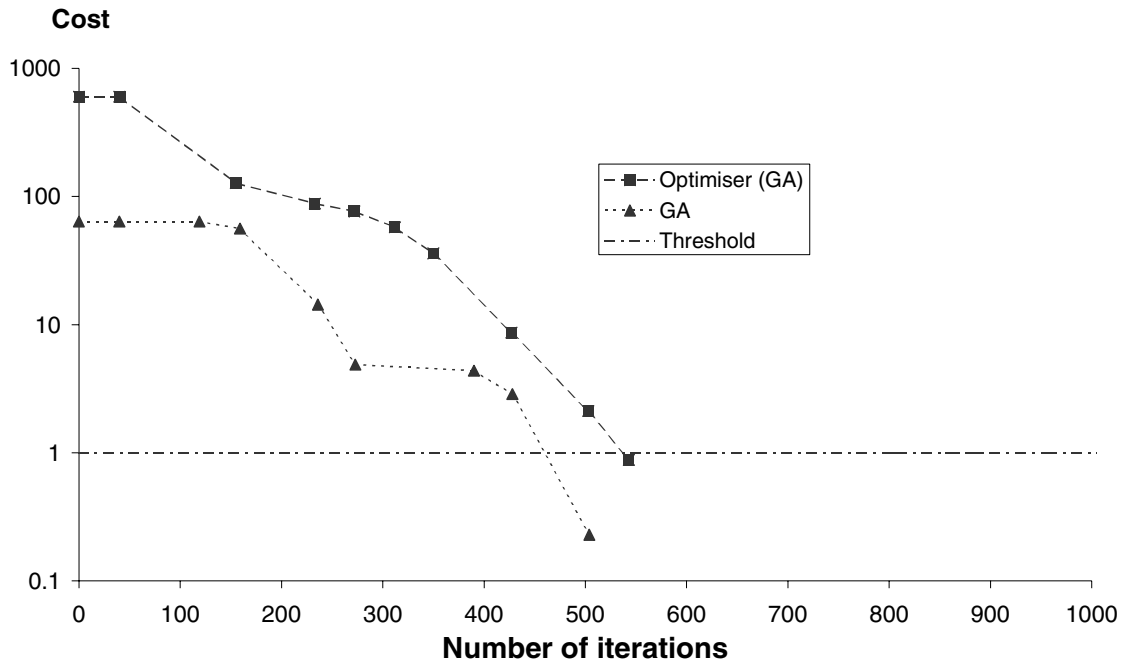


Figure 5-16 Comparison, 'pure' optimization and adaptation for control task 550_48

In Figure 5-16 the number of iterations needed is almost the same for adaptation and pure optimization. The same behavior is observed for tasks 625_32 and 624_4150. For tasks 700_48 and 700_4845 adaptation performed even worse than pure optimization. This indicates that GA's are not as advantageous for adaptation as PGSL.

5.4.2.4 Series 3: Testing clustering to overcome maintenance problems

Preliminary tests pointed out that retrieval and thereby the process of calculating case distances might outweigh computational time needed for adaptation. Clustering cases to limit the number of cases to be examined during retrieval is proposed to speed up this process. Different numbers of clusters have been evaluated on the set of cases present in the reference case-base. As quality criteria to indicate good clusterization, Calinski-Harabasz (C-H) has been employed (Section 2.4.3).

K	C-H
4	128.0901
5	122.8801
20	87.2135
35	150.1871
80	188.3619
100	220.9944
150	172.0012

Table 5-12 C-H evaluated for different values of K on the reference case-base (K=number of clusters)

The three most promising ones, k=5, k=35 and k=100, have been tested for cases marked in gray in Table 5-5 and Table 5-6. Results are given in Table 5-13.

Control-project	Retrieved cases							
	k=0		k=5		k=35		k=100	
	Case	Time [mm:ss]	Case	Time [mm:ss]	Case	Time [mm:ss]	Case	Time [mm:ss]
391_48	350_48 450_48 215_37&48 508_32 157_37&48	02:18	350_48 450_48 215_37&48 508_32 157_37&48	00:44	350_48 450_48 215_37&48 508_32 550_32	00:14	450_48 215_37&48	00:12
550_48	508_48 274_37&48 625_48 300_37&48 450_48	02:19	508_48 274_37&48 625_48 300_37&48 450_48	00:44	274_37&48 300_37&48 742_32 700_32 508_37	00:10	508_48	00:11
625_26	650_26 700_26 550_26 742_26 508_26	02:18	650_26 700_26 550_26 742_26 508_26	00:49	650_26 550_26 508_26 450_26 391_26	00:09	650_26	00:11
215_39&48	300_39 350_39 274_39 157_39&48 274_39&48	02:23	300_39 350_39 274_39 157_39&48 274_39&48	00:49	300_39 274_39 157_39&48 508_48 450_37	00:14	-	00:11
274_39&48	300_39&48 391_39 332_39&48 215_39&48 450_39	02:23	300_39&48 391_39 332_39&48 215_39&48 450_39	00:50	300_39&48 391_39 332_39&48 450_39 350_39	00:09	300_39&48 391_39 450_39 350_39	00:13

Table 5-13 Exemplary results for different values of k

Time decreased significantly for k=5. Retrieved cases are identical with those retrieved for k=0. More precisely: without decreasing retrieval quality, clustering resulted in a speed up by a factor of 3. With a value of k=35, results are still excellent: in most cases, at least the first one or two cases proposed were still the same. A supplementary speed up by the factor of three has been obtained. For k=100, accuracy decreased further without increases in speed.

5.4.2.5 Series 4: Performance enhancement over time

Compared to pure optimization, performance increases are observed when adapting good solutions close to the current task. In practical situations, these solutions are added to the case-base after validation on the structure. By constantly adding good cases, the case-base reasoning system should improve performance over time.

A growing cases-base is simulated in the following. Case-base I might represent a system at the beginning of its learning process; whereas case-base V could be close to a system that has already solved multiple cases and can be considered as “experienced” (Table 5-14).

Case base	I	II	III	IV	V
Number of cases	30	80	143	210	280

Table 5-14 Test case bases created

Case-base sizes in-between have been obtained by deleting four cases in each step while descending from case-base V to case-base I. Irregularities in the number of cases stem from the fact that the 28 cases discussed in Section 5.4.2.2 have been kept for testing purposes. PGSL has been used for all adaptation processes.

Results have been plotted in graphs that compare the different stages of the growing case-base and pure optimization regarding iterations needed to attain a best state. Starting with pure optimization at the left and increasing case-base size from left to right the number of iterations should decrease. This behavior is found, for example, for control-tasks 900_26, 274_3948 and 624_3745 (Figure 5-17, Figure 5-18 and Figure 5-19).

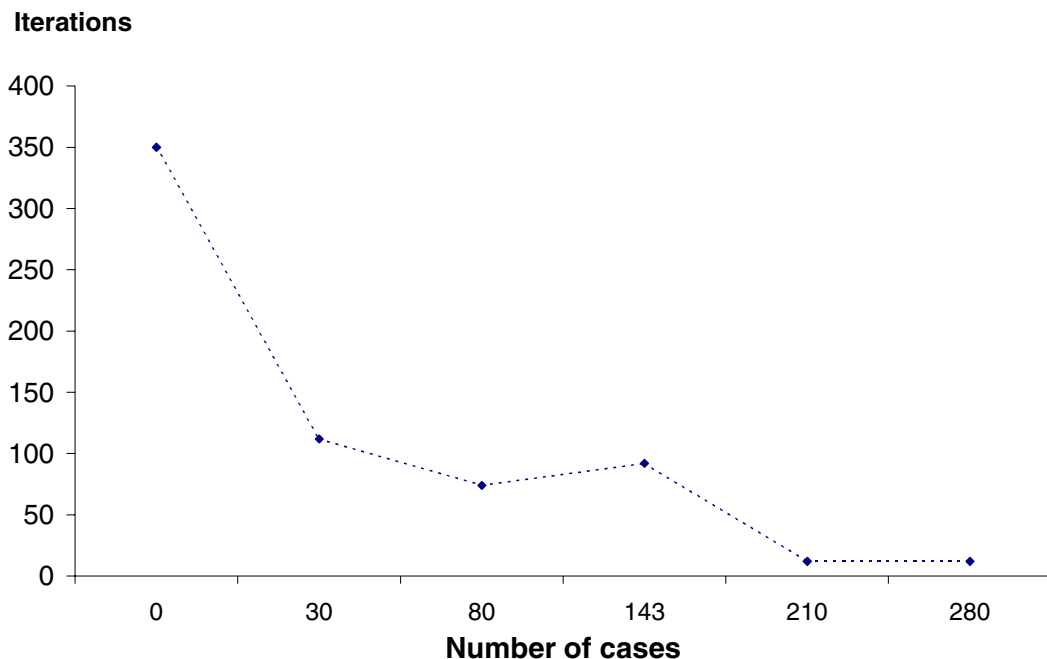


Figure 5-17 Improvement of performance for control-task 900_26

In some situations, performance decreased slightly when using a bigger case-base. Decreases are not significant, however, they can be related to the stochastic nature of the adaptation process.

