

Linköping Studies in Science and Technology. Dissertations No. 1444

# Optimal Design of Neuro-Mechanical Networks

Carl-Johan Thore



**Linköping University**  
**INSTITUTE OF TECHNOLOGY**

Division of Mechanics  
Department of Management and Engineering  
Linköping University, SE-581 83, Linköping, Sweden

Linköping 2012

Cover:

Title. The name of the author. Linköping University logotype.

Printed by:

LiU-Tryck, Linköping, Sweden

ISBN 978-91-7519-900-9

ISSN 0345-7524

Distributed by:

Linköping University

Department of Management and Engineering

SE-581 83, Sweden

© 2012 Carl-Johan Thore

No part of this publication may be reproduced, stored in a retrieval system, or be transmitted, in any form or by any means, electronic, mechanic, photocopying, recording, or otherwise, without prior permission of the author.

# Preface

The work presented in this thesis has been carried out at the Division of Mechanics at Linköpings Universitet, with partial financial support from the Swedish Research Council (VR).

I would like to thank my supervisor Prof. Anders Klarbring, without whom there would have been no thesis. My co-supervisors Prof. Matts Karlsson and Prof. Petter Krus should also be acknowledged. Thanks to everyone at the Division of Mechanics for good company. Special thanks to Dr. Jonas Stålhand, for your company, but also for introducing me to the field of mechanics.

I wish to express my gratitude to my parents, Johan and Ulla, and my brothers, Andreas and Fredrik, as well as to my other relatives, and my friends outside the world of mechanics. Last but not least, I thank my lovely wife Eva and my beautiful sons, August and Eric.

*Carl-Johan Thore*



# Abstract

Many biological and artificial systems are made up from similar, relatively simple elements that interact directly with their nearest neighbors. Despite the simplicity of the individual building blocks, systems of this type, network systems, often display complex behavior — an observation which has inspired disciplines such as artificial neural networks and modular robotics. Network systems have several attractive properties, including distributed functionality, which enables robustness, and the possibility to use the same elements in different configurations. The uniformity of the elements should also facilitate development of efficient methods for system design, or even self-reconfiguration. These properties make it interesting to investigate the idea of constructing mechatronic systems based on networks of simple elements.

This thesis concerns modeling and optimal design of a class of active mechanical network systems referred to as Neuro-Mechanical Networks (NMNs). To make matters concrete, a mathematical model that describes an actuated truss with an artificial recurrent neural network superimposed onto it is developed and used. A typical NMN is likely to consist of a substantial number of elements, making design of NMNs for various tasks a complex undertaking. For this reason, the use of numerical optimization methods in the design process is advocated. Application of such methods is exemplified in four appended papers that describe optimal design of NMNs which should take on static configurations or follow time-varying trajectories given certain input stimuli. The considered optimization problems are nonlinear, non-convex, and potentially large-scale, but numerical results indicate that useful designs can be obtained in practice.

The last paper in the thesis deals with a solution method for optimization problems with matrix inequality constraints. The method described was developed primarily for solving optimization problems stated in some of the other appended papers, but is also applicable to other problems in control theory and structural optimization.

This dissertation consists of an introduction and the following five papers:

- Paper I C-J Thore and A Klarbring, **Modeling and Optimal Design of Neuro-Mechanical Shape Memory Devices**, in Structural and Multidisciplinary Optimization, Volume 45 Issue 2, February 2012,
- Paper II C-J Thore, **Optimal Design of Neuro-Mechanical Oscillators**, Submitted
- Paper III C-J Thore, **Optimal Design of Neuro-Mechanical Oscillators with Stability Constraints**, In manuscript
- Paper IV C-J Thore and A Klarbring, **Some Aspects of Optimal Design of Neuro-Mechanical Shape Memory Devices**, In manuscript
- Paper V C-J Thore, **A Simple Method for Solving Nonlinear Non-convex Optimization Problems with Matrix Inequality Constraints with Applications in Structural Optimization**, Submitted