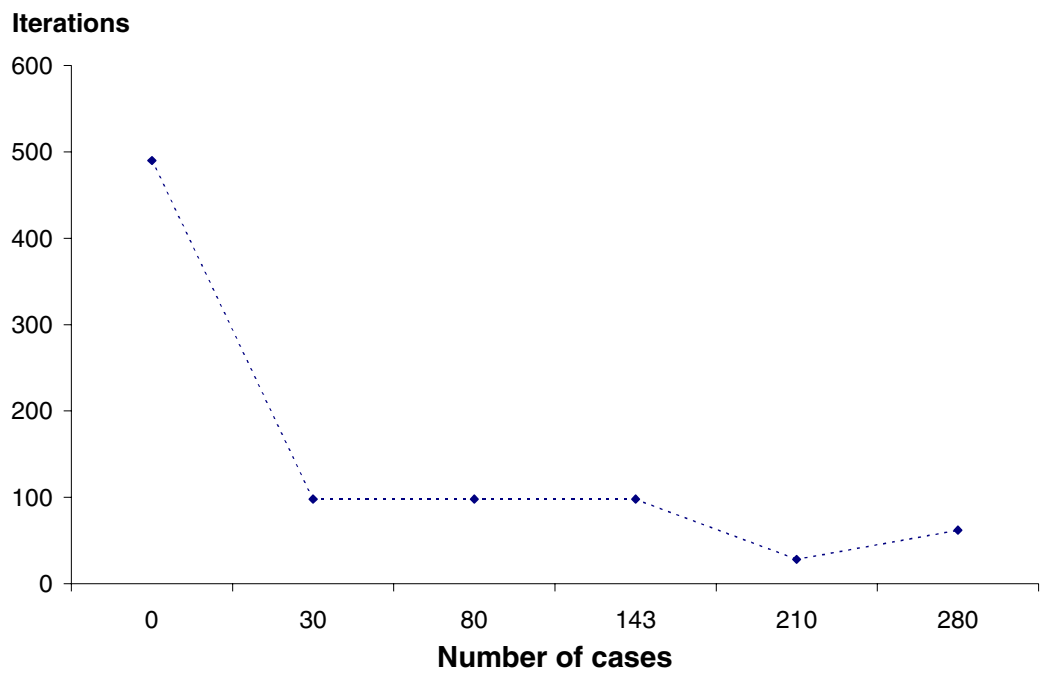


Figure 5-18 Improvement of performance for control-task 274_3948

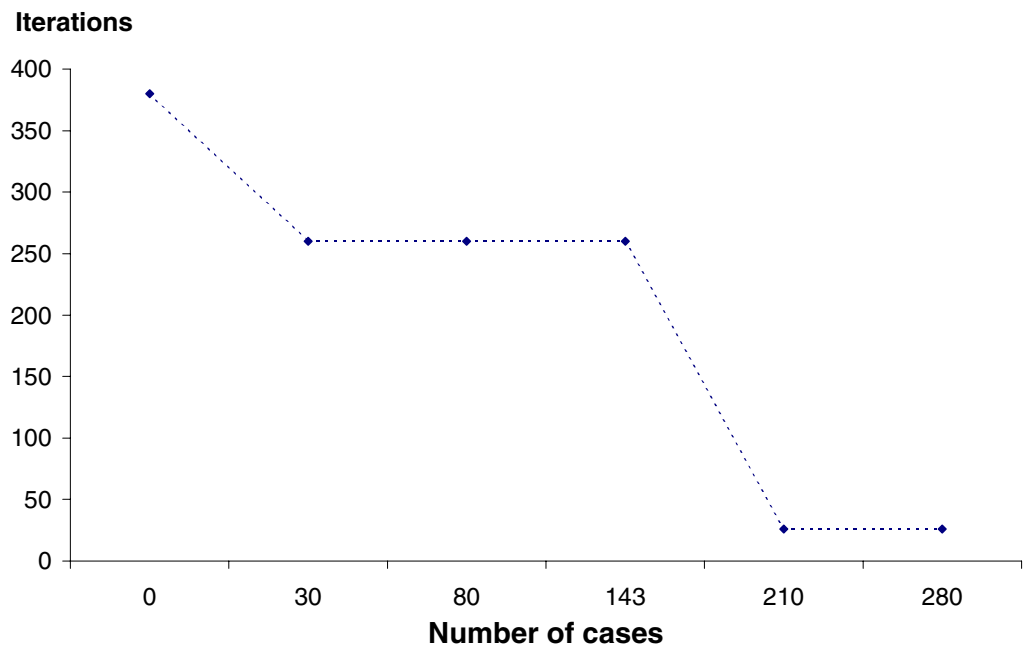


Figure 5-19 Improvement of performance for control-task 624_3745

5.5 Discussion of results

Case-based reasoning using PGSL during adaptation performs better than optimization alone for two different structural setups and two different control objectives:

1. Three-module structure with a 0-slope objective
2. Five-module structure with a constant slope objective

Starting interval limitations differed. Whereas for the five-module task rapid convergence is achieved with a limitation of 30%, the three-module task performs best with a value of 90%. This behavior indicates case-base density and gives also a first impression of the solution landscape of the chosen control objective. Solutions are apparently regularly distributed around tested cases. This might indicate a relation between starting interval limitation, case-base size and chosen control objective.

Although GA's were not advantaged from being launched close to a good solution, they converged faster than PGSL during pure optimization. Stable iterative performance observed in conjunction with the three-module structure leads one to propose an application using GA's instead of PGSL during "pure" optimization. Seeding multiple good solutions into the initial population might increase performance during adaptation.

The presence of multiple techniques for adaptation and optimization is still justified, since techniques are task dependent and preferences might change with changing tasks.

The total number of cases stored in the case-base is only one indicator for system performance. Priority should be given to cases that can easily be adapted and fit well into the space of solutions needed for anticipated tasks. The distance metric helped to retrieve good cases for adaptation, but did not retrieve the optimal case for adaptation in every case. This aspect requires further study.

Although the case-base for the five-module structure contains many more cases than the one used for the three-module structure, retrieval time of cases is less than adaptation time. Clustering shows potential to speed up case-retrieval without affecting the system's competence. The interesting aspect of clustering is that the same distance-metric that is used for retrieval can be applied. Performance decreases observed with $k=100$ can be explained with the time the clustering algorithm needs to find the closest centroid. Competence reduction is related to the size of the case-base (280 cases). Only a small number of cases are in each cluster. Virtual centroids might be seen as pivotal cases in the classification of (Smyth and Keane 1995). Clustering itself creates an additional administrative task for the case base: re-clusterization when adding new cases is computationally costly.

Tests employing different case-base sizes showed that the system improves performance over time with growing case-bases. Increases with the last two case-bases (IV and V) are not as important as in the beginning with case-bases I-III. This indicates that the chosen case-base size of 280 cases covers the space of possible solutions sufficiently for the range of load cases under study. Adding cases does not always lead to a monotonic decrease in the number of iterations. This is likely due to a weakness in the distance metric for selecting cases. More research is required in this area. Adding further cases will result in only small performance enhancements. Enhancements of the case-base regarding other load-cases including

horizontal loads as well as other control-objectives are thus possible without excessive case-base size.

6 Conclusions and future directions

6.1 Conclusions

6.1.1 Introduction

Advanced intelligent computational methods have been successfully applied to active structural control tasks. Task solution time decreased dramatically: First estimated to take centuries when applying a “brute-force” generate- and -test strategy, it decreased to hours when the space of solutions was sampled more intelligently by stochastic search and dropped in the end down to minutes when past experience was used through case-based reasoning.

Steps taken to build a computational framework for controlling active tensegrity structures in changing environments and control objectives were:

- Accuracy enhancement of the analytical model
- Comparing stochastic search algorithm for active structural control
- Creating a framework to increase structural performance over service life

Conclusions related to these three aspects are given in the following sections. Results are summarized in Section 6.1.5 and conclusions will be given in Section 6.1.6.

6.1.2 Accuracy enhancement of the analytical model

An analytical method (dynamic relaxation) is used for tensegrity structure analysis, since it solves simultaneously tasks of form-finding and structural analysis while considering geometrical non-linearity. Results of simulations have been compared with measured data. Differences observed are related to effects such as cable relaxation and nodal friction. The approach chosen for increases in accuracy differs from conventional methods. Whereas mechanical models are normally “tuned” to tasks by selecting, adapting and inserting additional mechanical parameters, a hybrid approach has been chosen. The computational model is left unchanged regarding number and type of parameters. It is nevertheless adapted as closely as possible to the real structure by fine-tuning present parameters. Neural networks then are used to decrease inaccuracies.

The network is trained by mapping deflections obtained by dynamic relaxation to deflections measured on the real structure. It has been shown to manage this task efficiently for two configurations of the laboratory structure as well as different mechanical characteristics. The structure was extended from three to five modules; electrical jacks, a new type of central node (Stage 3) and an enhanced version of the arc 264° type node have been installed.

Compared to conventional approaches, the hybrid system avoids over-parameterization and difficult tasks such as correct identification of values and unanticipated interdependencies.

Results obtained on the five-module structure lead to development of heuristics that are related to when the neural network should not be applied:

1. When only small deflections ($<1\text{mm}$) have to be corrected
2. When results of dynamic relaxation are almost equivalent to real structural behavior

Whereas the first case is easy to identify, the second is more difficult in practical situations.

Replacing dynamic relaxation completely with artificial neural networks was not successful at all. This coincides with good engineering practice: if explicit knowledge can be represented in a model, black-box methods, such as neural networks, should not be employed.

6.1.3 Comparing stochastic search algorithm for structural control

As a first technique, Fest(2002) shows that simulated annealing proposes successful control commands for the control of a tensegrity structure. These results were used as a basis for applying other stochastic search methods to the same task. Although multiple comparison of different search techniques on the same task exist, this test was necessary, since:

- A new technique (PGSL) has been integrated into the computational framework
- Testing of search techniques is not task-independent (no-free lunch theorem); other comparisons cannot be transferred to different tasks

All techniques provided good solutions. Nevertheless, convergence behavior differed. PGSL and genetic algorithms (GA's) showed much better convergence in the beginning phase of the process. While the number of objective function evaluations needed varied significantly with each PGSL run, GA's showed more consistent behavior. Both PGSL and GA's have advantages over simulated annealing (SA) since parameters to be adapted to the optimization task are fewer and more straightforward to determine.

PGSL is advantageous when rapid convergence over a small number of iterations can be attained. In almost every case tested, it outperformed SA and GA's over the first stage of iterations. GA's are robust and generally applicable optimization tools. It has to be acknowledged that PGSL needs more trials to attain the same level of maturity as GA's. Nevertheless, PGSL is more attractive for search in the neighborhood of good solutions.

6.1.4 Performance enhancement over service life

The computational framework developed in the context of this thesis can be transferred to other applications. This study takes a first step away from rigid systems that are bound to one control objective and cannot adapt over time to flexible systems that can increase performance over time and adapt therefore to changing environments.

The methodology chosen is a case-base reasoning system, where pairs of task-solution descriptions are stored and re-used to identify system commands close to new situations. Solutions have to be adapted since the probability to retrieve an exact match is low. For adaptation, search techniques tested in Chapter 4 have been employed. Tests on the three-

module prototype system in conjunction with simulated annealing revealed that it did not take advantage of being started close to a good solution. GA's and PGSL outperformed simulated annealing. In best cases, PGSL was 20 times faster than pure optimization.

The application of stochastic search techniques for case-adaptation avoids a problem outlined by Smyth and Keane (1998): the retrieval of a case that might not be suitable for adaptation. By definition, stochastic search should find a solution in any case. It has been observed, however, that some cases are faster to adapt than others.

Testing on five modules concentrated on comparing the most successful techniques discovered during testing the prototype for the three-module task. Performance increased when pure-optimization is compared with adaptation using PGSL with a starting interval limitation of 30%. No increases are observed when the same comparison is made for GA's. In the current version, they do not show any advantage from being started from a good solution. GA's performance during adaptation might be increased by seeding good solutions into their initial population.

Although with the five-module structure a bigger case-base was employed for testing, no maintenance problems occurred. K-means clustering has been proposed to avoid bottlenecks. Cases are sorted into sets of different clusters. During retrieval, only the similarity of cases in the cluster close to the current case are calculated. A value for the number of clusters (5) where retrieval time decreased significantly without affecting system competence has been found.

6.1.5 Summary of results

- A computational framework for active structural control has been developed. It consists of
 - A central database to assure efficacy and accuracy of data used
 - General tools for the analysis of tensegrity structures: generating structures employing IMAC's module, displaying a 3-D model of the generated system and performing a structural analysis.
 - A software module to search for good control commands that are governed by a pre-defined objective function and constraints, search techniques implemented are simulated annealing (SA), PGSL and genetic algorithm (GA).
 - A module which models the case-base reasoning process to re-use good past control commands and adapt them to the current situation. Performance is maintained by clustering stored cases.

6.1.6 List of conclusions

- Neural networks increase accuracy of the analysis when they are used to correct results from dynamic relaxation
- Replacing an analytical model (dynamic relaxation) by a neural network cannot be recommended.
- Employing stochastic search to find good control solutions for an actively controlled tensegrity structure provided good results regardless of the technique employed.
- Case-based reasoning increased the performance of search when good cases have been used as a starting point for search. This has been demonstrated for two different structural configurations. PGSL and GA have been identified as best techniques for this task. When started from good solutions PGSL adapts cases the most efficiently.
- Clustering of cases stored in the case-base showed potential to increase performance of retrieval while maintaining competence.
- With a growing case-base, system performance increases over service life in most cases.

6.2 Future directions

Work presented concentrated on building and testing an operational framework for active structural control. The objective was to investigate techniques for increasing structural performance over time. Increases in system performance might be possible.

Apart from enhancements, new research directions are discussed. Some of them have already been incorporated within a research proposal for the Swiss National Science Foundation. This proposal has been granted.

6.2.1 Study of the distance metric

Testing different case-base sizes on the same task showed that the system takes advantage from experience and improves its performance over time. However, cases have been observed where performance decreased significantly. For control-task 157_3948 (Figure 6-1), adapting a case retrieved from case-base III needed more iterations than pure optimization. This might be related to a weakness in the distance metric and requires further study.

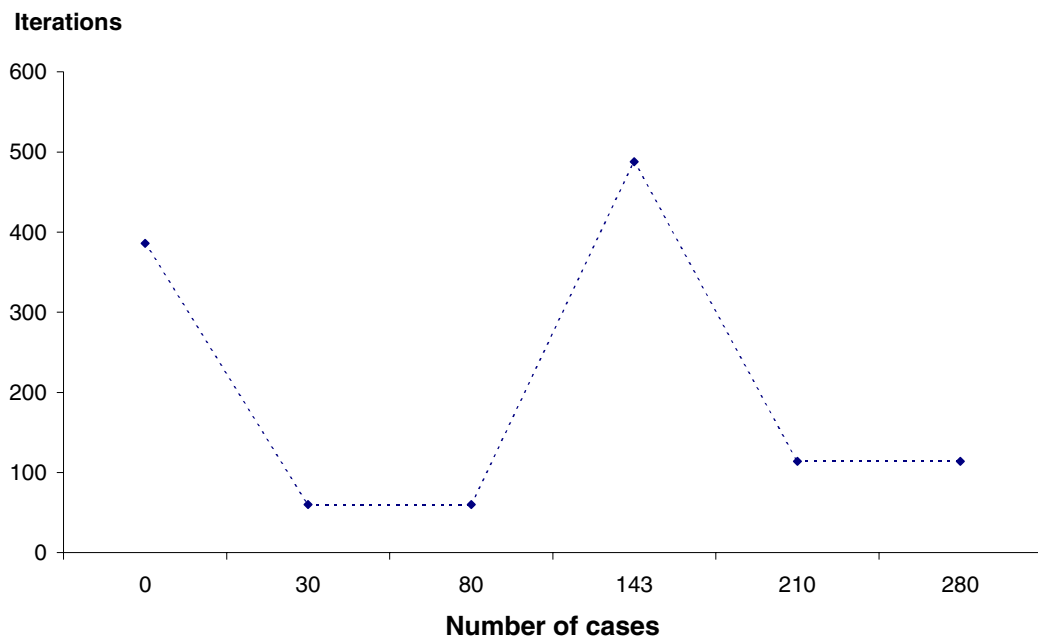


Figure 6-1 Decreases in performance for control-task 157_3948

6.2.2 System identification

Throughout testing, the control task to be completed was assumed to be well known: the magnitude and position of the load are important parameters needed during each calculation of the objective function, since dynamic relaxation is used. This assumption is only true for tests under laboratory conditions. Strategies should be developed to detect correct values under practical conditions when the structure is subjected to wind loading, etc.

6.2.3 Enhancing accuracy

The choice of training data for neural networks as well as the total number of patterns employed has an impact on result quality. Much research has shown that by covering the space of possible solutions evenly, accuracy increases.

Other types of neural networks than the one employed throughout this thesis should be tested to detect further possible advantages. Real data has to be verified to assure training towards a trustworthy system.

6.2.4 Increase adaptation performance

During testing of genetic algorithms for adaptation with the five-module structure, they did not show any advantage from being started close to a good solution. Seeding good solutions retrieved by the CBR-system into the initial population might provide the means to speed up convergence behavior.

Search spaces, such as the one observed for the five-module structure in conjunction with the constant slope control objective, might best be accommodated with heuristic instead of a stochastic search technique when cases are adapted. A hybrid approach might be used for cases when the heuristic fails to deliver a solution; stochastic search is still applicable in such situations.

6.2.5 Online training of an existing neural network

Online training would enable neural nets to be modified when environmental conditions change. Active tensegrity structures in practical situations could use such functionality. Online training consists of, first, adding new measurements taken during the service life of the structure to the training data and, second, retraining the network. Since computational time can be excessively long during service, the training time needed for the network becomes an important issue. The following issues are examples of important aspects:

- Evaluation of the quality of the training pattern proposed by the measurement system
- Deletion of old pattern sets without affecting accuracy

These aspects are similar to concepts of case-based reasoning and case maintenance (Smyth and Keane 1995).

First tests have been made with the most promising networks for the three-module structure: the one layer (3-8-3) configuration and the two-layer configuration (3-10-10-3). The error decreased by 1.9 % for the 3-8-3 configuration and by 0.55 % for the 3-10-10-3 configuration (Section 3.4.4). These decreases are not large enough to warrant such functionality at the time of testing. Further work is required to determine the characteristics of useful training sets.

6.2.6 Using a neural network during control command search

In most of the control tasks, search compensated slope deviation caused by loading almost to 100% (Fest 2002). For the double loading at nodes 41 and 50, however, results were not

convincing. Nodes 41 and 50 are not supported. Neither of these nodes is one of the nodes measured; they are on the perimeter of the structure.

Employing a neural network during control command search might compensate for these effects. Solutions found with and without artificial neural networks could be compared regarding quality before application to the structure.

6.2.7 Controlling space of possible solutions

Tests were carried out using a relatively small set of possible load scenarios. They do not cover the entire range of loads that might occur in real situations. Cases have to be carefully evaluated and selected to gain a better understanding of possible control tasks. Initial tasks could be evaluated by “pure” optimization and stored in the case-base to “train” the system for practical applications during a commissioning phase of the structure. Leake’s Problem distribution regularity is thus assured. This phase may be essential to take full advantage of the re-usability of tensegrity structures.