# Contents

<b>Preface</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Contents</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background and related work</b>	<b>2</b>
2.1 Active trusses . . . . .	5
<b>3 Theoretical background</b>	<b>8</b>
3.1 Nonlinear systems . . . . .	9
3.2 Optimal design . . . . .	11
<b>4 The mathematical model</b>	<b>16</b>
4.1 The mechanical subsystem . . . . .	16
4.2 The neural subsystem . . . . .	22
<b>5 Optimal design problems</b>	<b>24</b>
5.1 Neuro-Mechanical Shape Memory Devices . . . . .	25
5.2 Neuro-Mechanical Oscillators . . . . .	28
<b>6 Concluding remarks and future work</b>	<b>32</b>
<b>I An invariance principle</b>	<b>34</b>
<b>II A design-to-state mapping</b>	<b>34</b>
<b>Paper I: Modeling and Optimal Design of Neuro-Mechanical Shape Memory Devices</b>	<b>45</b>
<b>Paper II: Optimal Design of Neuro-Mechanical Oscillators</b>	<b>81</b>
<b>Paper III: Optimal Design of Neuro-Mechanical Oscillators with Stability Constraints</b>	<b>113</b>
	<i>vii</i>

<b>Paper IV:</b> Optimal Design of Neuro-Mechanical Oscillators with Stability Constraints	145
<b>Paper V:</b> A Simple Method for Solving Nonlinear Non-convex Optimization Problems with Matrix Inequality Constraints with Applications in Structural Optimization	163

# 1 Introduction

A noticeable trend in the development of mechanical artifacts during the 20th and the present centuries is the increasing level of integration of electronics into what had previously been purely mechanical systems. A car in the 1960s typically contained no advanced electronics save for the radio [18, Chapter 1], whereas nowadays a standard car contains multiple microprocessors that control fuel injection, airbags, anti-spin systems, and collision warning systems — and self-driving cars have been demonstrated successfully. In the late 1960s, an engineer working at a Japanese company coined the term "mechatronics" [73], combining parts from "mechanism" and "electronics", which describes the integration of electronics, mechanics, and information technology in a broad sense (including control theory and artificial intelligence), that characterizes many modern systems [18, 60], including cars.

Over the years, a number of concepts of mechatronic character have appeared, including Adaptronics [62], Intelligent Structures [26], Smart Structures [28], and Structronics [42]. Although this thesis does not deal with the subject explicitly, the field of robotics [5] should also be mentioned here. Today, mechatronic systems are almost omnipresent in large parts of the world, ranging in size from very large such as earthquake resistant buildings [120], to vehicles and robots, to very small in the form of micro-electro mechanical systems (MEMS) [125]. Based on the rapidly decreasing cost-per-function of computing devices [61], sensors and actuators, together with expected improvements in energy storage and energy harvesting [6], there is every reason to believe that mechatronic systems will continue to become increasingly important.

An interesting class of mechatronic systems are modular (self-reconfigurable) robots [63, 130, 119]. Such a robot constitutes an assembly of identical modules that interact locally with their nearest neighbors. This idea is captured by the notion of *network systems*. A characteristic attribute of network systems is that they may display complex behavior that could not have been inferred from studying individual elements; it is primarily the interaction between the elements that give rise to complexity — a phenomena known as *emergence*. Apart from modular robots, other man-made network systems include artificial neural networks [49, 25] and cellular automata [88], and linear network systems have been studied by researchers in control theory since at least the 1970s; see [29] and references therein. Although one could argue about to what extent biological systems fit the description of network systems [69], it is nevertheless clear that nature can serve to inspire the creation of such systems. From a mechanical point of view, the vertebrate heart, consisting of a large number of cells that collaborate to carry out a complicated mechanical motion, or the could be considered an interesting example.