A landscape of solution quality could be generated by creating a cross table using each case to adapt another one. For the five-module task, such a table would consist out of 289 x 289 cells. Results could contribute to detecting which cases deliver good adaptation results and which do not.

6.2.8 Security mechanisms

Fest (2002) described a mechanical security system that avoids structural failure. Nevertheless, this does not prevent the control from encountering situations where a control command cannot be applied. This might happen when telescopic bars have reached the end of their stroke, etc. Strategies to avoid such problems should be provided and operate on two levels:

- Avoidance of such critical situations by adding additional constraints to the control objective function that is used during search.
- When the situation is unavoidable, provide means for reaching the control objective via intermediate, semi-optimal structural states.

6.2.9 Control mechanics of tensegrity systems

Greater knowledge of real structural behavior is required. An intensive study of the equilibrium matrix as presented under 2.1.3 would give significant insight into fundamental structural properties. This objective includes a theoretical and experimental study of control mechanics for non-linear tensegrity structures. For example, geometrically non-linear structures, such as tensegrity structures, are sensitive to the order of load application. Control moves are currently restricted within pre-defined magnitudes in order to mitigate difficulties associated with the order of their application. A detailed analysis of sequence effects would allow the application of larger control moves. Through consideration of sequence effects, better decisions related to the order of control commands could be made. This would speed up the time that is necessary to carry out a command set.

6.2.10 Multi-criteria control

This objective involves identifying control objectives in addition to roof-slope control. Examples of additional objectives are deflections and relative movements of cover attachments. There are also operational control objectives, such as flexibility, for future control movements, minimizing the energy needed to react, maximize distance from critical states and minimize number of bar movements. Pareto (1896) filtering shall be used to identify optimal surfaces and target tradeoff slopes will assist identification of control commands. This approach will be compared with other procedures such as ordering objectives and multi-step filtering. Commissioning procedures, to be executed just after erection will be examined for their ability to refine strategies.

6.2.11 Potential for fault-tolerance and self-repair

In contrast with other tensegrity designs, the structure at IMAC is capable of maintaining stability after failure of a member. Several questions thus arise. Can a fault tolerant system be developed? Can the structure detect failure of a single one of its parts, even if it is not fully equipped with sensors? Can it propose control movements for auto-repair, thereby increasing strength compared with the damaged structure?

6.3 Transferability of results

Algorithms and ideas presented are not exclusively tied to tensegrity structures. They can be transferred to many other applications. For example

Prestressed bridges

One limiting constraint of the amount of prestress introduced by the cable in the structure is the ultimate limit stress of concrete for compression when no additional charges counteract stress induced by the cable. Varying the stress can account for different situations and increase the load-bearing range.

Trim of sailing boats

The shape of sails is influenced, to a large extent, by the form of the mast. The mast is kept in shape by the forestay, the backstay and the shrouds. Tension in these cables can be regulated to form the sails such that they find the optimal position in relation to the wind direction for a range of situations.

References

- Adeli, H., and Saleh, A. (1999).** *Control, Optimization, and Smart Structures*, John Wiley & Sons Inc., New York.
- Adriaenssens, S., and Barnes, M. R. (2001).** "Tensegrity spline beams and grid shell structures." *Engineering Structures*, 23, 29-36.
- Anderberg, M. R. (1973).** "Nonhierarchical clustering methods." *Cluster analysis for application*, Academic Press, New York, 156-175.
- Anderson, D., Hines, E. L., Arthur, S. J., and Eiap, E. L. (1997).** "Application of artificial neural networks to the prediction of minor axis steel connections." *Computers & Structures*, 63(4), 685-692.
- Argyris, J. H., Angelopoulos, T., and Bichat, B. (1974).** "A General Method for the Shape Finding of Lightweight Tension Structures." *Computer Methods in Applied Mechanics and Engineering*, 3, 135-149.
- Arup. (2002).** "The millennium bridge." www.arup.com/millenniumbridge. Last accessed 12/2002.
- Averseng, J. Kazi-Aoual, M. N., Crosnier, B. (2002).** "Tensegrity systems selfstress state implementation methodology." *5th International Conference on Space Structures*, University of Surrey, Guildford, 31-38.
- Bach, K. (1975).** *Netze in Natur und Technik*, Karl Krämer Verlag, Stuttgart.
- Bach, K., Burkhardt, B., and Otto, F. (1988).** *Seifenblasen-Forming Bubbles*, Karl Krämer Verlag, Stuttgart.
- Barnes, M. R. (1977).** "Form Finding and Analysis of Tension Space Structures by Dynamic Relaxation," PhD-Thesis, The City University of London, London.
- Barnes, M. R. (1994).** "Form and stress engineering of tension structures." *Structural Engineering Review*, 6(3-4), 175-202.
- Barnes, M. R. (1999).** "Form Finding and Analysis of Tension Space Structures by Dynamic Relaxation." *International Journal of Space Structures*, 14(2), 89-104.
- Baumann, F. (2001).** "Développement d'un système de mesure pour évaluer la tension dans les câbles du Tensegrité." *Projet de laboratoire*, Swiss Federal Institute of Technology, ENAC-IS-IMAC, Lausanne.

- Bel Hadj Ali, N., Mangin, J.-C., and Cutting-Decelle, A.-F. (2002).** "Optimisation de la conception des structures metalliques avec les algorithmes genetiques." *3rd International Conference on Decision Making in Urban & Civil Engineering*, London, CD-ROM.
- Brown, A., Ankireddi, S., and Yang, H. T. Y. (1999).** "Actuator and Sensor Placement for Multiobjective Control of Structures." *Journal of Structural Engineering*, 127(2), 757-765.
- Bruno, R., Toomarian, N., and Salama, M. (1994).** "Shape estimation from incomplete measurements: a neural-net approach." *Smart Materials Structures*, 3(2), 92-97.
- Bubner, E. (1972).** "Zum Problem der Formfindung vorgespannter Seilnetzflächen," PhD-Thesis, Universität Stuttgart.
- Bullock, G. N., Denham, M. J., Parmee, I. C., and Wade, J. G. (1995).** "Developments in the use of the genetic algorithm in engineering design." *Design Studies*, 16, 507-524.
- Burkhardt, B. (1974).** *Gitterschalen, IL 10*, Karl Krämer Verlag, Stuttgart, Stuttgart.
- Cagan, J., Degentesh, D., and Yin, S. (1998).** "A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout." *Computer-Aided Design*, 30(10), 781-790.
- Calladine, C. R. (1978).** "Buckminster Fuller's "Tensegrity" Structures and Clerk Maxwell's Rules for the Construction of Stiff Frames." *Int. J. Solids Structures*, 14, 161-172.
- Carbonell, J. G. (1986).** "Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition." *Machine Learning: An Artificial Intelligence Approach* (Vol. 2), R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, eds., Morgan Kaufmann, Los Altos, CA, 371-392.
- Ceranic, B., Fryer, C., and Baines, R. W. (1999).** "An application of simulated annealing to the optimum design of reinforced concrete retaining structures." *Optimization and Control in Civil and Structural Engineering*, 47-53.
- Chang, C. C., Chang, T. Y. P., and Xu, Y. G. (2000).** "Adaptive neural networks for model updating of structures." *Smart Materials Structures*, 9(1), 59-68.
- Chattopadhyaya, A., and Seely, C. E. (1994).** "A simulated annealing technique for multiobjective optimization of intelligent structures." *Smart Materials Structures*, 3(2), 98-106.
- Chopra, A. (2001).** "Earthquake Dynamics of Base-Isolated Buildings." *Dynamics of Structures*, Prentice Hall, Upper Saddle River, NJ, 731-754.
- Chou, J.-H., and Ghaboussi, J. (2001).** "Genetic algorithm in structural damage detection." *Computers & Structures*, 79, 1335-1353.
- Coello Coello, C. (2001).** "A Short Tutorial on Evolutionary Multiobjective Optimization." *EMO 2001*, 21-40.

- Connor, A. M., and Shea, K. (2000).** "A comparison of semi-deterministic and stochastic search techniques." *Evolutionary Design and Manufacture, Selected Papers from ACDM '00*, University of Plymouth, UK, 287-298.
- Dallard, P., Fitzpatrick, A. J., Flint, A., Le Bourva, S., Low, A., Ridsdill Smith, R. M., and Willford, M. (2001).** "The London Millenium Footbridge." *The structural engineer*, 79(22), 17-33.
- De Jong, K., Arciszewski, T., and Vyas, H. (1999).** "An Overview of Evolutionary Computation and its Application to Engineering Design." *6th EG-SEA-AI workshop*, Wierzba, Poland, 9-21.
- Deb, K., and Gulati, S. (2001).** "Design of truss-structures for minimum weight using genetic algorithms." *Finite Elements in Analysis and Design*, 447-465.
- Djouadi, S., Motro, R., Pons, J. C., and Crosnier, B. (1998).** "Active Control of Tensegrity Structures." *Journal of Aerospace Engineering*, 37, 37-44.
- Dowsland, K. (1995).** "Simulated Annealing." Modern heuristic techniques for combinatorial problems, C. Reeves, ed., McGraw-Hill, London, 20-69.
- El-Beltagy, M. A., and Keane, A. J. (1999).** "A comparison of various optimization algorithms on a multilevel problem." *Engineering Applications of Artificial Intelligence*, 12, 639-654.
- Emmerich, D. G. (1964).** "Construction de réseaux autotendants.", Patent, Ministère de l'industrie, France.
- Emmerich, D. G. (1990).** "Composite Polyhedra." *International Journal of Space Structures*, 5(3&4), 281-296.
- Farsangi, H. E. (2002).** "Topological optimisation of double layer grids using a genetic algorithm." *Space Structures 5*, University of Surrey, Guildford, 459-468.
- Fenves, S. (1989).** "Expert Systems: Expectations versus Realities." *IABSE Colloquium, Report No. 58, International Association for Bridge and Structural Engineering*, 1-19.
- Fest, E. (1999).** "Construction et Modélisation des structures tensegrités.", FN-Report, École polytechnique fédérale de Lausanne, Lausanne.
- Fest, E. (2002).** "Tenségrité: une structure active.", PhD-Thesis, Swiss Federal Institute of Technology (EPFL), Lausanne.
- Fest, E., Shea, K., Domer, B., and Smith, I. F. C. (2003).** "Adjustable tensegrity structures.", accepted for publication in: *Journal of Structural Engineering*.
- Fleming, P. J., and Purshouse, R. C. (2001).** "Genetic Algorithms in Control Systems Engineering." Report 789, Department of Automatic Control and Systems Engineering, University of Sheffield.