This thesis focuses on what can be described as a type of biologically inspired mechatronic network systems, referred to as *Neuro-Mechanical Networks* (NMNs). The concept of NMNs was introduced by Krus and Karlsson [72], and subsequently

studied and developed by Sethson et al [110, 111]. Simulation and optimization of truss-like NMNs served to exemplify the ideas. More recently, Magnusson et al [83] considered the problem of designing so-called *active trusses* for adaptive stiffness in a small-deformation setting inspired by traditional structural optimization of passive structures. A prototype for an NMN was built by Magnus Sethson and is described in [82]. The present work is theoretical however, based on a mathematical model somewhat similar to the one used in [84, 82, 83], but extended to account for large deformations and other sources of nonlinear characteristics. Most importantly though, new examples of NMNs used in static as well as dynamic situations are treated.

The primary objectives of this study are:

1. To develop the concept of NMNs further and consider potential applications;
2. To establish proper mathematical models for NMNs;
3. To investigate the use of numerical optimization methods for design of NMNs.

It is hoped that the present work will be seen as a contribution towards a more long-term goal to enable automatic, optimal design of all kinds of mechatronic systems. As discussed in section 2.1, network systems like NMNs have properties that should facilitate the accomplishment of this goal, which if attained will mean dramatically reduced development times, and, in particular, the possibility of obtaining systems whose performance exceed what could have been achieved using traditional design methods.

This work has borrowed ideas from literature on both structural optimization and recurrent neural networks. These disciplines in turn are based on the theory of numerical optimization and nonlinear systems. Therefore, Section 3 constitutes a very brief overview of these subjects. This is followed by a derivation of the mathematical model used throughout this work in Section 4. Examples of optimal design problems based on this model and some numerical results are presented in Section 5. Concluding remarks and ideas for future work are then given in Section 6. First, however, we present some related work and discuss practical implementation of NMNs.

## 2 Background and related work

One way to define the term *structure* is to say that it is anything designed to carry a mechanical load [28]. In this thesis, the term *active structure* is used to describe any structure capable of generating deformation and/or reaction forces when subjected to non-mechanical stimuli (e.g. electrical or thermal). *Adaptive*, or *smart*, structures is then a subset of all active structures, namely those equipped with some form of sensors and feedback control. *Intelligent structures* finally, constitute a subset of the smart structures with higher cognitive abilities, e.g. task planning.

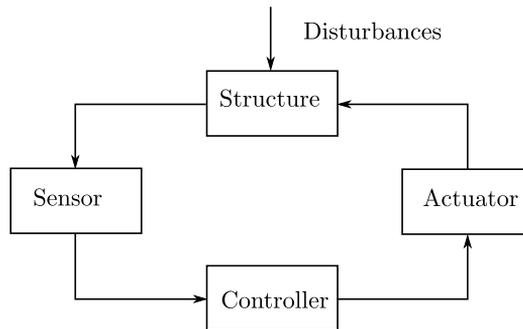
Although NMNs has here been positioned as a concept of mechatronic character, a more precise description of the present work is that it concerns optimization of active structures that do not undergo rigid body displacements, or where such displacements

---

are not of primary concern. This excludes active structures such as, e.g., vehicles and robotic arms, and limits the scope of what is considered to be related work.

Generally speaking, the motivation for considering active structures is either to compensate for inadequate passive performance, such as lack of stiffness in light-weight structures, or to realize functionality that could not have been achieved by a passive structure. In the former case, a caveat is of course that active structures may be more complicated than their passive counterparts, hence potentially more prone to errors and in need of additional maintenance.

As shown in Fig. 1, a smart structure comprises four building blocks: a "skeleton" in the form of structural material; actuators that constitute "muscle"; a "nervous system" in the form of sensors; and a "brain" in the form of electronics in which signal processing and control algorithms can be implemented. Following several authors,



**Figure 1:** The four building blocks of a smart structure.

e.g., Giurgiuti [38, 39], we have here noted the opportunity for biological inspiration when designing active systems. In passing it may also be remarked that biological systems have two very attractive properties: the ability to self-repair and the capability of self-replication [10]. Realization of these properties in non-biological mechanical systems is, however, outside the scope of this thesis.