- Flood, I., Nandy, S., and Muscynski, L (2000).** "Assessing External Reinforcement on RC Beams Using Neural Nets." *Computing in Civil and Building Engineering, ICCCBE-VIII*, Stanford, CA, 1114-1120.
- Fox, M. A. (2001).** "Beyond Kinetic." 12p., Kinetic Design Group, MIT.
- Francis, A. G. J., and Ram, A. (1993).** "The Utility Problem in Case-Based Reasoning." *Workshop AAAI 1993, Technical Report WS-93-01*, American Association of Artificial Intelligence, 160.
- Fuller, B. (1962).** "Tensile Integrity Structures." US Patent N° 3,063,521.
- Garrett, J. H. J., Gunaratnam, D. J., and Ivezić, N. (1997).** "Introduction." *Artificial Neural Networks for Civil Engineers: Fundamentals and Applications*, N. Kartam, et. Al., ed., ASCE, 1-18.
- Gerb. (2002).** "Passive damping". www.gerb.de, Last accessed 12/2002.
- Goldberg, D. E. (1989a).** *Genetic algorithms for search, optimization and machine learning*, Addison-Wesley, Reading, MA.
- Goldberg, D. E. (1989).** "Sizing Populations for Serial and Parallel Genetic Algorithms." *3rd International Conference on Genetic Algorithms, Morgan Kaufmann, Los Altos, CA*, 70-79.
- Gómez de Silva Garza, A., and Maher, M. L. (1999).** "An Evolutionary Approach to Case Adaptation." *ICCBR 99, LNCS 1650, Springer Verlag*, 162-172.
- Gossen, P. A., Chen, D., and Mikhlin, E. (2002).** "The First Rigidly Clad "Tensegrity" Type Dome, The Crown Coliseum, Fayetteville, North Carolina.", www.geigerengineers.com, Last accessed 12/2002.
- Grefenstette, J. J. (1986).** "Optimization of Control Parameters for Genetic Algorithms." *IEEE Transactions on Systems, Man and Cybernetics*, 16(1), 122-128.
- Guha, S., Rastogi, R., and Shim, K. (1998).** "CURE: An Efficient Clustering Algorithm for Large Databases." *ACM SIGMOD Conference*.
- Hajela, P., and Berke, L. (1991).** "Neurobiological computational models in structural analysis and design." *Computers & Structures*, 41(4), 657-667.
- Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001).** "On Clustering Validation Techniques." *Journal of Intelligent Information Systems*, 17(2/3), 107-145.
- Haykin, S. (1999).** *Neural Nets, A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ.
- Heister, F., and Wilke, W. (1998).** "An Architecture for Maintaining Case Based Reasoning Systems." *EWCBR 98, LNAI 1488, Springer Verlag*, 221-232.

- Heo, G., Wang, M. L., and Satpathi, D. (1997).** "Optimal transducer placement for health monitoring of long span bridge." *Soil Dynamics and Earthquake Engineering*, 16, 495-502.
- Housner, G. W., et al. (1997).** "Structural Control: Past, Presence and Future." *Journal of Engineering Mechanics*, 9(123).
- Hua, K., Faltings, B., and Smith, I. F. C. (1996).** "CADRE: case-based geometric design." *Artificial Intelligence in Engineering*, 10, 171-183.
- Hustin, S. (1988).** "Tim, A New Standard Cell Placement Program Based on the Simulated Annealing Algorithm," Master Thesis, University of California, Berkeley.
- Ingber, D. E. (1998).** "The Architecture of Life." *Scientific American January 1998*, 30-39.
- Jenkins, W. M. (1997).** "Approximate analysis of structural grillages using a neural network." *ICE, Structs. & Buildings 122*, 355-363.
- Jingsheng Shi, J. (2002).** "Clustering Technique for Evaluating and Validating Neural Network Performance." *Journal of Computing in Civil Engineering*, 16(2), 152-155.
- Kahla, N. B., Moussa, B., and Pons, J. C. (2000).** "Nonlinear dynamic analysis of tensegrity systems." *Journal of the International Association for Shell and Spacial Structures: IASS*, 41(132), 49-58.
- Kaveh, A., and Iranmanesh, A. (1998).** "Comparative Study of Backpropagation and Improved Counterpropagation Neural Nets in Structural Analysis and Optimization." *International Journal of Space Structures*, 13(4), 177-185.
- Kawaguchi, K., and Lu, Z. Y (2002).** "Construction of three-strut tension systems." *International Conference on Space Structures 5*, Guildford, 1-10.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, P. (1983).** "Optimization by Simulated Annealing." *Science*, 220(4598), 671-680.
- Kitagawa, Y., and Midorikawa, M. (1998).** "Seismic isolation and passive response-control buildings in Japan." *Smart Materials and Structures*, 7(5), 581-587.
- Kobori, T., Takahashi, M., Nasu, T., and Niwa, N. (1993).** "Seismic Response Controlled Structure With Active Variable Stiffness System." *Earthquake Engineering and Structural Dynamics*, 22, 925-941.
- Kolodner, J. (1993).** *Case-Based Reasoning*, Morgan Kaufmann, San Mateo, CA.
- Koumousis, V., and Georgiou, P. (1994).** "Genetic Algorithms in discrete optimization of steel truss roofs." *Journal of Computing in Civil Engineering*, 8(3), 309-325.
- Kumar, B., and Raphael, B. (2001).** *Derivational Analogy Based Structural Design*, Saxe Coburg, Stirling.
- Lalvani, H. (1996).** "Origins of Tensegrity: Views of Emmerich, Fuller and Snelson." *International Journal of Space Structures*, 11(1&2), 27-55.

- Leake, D. B. (2001).** "Artificial intelligence." Van Nostrand Scientific Encyclopedia, Wiley, New York.
- Leake, D. B., Smyth, B., Qiang, Y., and Wilson, D. C. (2001).** "Introduction to the Special Issue on Maintaining Case-Based Reasoning Systems." *Computational Intelligence*, 17(2), 193-195.
- Leake, D. B., and Wilson, D. C. (1998).** "Categorizing Case-Base Maintenance: Dimensions and Directions." *EWCBR-98, LNAI 1488, Springer Verlag*, 196-207.
- Leake, D. B., and Wilson, D. C. (1999).** "When Experience is Wrong: Examining CBR for changing Tasks and Environments." *ICCBR 99, LNCS 1650, Springer Verlag*, 218-232.
- Leake, D. B., and Wilson, D. C. (2000).** "Remembering Why to Remember: Performance-Guided Case-Base Maintenance." *EWCBR-2K, LNAI 1898, Springer Verlag*, 161-172.
- Legendre, P. (2001).** "Program K-means User's Guide." www.fas.umontreal.ca/bio/casgrain/en/labo/k-means.html. Last accessed 12/2002.
- Leite, J. P. B., and Topping, B. H. V. (1999).** "Parallel simulated annealing for structural optimization." *Computers and Structures*, 73(1-5), 545-564.
- Linkwitz, K., and Schek, H. J. (1971).** "Einige Bemerkungen zur Berechnung vorgespannter Seilnetzkonstruktionen." *Ing.-Arch.*, 40(145).
- Manoharan, S., and Shanmuganathan, S. (1999).** "A comparison of search mechanisms for structural optimization." *Computers & Structures*, 73(1-5), 363-372.
- McKenna, E., and Smyth, B. (1998).** "A Competence Model for Case-Based Reasoning." *9th Irish conference on artificial intelligence and cognitive sciences*, Dublin, Ireland.
- Miles, J., Moore, L., and Bradley, I. (1998).** "Customisable Knowledge Bases for Conceptual Design." *Artificial Intelligence in Structural Engineering, LNAI 1454, Springer Verlag*, 214-221.
- Motro, R. (1987).** "Tensegrity Systems for Double Layer Space Structures." *International conference on the design of non-conventional structures*, Edinburgh, Scotland, 43-51.
- Motro, R. (1992).** "Tensegrity Systems: The State of the Art." *International Journal of Space Structures*, 7(2), 75-83.
- Motro, R. (1997).** "Système de Tenségrité." *Actes du seminaire, Éditions de l'espérou, Montpellier*.
- Motro, R. (2002).** "Tensegrity: the state of the art." *Space Structures 5*, University of Surrey, Guildford, 97-106.
- Motro, R., and Raducanu, V (2001).** "Tensegrity Systems and Tensile Structures." *International Symposium on Theory, Design and Realization of Shell and Spatial Structures*, Nagoya, CD-Rom.