Given the complexity of mechatronic systems and the successful application of numerical optimization methods for passive structures [14, 115], it is not surprising that interest in similar techniques for design of active structures is steadily increasing. Partly inspired by Frecker [37], we list five classes of optimization problems for design of active structures that have been treated in the literature:

1. actuator placement on given (passive) structures;
2. actuator placement and controller design (e.g. selection of feedback gains and actuator control voltages) for given structures;
3. design of coupling structure to enhance performance of a given actuator [1];
4. simultaneous design of structure (e.g. passive material distribution) and actuator placement;

5. simultaneous design of structure, actuator placement, and optimization of controller parameters (including sensor placement).

The word *simultaneous* may be noted, for it is possible to carry out the design of the structure first, followed by searching for a suitable placement of actuators, and finally optimizing control parameters. Such an approach however, as intuition says, generally leads to suboptimal solutions [34, 104]. For this reason, "simultaneous engineering" [60] is advocated.

The optimization problems treated in this thesis fall under the second and fifth of the categories listed above. Therefore, to give some additional background to the present work, a few selected papers dealing with problems from the listed classes will be referred.

Early work on design of active structures using optimization methods was concerned with placement of actuators on existing passive structures, one example being a paper from 1985 by Haftka and Adelman [43] wherein actuator placement on space antennas using analytical methods was described. The objective was to minimize the distortion of such antennae from their desired shapes. Subsequently, the actuator placement problem has been treated numerically [94, 37, 59], often modeled as a discrete problem and attacked by heuristic methods, such as genetic algorithms [48].

Methods from topology optimization [14] have the potential of making design of active structures more systematic. Examples of this are papers by Sigmund [113, 114] and Raulli and Maute [103] on topology optimization of electro-thermally actuated and electrostatically actuated, respectively, MEMS. These papers exemplify simultaneous design of structure and actuator placement (or more precisely, distribution of active material). Topology optimization of beams, plates, or shells bonded by layers of piezoelectric material has recently been treated in, e.g., [70, 30, 67, 66].

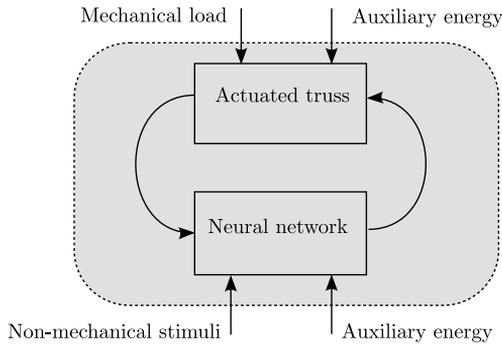
If an active structure is outfitted with sensors, control strategies for vibration control, or active damping, can be implemented [124, 102, 101, 3]<sup>1</sup>. (Note that, according to the definitions above, vibration control requires smart structures.) Beginning in the 1980s [15, 107], the problem of simultaneous optimization of structure and control system has been considered by several researchers. In principle, the idea is to select a controller, based on methods for linear quadratic [13, 12] or  $H_\infty$  [131, 58] control, which is optimal — with respect to the criteria of the respective method — for a large class of structures. Given an optimal controller one then poses an optimization problem to find, e.g., the optimal material distribution and actuator placement. Possible objectives can be to minimize the weight of the structure and the control effort, but also to maximize robustness and controllability/observability measures [12]. The controller in a problem of this type is parameterized by (a subset of) the optimization variables. Often, a *nested* strategy (see section 3.2.3) is used, meaning that one has to solve a set of algebraic (Ricatti) equations defining the controller for each candidate design [12].

---

<sup>1</sup>Sensors also enable structural health monitoring [53].