- Müller, K. R., Mika, S., Rätsch, G., K., T., and Shölkopf, B. (2001).** "An Introduction to Kernel-Based Learning Algorithms." *IEEE Transactions on Neural Networks*, 12(2), 181-201.
- Murakami, H. (2001a).** "Static and dynamic analysis of tensegrity structures. Part 1. Nonlinear equations of motion." *International Journal of Solids and Structures*, 38(20), 3599-3613.
- Murakami, H. (2001b).** "Static and dynamic analysis of tensegrity structures. Part II. Quasi-static analysis." *International Journal of Solids and Structures*, 38(20), 3615-3629.
- Nawrotzki, P. (2001).** "Passive Control For Buildings in Seismically Active Regions." *European Conference on Computational Mechanics*, Cracow, Poland.
- Nishimura, I., Yamada, T., Sakamoto, M., and Kobori, T. (1992).** "Active tuned mass damper." *Smart Materials Structures*, 1(4), 306-311.
- Nishimura, I., Yamada, T., Sakamoto, M., and Kobori, T. (1998).** "Control performance of active-passive composite tuned mass damper." *Smart Materials and Structures*, 7(5), 637-653.
- Olympiapark. (2002).** "Olympiapark München." www.olympiapark.de. Last accessed 12/2002.
- Oppenheim, I. J., and Williams, W. O. (2001).** "Vibration and Damping in Three-Bar Tensegrity Structure." *Journal of Aerospace Engineering*, 14(3), 85-91.
- Pankoke, U., et. Al. (1972).** *Wandelbare Dächer*, Karl Krämer Verlag, Stuttgart.
- Papadrakakis, M. (1981).** "A Method for the automatic evaluation of dynamic relaxation parameters." *Computer Methods in Applied Mechanics and Engineering*, 25, 35-48.
- Pareto, W. (1896).** *Cours d'economie politique, Vols. A and 2*, Rouge, Lausanne.
- Passera, and Pedretti. (2002).** www.ppeng.ch. Last accessed 12/2002.
- Pellegrino, S. (1990).** "Analysis of prestressed mechanisms." *International Journal of Solids and Structures*, 26(12), 1329-1350.
- Pellegrino, S. (1992).** "A Class of Tensegrity Domes." *International Journal of Space Structures*, 7(2), 127-142.
- Pellegrino, S., and Calladine, C. R. (1986).** "Matrix Analysis of Statically and Kinematically Indeterminate Frameworks." *International Journal of Solids and Structures*, 22(4), 409-428.
- Perelli, Y. (2000).** "Comportement des structures Tenségrités." EPFL, ENAC-IS-IMAC, Diploma Thesis, unpublished, Lausanne.
- Pfeiffer, R., and Scheier, C. (1999).** "Neural Networks for Adaptive Behavior." *Understanding intelligence*, MIT Press, Cambridge, MA, 139-177.

- Purvis, L., and Pu, P. (1995).** "Adaptation Using Constraint Satisfaction Techniques." *ICCBR-95, LNAI 1010, Springer Verlag, Sesimbra, Portugal*, 289-300.
- Rafiq, M. Y., Bugmann, G., and Easterbrook, D. J. (2000).** "Artificial neural networks to aid conceptual design." *The structural engineer*, 78(3), 25-32.
- Raphael, B., Robert-Nicoud, Y., Blanc, D., Dubey, C., Simonato, A., and Smith, I. F. C. (2001).** "Using cases and probabilistic analysis for monetary risk assessment." *Safety, Risk and Reliability - Trends in Engineering, International Conference of CIB, ECCS, fib, IABSE and RILEM, Malta*, 611-616.
- Raphael, B., and Smith, I. F. C. (2000).** "A probabilistic search algorithm for finding optimally directed solutions." *Construction Information Technology, Icelandic Building Research Institute, Reykjavik*, 708-721.
- Reeves, C. R. (1995).** "Genetic Algorithms." *Modern heuristic techniques for combinatorial problems*, C. Reeves, ed., McGraw-Hill, London, 20-69.
- Rehak, D. R., and Garrett, J. H. J. (1992).** "Neural Computing for Intelligent Structural Systems." *Intelligent Structures - 2: monitoring and control: proceedings of the International Workshop on Intelligent Systems, Perugia, Italy*, 147-161.
- Ricciardelli, F., Occhiuzzi, A., and Clemente, P. (2000).** "Semi-active Tuned Mass Damper control strategy for wind-excited structures." *Journal of Wind Engineering and Industrial Aerodynamics*, 88(1), 57-74.
- Robert-Nicoud, Y., Raphael, B., and Smith, I. F. C. (2000).** "Decision support through multiple models and probabilistic search." *Construction Information Technology 2000, Icelandic Building Research Institute*, 765-779.
- Rogers, J. L. (1994).** "Simulating structural analysis with neural networks." *Journal of Computing in Civil Engineering*, 8(2).
- Rojas, R. (1996).** *Neural networks - a systematic Introduction*, Springer Verlag, Heidelberg.
- Rossier, S. (1994).** "Optimization of Cable Structures Using Genetic Algorithms," Master Thesis, Herriot Watt University/EPFL, unpublished.
- Salajegheh, E., and Lotfi, K. (2002).** "Optimum design of space structures by combining genetic algorithms and simulated annealing using a response surface approximation." *Space Structures 5*, University of Surrey, Guildford, 487-497.
- Salama, M., Umland, J., Bruno, R., and Garba, J. (1993).** "Shape adjustment of precision truss structures: analytical and experimental validation." *Smart Materials and Structures*, 2(4), 240-248.
- Schaffer, J. D., Caruana, R. A., Eshelman, L. J., and Das, R. (1989).** "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization." *3rd International Conference on Genetic Algorithms, Morgan Kaufmann, Los Altos, CA*, 51-60.

- Schek, H. J. (1973).** "The Force Density Method for Form Finding and Computation of General Networks." *Computer Methods in Applied Mechanics and Engineering*, 3, 115-134.
- Shea, K., Fest, E., and Smith, I. F. C. (2002).** "Developing Intelligent Tensegrity Structures with Stochastic Search." *Advanced Engineering Informatics*, 16(1), 21-40.
- Shea, K., and Smith, I. F. C. (1998).** "Intelligent Structures: A New Direction in Structural Control." *Artificial intelligence in structural engineering, LNAI 1454, Springer Verlag*, 398-427.
- Shea, K., and Smith, I. F. C. 1999).** "Applying shape annealing to full-scale transmission tower re-design." *DETC99, 1999 ASME Design Engineering Technical Conference*, Las Vegas, NV.
- Shoureshi, R. (1995).** "Intelligent control and actuators for structures." *Smart Materials Structures*, 4(2), 132-139.
- Skelton, R. E., et al. (2000).** "An introduction to the mechanics of tensegrity structures." Handbook on mechanical systems design, Electrical Engineering CRC Press.
- Smith, I. F. C. (2002).** "Beyond Simulation." *IABSE Symposium Melbourne,, International Association for Bridge and Structural Engineering*, Reports 86, Melbourne, 32-34.
- Smith, I. F. C., and Shea, K. (1999).** "Extended Active Control to Build Intelligent Structures." *Structures for the Future - The Search for Quality, IABSE Symposium, Report No. 83, International Association for Bridge and Structural Engineering*, 1057-1064.
- Smyth, B., and Cunningham, P. (1996).** "The Utility Problem Analysed." *Advances in Case-Based Reasoning*, 392-399.
- Smyth, B., and Keane, M. T. (1995).** "Remembering To Forget." *14th International Joint Conference in Artificial Intelligence*, Montreal, Canada, 377-382.
- Smyth, B., and Keane, M. T. (1998).** "Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning." *Artificial Intelligence*, 2(102), 249-293.
- Smyth, B., and McKenna, E. (1998).** "Modelling the competence of case-bases." *ICCBR 98*, 208-220.
- Snelson, K. (2002).** "Letter to Rene Motro." www.grunch.net/snelson/motro.htm. Last accessed 12/2002.
- SNNS. (2001).** "Stuttgart Neural Net Simulator." www-ra.informatik.uni-tuebingen.de/SNNS/. Last accessed 12/2002.
- Spencer, B. F. J., and Sain, M. K. (1997).** "Controlling Buildings: A New Frontier in Feedback." *IEEE Control Systems Magazine on Emerging Technology*, 17(6, Special Issue), 19-35.