## 2.1 Active trusses

The type of NMNs dealt with here is referred to as *active trusses*, a term adopted from [83]. An active truss consists of two subsystems: a *mechanical* in the form of a truss in which the members are capable of actuation to actively change their length; and a *neural* in the form of a nonlinear, additive, artificial recurrent neural network (RNN) [55] that controls the actuation based on external stimuli and sensory information obtained from the mechanical subsystem. A high-level diagram of the two subsystems and their interaction with each other and the surroundings, through both mechanical and non-mechanical stimuli, is shown in Fig. 2. The arrows marked



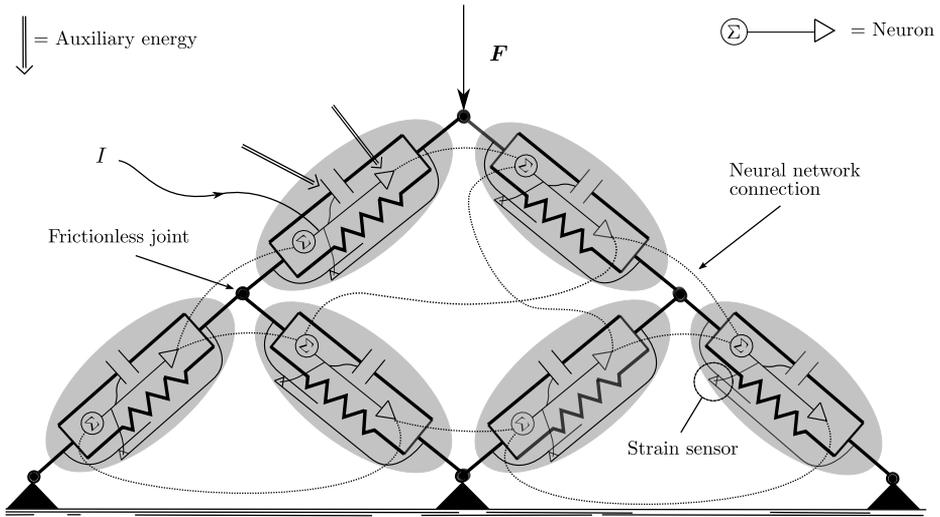
**Figure 2:** High-level diagram showing the two subsystems of an active truss and their interaction with each other and the surroundings.

”Auxiliary energy” indicate flow of energy required to maintain the active properties of the system — it could be electric, thermal, or other forms of non-mechanical energy.

Physically, the two subsystems that make up an active truss are tightly integrated as each neuron in the RNN is located within one element in the truss; i.e., it is not the case that the RNN is used as an external controller. A schematic picture of an active truss consisting of six elements is shown in Fig. 3. It is clearly seen that each element consists of an actuator and an artificial neuron, the latter of which should be thought of as a very simple dynamical system. The elements are connected mechanically at frictionless joints. Strain sensors convey information about the mechanical state to the neurons, which are directly connected to each other through neural network connections, thus forming an RNN.

As mentioned in the introduction, NMNs are examples of so-called network systems. Such systems have several potential advantages compared to more conventional systems. For instance:

- Since the functionality of the system is distributed across the elements, there is potential for graceful degradation, i.e. a slow decline in performance as an increasing number of individual elements fail rather than an immediate catastrophic failure due to malfunctioning of a central component.



**Figure 3:** An active truss consisting of six elements. A force  $F$  and a non-mechanical input stimulus  $I$  are applied.

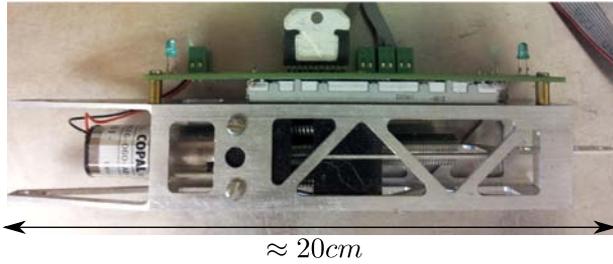
- The use of uniform elements facilitates development of efficient methods for design (or even self-reconfiguration) and implementation.
- If assembly and disassembly is straightforward (by careful design of connectors), multiple, possibly very different, systems could readily be constructed using the same physical building blocks (cf. LEGO®).