- Steelcon. (2002).** "Tuned Mass Damping Systems.", www.members.aol.com/Steeltherm/tuned.htm, Last accessed 12/2002.
- Sultan, C. (1999).** "Modeling, design, and control of tensegrity structures with applications," PhD-Thesis, Purdue University.
- Sultan, C., Corless, M., and Skelton, R. E. (2002).** "Linear dynamics of tensegrity structures." *Engineering Structures*, 24(6), 671-685.
- Svanerudh, P., Raphael, B., and Smith, I. F. C. (2002).** "Lowering Costs of Timber Shear-Wall Design using Global Search." *Engineering with Computers*, 18(2), 93-108.
- Swartz, W., and Sechen, C. (1990).** "New Algorithms for the Placement and Routing of Macro Cells." *IEEE Conference on Computer-Aided Design*, Santa Clara, CA, 336-339.
- Symans, M., and Constantinou, M. (1999).** "Semi-active control systems for seismic protection of structures: a start of the art review." *Engineering Structures*, 6(21), 469-487.
- Szilard, R. (1982).** *Finite Berechnungsmethoden der Strukturmechanik*, Ernst & Sohn Verlag, Berlin.
- Tensys. (2002).** "inTENS: Integrated Program Suite for Tension Structure Design." Tensys. www.tensys.com/inTENS-28-10-01.pdf. Last accessed 12/2002.
- Tibert, A. G. (2002).** "Deployable Tensegrity Structures for Space Applications," PhD-Thesis, Royal Institute of Technology, Stockholm.
- Tibert, A. G., and Pellegrino, S. (2001).** "Review of Form-Finding Methods for Tensegrity Structures." *submitted for publication*.
- U.S. Geological Survey. (2002).** "Earthquake Hazards Program - Northern California." quake.wr.usgs.gov/research/seismology/wg99/. Last accessed 12/2002.
- Underwood, P. (1983).** "Dynamic Relaxation." Computational methods for transient analysis, T. Belytschko, Hughes, T.J.R, ed., 246-265.
- Vassart, N., Laporte, R., and Motro, R. (2000).** "Determination of mechanism's order for kinematically and statically indetermined systems." *International Journal of Solids and Structures*, 37(28), 3807-3839.
- Wakefield, D. S. (1999).** "Engineering analysis of tension structures: theory and practice." *Engineering Structures*, 21(8), 680-690.
- Wall, M. (1996).** "GAlib: A C++ Library of Genetic Algorithm Components." lancet.mit.edu/ga/. Last accessed 12/2002.
- Wess, S., Althoff, K. D., and Derwand, G. (1993).** "Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning." *EWCBR93*, LNAI 1168, Springer Verlag, 167-181.

- Williamson, D., and Skelton, R. E. (1998).** "A General Class of Tensegrity Systems." *Engineering Mechanics for the 21st Century*, La Jolla, California.
- Wolpert, D. H., and Macready, W. G. (1997).** "No Free Lunch Theorems for Optimization." *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82.
- Zagar, Z., and Delic, D. (1993).** "Intelligent computer integrated structures: A new generation of structures." *Advanced Technologies*, 371-378.
- Zitzler, E. (1999).** "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," PhD-Thesis, Swiss Federal Institute of Technology, Zürich.

List of figures

<i>Figure 2-1 Nature's tensile structure. a spider web (Bach 1975).....</i>	<i>5</i>
<i>Figure 2-2 Munich Olympic stadium (Olympiapark 2002).....</i>	<i>6</i>
<i>Figure 2-3 Form finding with soap films (Bach et al. 1988)</i>	<i>7</i>
<i>Figure 2-4 Apparatus for the creation of minimal surfaces with soap films</i>	<i>7</i>
<i>Figure 2-5 Experimental model of the Munich stadium roof (Bach 1975).....</i>	<i>8</i>
<i>Figure 2-6 Measuring cable forces (Bach 1975).....</i>	<i>8</i>
<i>Figure 2-7 Measuring deflections with stereoscopic cameras (Bach 1975).....</i>	<i>9</i>
<i>Figure 2-8 Simplified scheme of dynamic relaxation.....</i>	<i>11</i>
<i>Figure 2-9 Maximum of kinetic energy in the equilibrium position of the structure</i>	<i>12</i>
<i>Figure 2-10 Kinetic damping.....</i>	<i>12</i>
<i>Figure 2-11 Needle tower II, Kroller Museum in Otterlo, Holland (Snelson 2002).....</i>	<i>13</i>
<i>Figure 2-12 Assembly of the equilibrium matrix (Pellegrino and Calladine 1986)</i>	<i>15</i>
<i>Figure 2-13 Classification of structural control.....</i>	<i>18</i>
<i>Figure 2-14 Map of seismic risks in California (U.S. Geological Survey 2002).....</i>	<i>19</i>
<i>Figure 2-15 Tuned mass dampers for chimneys (Steelcon 2002)</i>	<i>20</i>
<i>Figure 2-16 Hotel building in the Emirates (Gerb 2002)</i>	<i>20</i>
<i>Figure 2-17 Millennium bridge, London (Arup 2002).....</i>	<i>21</i>
<i>Figure 2-18 Four strategies for passive control</i>	<i>21</i>
<i>Figure 2-19 Spherical sliding bearing</i>	<i>22</i>
<i>Figure 2-20 Schematic of active structural control</i>	<i>22</i>
<i>Figure 2-21 Active control for the Kyobashe-Seiwa tower (Spencer and Sain 1997)</i>	<i>23</i>
<i>Figure 2-22 Variable stiffness device mechanism.....</i>	<i>23</i>
<i>Figure 2-23 Electrorheological damper</i>	<i>24</i>
<i>Figure 2-24 Schematic comparison of simulated annealing and a descent strategy.....</i>	<i>28</i>
<i>Figure 2-25 Example of accept rate and temperature schedules.....</i>	<i>29</i>
<i>Figure 2-26 Algorithm of the four nested loops of PGSL</i>	<i>31</i>
<i>Figure 2-27 Probability distribution at the beginning of the PGSL algorithm.....</i>	<i>32</i>

<i>Figure 2-28 Change of PDF in the PUC</i>	<i>32</i>
<i>Figure 2-29 Change of segmentation in the FC.....</i>	<i>32</i>
<i>Figure 2-30 Subdomain cycle</i>	<i>33</i>
<i>Figure 2-31 Basic scheme of genetic algorithms (Wall 1996).....</i>	<i>35</i>
<i>Figure 2-32 Crossover</i>	<i>36</i>
<i>Figure 2-33 Mutation.....</i>	<i>36</i>
<i>Figure 2-34 Principle elements of a neural network</i>	<i>39</i>
<i>Figure 2-35 Node of the hidden layer</i>	<i>40</i>
<i>Figure 2-36 Hypercube (as presented in (Rafiq et al. 2000)).....</i>	<i>42</i>
<i>Figure 2-37 Case-base size vs. efficiency and quality (Smyth and Cunningham 1996).....</i>	<i>46</i>
<i>Figure 2-38 K-means clustering algorithm.....</i>	<i>48</i>
<i>Figure 3-1 Tensegrity module.....</i>	<i>50</i>
<i>Figure 3-2 Modular nodal connector (Fest 2002).....</i>	<i>51</i>
<i>Figure 3-3 Open arc with rings (Fest 2002).....</i>	<i>51</i>
<i>Figure 3-4 Three stages of nodal development (Fest 2002)</i>	<i>51</i>
<i>Figure 3-5 Three module tensegrity structure, A, B and C are supports (Fest 2002).....</i>	<i>52</i>
<i>Figure 3-6 Telescopic bar (left: manual, right: electrical) (Fest 2002).....</i>	<i>52</i>
<i>Figure 3-7 Five module tensegrity structure (Fest 2002).....</i>	<i>53</i>
<i>Figure 3-8 Tension gauge, mechanical principle</i>	<i>54</i>
<i>Figure 3-9 Basic parameters of the tensegrity module</i>	<i>55</i>
<i>Figure 3-10 One module</i>	<i>55</i>
<i>Figure 3-11 System architecture of TSACS; DLL: Dynamic link library.....</i>	<i>57</i>
<i>Figure 3-12 Screenshot “Festorder”</i>	<i>58</i>
<i>Figure 3-13 Screenshot of link and node numbering.....</i>	<i>58</i>
<i>Figure 3-14 Screenshot of Tensgraph.....</i>	<i>59</i>
<i>Figure 3-15 Screenshot of Dynarex</i>	<i>59</i>
<i>Figure 3-16 Example for accuracy discrepancy (Fest 2002)</i>	<i>60</i>
<i>Figure 3-17 The control objective.....</i>	<i>61</i>
<i>Figure 3-18 Hybrid approach.....</i>	<i>62</i>
<i>Figure 3-19 Pattern sets for one load case</i>	<i>62</i>
<i>Figure 3-20 Node numbers of loaded joints, corresponding to Table 3-3.....</i>	<i>63</i>
<i>Figure 3-21 Accuracy loss for the 3-12-3 and the 3-14-14-3 topology</i>	<i>65</i>
<i>Figure 3-22 Accuracy enhancement for the 3-8-3 and the 3-10-10-3 topologies.....</i>	<i>67</i>
<i>Figure 3-23 Comparison of the two layer/one layer configuration.....</i>	<i>67</i>

Figure 3-24 Percentage deviations for the test patterns (sigmoid activation function)	69
Figure 3-25 Percentage deviations for the test patterns (tanh activation function).....	69
Figure 3-26 Five module structure with loaded nodes and controlled struts.....	70
Figure 3-27 Assembly of the loading	72
Figure 3-28 Improvement of accuracy.....	73
Figure 3-29 Increases in accuracy of the perturbed input patterns.....	74
Figure 3-30 Improvement of accuracy, obtained with corrected unseen patterns,	74
Figure 4-1 Schematic of active structural control	77
Figure 4-2 Control objective constant slope, nodes used for slope control.....	79
Figure 4-3 Multiple valid solutions of the control objective slope control.....	80
Figure 4-4 Extended basic scheme of structural control	81
Figure 4-5 Screen-shots of general settings.....	81
Figure 4-6 Screen shot “Optimiser”	82
Figure 4-7 Structural response as function of modified bars (Fest 2002).....	82
Figure 4-8 The seven basic load cases.....	83
Figure 4-9 Configuration for two telescopic struts per module.....	83
Figure 4-10 Configuration for five telescopic struts per module.....	84
Figure 4-11 Parameter study	86
Figure 4-12 Best-so-far curve load case 1	89
Figure 4-13 Best-so-far curve load case 2	89
Figure 4-14 Best-so-far curve load case 3	90
Figure 4-15 Best-so-far curve load case 4	90
Figure 4-16 Best-so-far curve load case 5	91
Figure 4-17 Best-so far curve load case 6	91
Figure 4-18 Best-so-far curve load case 7.....	92
Figure 4-19 Best-so-far curves for multiple runs, load case 2	93
Figure 4-20 Best-so-far curves for multiple runs, load case 6	93
Figure 5-1 Schematic of intelligent structural control.....	95
Figure 5-2 Active control components	96
Figure 5-3 Screenshot of the Retrieval module.....	97
Figure 5-4 Screenshot of the Adaptation module.....	98
Figure 5-5 Screenshot of the Application module (sequence check)	99
Figure 5-6 Screenshot of the Maintenance module.....	99
Figure 5-7 Three-module structure with loaded nodes; A, B and C are supports.....	100

<i>Figure 5-8 Distance metric for load case 1.1-776</i>	<i>104</i>
<i>Figure 5-9 Gradient behavior for load case 1.1-776, database III</i>	<i>106</i>
<i>Figure 5-10 Results obtained using different case bases; adapted control task 1.3-776</i>	<i>106</i>
<i>Figure 5-11 Gradient behavior for different search techniques and control-task 1.3-776 ..</i>	<i>107</i>
<i>Figure 5-12 Control objective.....</i>	<i>108</i>
<i>Figure 5-13 Five-module structure with loaded nodes and controlled struts</i>	<i>109</i>
<i>Figure 5-14 Control-task 391_48 adapted with results from control-task 350_48.....</i>	<i>114</i>
<i>Figure 5-15 Comparison of pure optimization (Optimiser) with adaptation</i>	<i>116</i>
<i>Figure 5-16 Comparison, 'pure' optimization and adaptation for control task 550_48.....</i>	<i>117</i>
<i>Figure 5-17 Improvement of performance for control-task 900_26.....</i>	<i>119</i>
<i>Figure 5-18 Improvement of performance for control-task 274_3948.....</i>	<i>120</i>
<i>Figure 5-19 Improvement of performance for control-task 624_3745</i>	<i>120</i>
<i>Figure 6-1 Decreases in performance for control-task 157_3948</i>	<i>127</i>

List of tables

<i>Table 3-1 Material properties of cables (Fest 2002)</i>	50
<i>Table 3-2 Material properties of bars (Fest 2002)</i>	50
<i>Table 3-3 Magnitudes and location of the loading</i>	63
<i>Table 3-4 Comparison of four network topologies</i>	65
<i>Table 3-5 Comparison of four network topologies trained with the average of data triples..</i>	66
<i>Table 3-6 Single node loading.....</i>	71
<i>Table 3-7 Loading at two nodes simultaneously.....</i>	71
<i>Table 4-1 Moore's law applied to the control problem</i>	78
<i>Table 4-2 Test cases for parameter adjustment of PGSL.....</i>	85
<i>Table 4-3 Parameters used with genetic algorithms.....</i>	86
<i>Table 4-4 Time needed to find the best solution for 2 moveable bars per module</i>	87
<i>Table 4-5 Costs and iterations for 5 moveable bars per module</i>	88
<i>Table 5-1 Kernel cases and new load cases.....</i>	101
<i>Table 5-2 Test case bases created.....</i>	101
<i>Table 5-3 Overview of cases, cases 1-7 are kernel cases</i>	102
<i>Table 5-4 Case base III, Threshold: 0.1.....</i>	107
<i>Table 5-5 Single node loading.....</i>	110
<i>Table 5-6 Loading on two nodes</i>	111
<i>Table 5-7 Parameters employed with PGSL</i>	113
<i>Table 5-8 Parameters employed with GA's</i>	113
<i>Table 5-9 Performance of adaptation using PGSL.....</i>	114
<i>Table 5-10 Performance of adaptation using a starting interval limited to 90%.....</i>	115
<i>Table 5-11 Comparison of pure optimization and PGSL, starting interval limitation 30%.</i>	116
<i>Table 5-12 C-H evaluated for different values of K on the reference case-base</i>	118
<i>Table 5-13 Exemplary results for different values of k.....</i>	118
<i>Table 5-14 Test case bases created.....</i>	119

Publications

2003

Domer, B., Fest, E., Lalit, V., Smith, I. F. C.: Combining the Dynamic Relaxation Method with Artificial Neural Networks to Enhance the Simulation of Tensegrity Structures, accepted for publication in: ASCE Journal of Structural Engineering.

Domer, B., Raphael, B., Shea, K., Smith, I. F. C.: Comparing two stochastic search techniques for structural control, accepted for publication in: ASCE Journal of Computing in Civil Engineering.

Fest, E., Shea, K., Domer, B., Smith, I. F. C.: Adjustable Tensegrity Structures, accepted for publication in: ASCE Journal of Structural Engineering

2002

Domer, B., Smith, I. F. C.: Performance Enhancement of Active Structures During Service Lives, 4th international PhD Symposium, Munich, Germany, Springer Verlag, 92-100.

Domer, B., Smith, I. F. C.: Simulating Tensegrity Systems with Dynamic Relaxation and Neural Networks, IABSE Symposium, Melbourne, Australia, Report 86, CD-Rom.

Domer, B., Smith, I. F. C.: Using Case-Based Reasoning to Build Intelligent Structures, Proceedings of the 9th International EG-ICE Workshop, Darmstadt, VDI Fortschrittsberichte 4(180), 102-111.

Fest, E., Domer, B., Smith, I. F. C.: Integrating Structure and Control Design - Towards Intelligent Tensegrity Structures, Proceedings of the 3rd World Conference of Structural Control, Como, Italy, John Wiley Ltd.

2000

Domer, B.: GallusPark II, Beton- und Stahlbetonbau, Heft 1/2000, 36-43.

Schnellenbach-Held, M. Domer, B.: Software-Agents: New Possibilities in Modelling Structural Systems, in: Proceedings of the Integrated Life-Cycle Design of Materials and Structures, Helsinki, Finland, 22.-24.5.2000.

1999

Domer, B., Schnellenbach-Held, M.: Objektkommunikation zur Unterstützung von Entwurfsprozessen, Forum Bauinformatik, Junge Wissenschaftler forschen, Darmstadt 1999, Fortschritt Berichte VDI Reihe 4 Nr. 156, VDI-Verlag Düsseldorf.

Domer, B., Schnellenbach-Held, M.: The Use of Object Communication for Optimized Design Processes, Proceedings of the 6th EG-SEA-AI Workshop, Wierzba, Poland.

1998

Domer, B., Schnellenbach-Held, M.: Computational Methods in structural engineering - state of the art, Darmstadt Concrete, Darmstadt 1998, Publikation des Institutes für Massivbau in Darmstadt, Eigenverlag.

Domer, B., Schnellenbach-Held, M.: Post strengthening of reinforced concrete structures with carbon fiber reinforced plastics, Darmstadt Concrete, Darmstadt 1998, Publikation des Institutes für Massivbau in Darmstadt, Eigenverlag.

Domer, B.: Galluspark II - Die bedarfsgerechte Sanierung eines Industriekomplexes aus der Jahrhundertwende, Seminarband "Kreative Ingenieurleistungen", Darmstadt 1998, Eigenverlag.

1995

Domer, B.: Integration vorhandener Software im Bereich der Tragwerksplanung - Lastabtrag, Forum Bauinformatik, Junge Wissenschaftler forschen, Hannover 1995, Fortschritt Berichte VDI Reihe 20 Nr. 173, VDI-Verlag Düsseldorf.

Grzeschkowitz, R., Domer, B., Schnellenbach, M.: Weiterentwicklungen im Bereich der Tragwerksplanung. Technische Nachrichten der Philipp Holzmann AG Nr. 2/95, Neu-Isenburg, 1995.

Curriculum Vitae

Bernd Domer

Les Porrades Sud

1184 Luins

Date of birth: 03.09.1967

Nationality: German

Academic qualifications

07/1992 Degree in Civil Engineering (Dipl.-Ing.), specialized in computer science

Professional Experience

École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

10/1999-
present Research Engineer and PhD Student

Main research areas: optimization algorithms, application of artificial intelligence techniques to civil engineering structures

Darmstadt University of Technology, Germany

11/1997-
09/1999 Research Assistant

Taught concrete engineering (dimensioning of structures) and finite elements for concrete engineering. Developed teaching material from scratch, supervised semester and diploma projects of students (3-4 students), tested and licensed a new cantilever fixation device, drew up expert reports on damaged structures

Philipp Holzmann AG, Frankfurt/Main, Germany

01/1996-
09/1997 Claim Manager

Handled contractual and budgetary issues for all non-contractual work demanded by the client across a building project.

11/1995-
12/1996 Construction Site Manager

Managed structural and façade renovation of a historical building

01/1994-
10/1995 Project Manager for structural engineering projects

Coordinated multiple structural engineering projects (office centres, residential buildings)

08/1992-
12/1994 Specialist for CAD and finite element systems

Prepared structural calculation with finite elements as a member of company's center of competence

Languages

German	Mother tongue
English, French	Fluent
Italian	Basic knowledge

Awards

12/1993 Stifterverband für die Deutsche Wissenschaft: for studies performed in an exemplary way regarding marks and time

09/2002 Best paper and presentation award (out of 131 presentations) at the 4th international PhD-symposium, Munich