It should be noted that graceful degradation is not an inherent property of systems made up from simple building blocks, but has to be ensured by the designer. In the present work it is the second of the items listed above that is central, namely how to design network systems.

### 2.1.1 Practical implementation

Realization of practically useful NMNs, or NMN-like systems, may lie years ahead. Nevertheless, all kinds of practical implementations of NMNs are important to advance the concept and to test the validity of the mathematical models. Currently, there are no prototypes of NMNs up and running, although one was once built; see Fig. 4. The purpose of this section is to provide a very brief overview of some of the technologies available today, or in a not-to-distant future to realize practical implementations.

Starting with the neurons, it is clear that the necessary building blocks already exist, in the form of electronic components. Furthermore, these components are so small that their impact on the mechanical design will be negligible. Assuming a



**Figure 4:** An element of the prototype built by Magnus Sethson. Contraction or elongation is generated by a screw mechanism driven by an electric motor (seen on the left side). See [82] for details.

digital implementation is sought<sup>2</sup>, one could mention that the microcontroller ATtiny 88 [9], a readily available, cheap, product with more than enough functionality for implementing an artificial neuron, can be found in packages of size 4 by 4 millimeters (the silicon die itself is probably much smaller). It is also reasonable to assume that any sensors — we think here in particular of the strain sensors on the systems discussed in section 5.1 — can be made small. For example, thin-film technology has been used for strain gauges since the 1950s [128], and modern cellphones contain both accelerometers, gyroscopes, and pressure sensors in the form of MEMS. Thus, from a mechanical design point of view, the actuator is the most interesting part as it will constitute the bulk of the volume of an element and consume most of the supplied energy.

A *linear* actuator is a device that generates translatory motion. Some of the currently available linear actuator types together with important characteristics can be found in Tab. 1. In this work we have assumed that the elements are capable of producing active strains of up to 40 percent. Looking at Table 1, this seems to rule out the use of actuators based on, e.g., piezoelectric or magnetostrictive materials. Thus, if restricted to linear actuators, some kind of electroactive polymer, or, particularly for larger structures, hydraulic or pneumatic actuators appear preferable. Alternatively, mechanisms for converting rotary motion into translational, such as the one used in the element shown in Fig. 4, could be considered. There are also means to overcome the limitations of some of the actuator types listed in Tab. 1, an example of this is the arrangement of piezoelectric material to form traveling wave or inchworm motors [99].

The rapidly decreasing cost-per-function of microprocessors and other devices based on semiconductors means that nowadays, significant computing power can be integrated in individual components of a mechatronic system. Sensors and actuators, for example, can be made to appear from the outside in essentially any way desired; one may for instance want the displacement generated by an actuator to depend

<sup>2</sup>Modern-day computing is by far dominated by digital devices, but analog computation might become an important alternative in the future [79]. In section 4.2 an analog implementation of an artificial neuron is considered.

Type	Max. strain [%]	Max. stress [MPa]	Bandwidth [Hz]
Hydraulic	10-100	20-70	<300
Pneumatic	10-100	0.5-0.9	<300
Piezoelectric	0.1	1-35	$10^3$
Magnetostrictive	0.06-0.2	10-70	$> 10^3$
Shape Memory Alloy	0.7-7	100-700	0.2-7
Electroactive polymer	0.2-380	0.2-50	$1-10^3$
Muscle	20-40	0.1-0.4	$> 50$
Electro-static	$>10$	0.04	1

**Table 1:** Important characteristics for various actuator types (the numbers should be viewed primarily as order of magnitude estimates). Data is compiled from [56, 99, 60, 18, 81, 52]. Note that electroactive polymer actuators constitute a large family of different actuator types. The highest values of strain are observed in actuators based on dielectric elastomers, and the lowest in some based on ionic polymer metals [80].

linearly on the control voltage. This is achieved by actively compensating for performance defects such as, e.g., sensitivity to noise; cross-sensitivity, in particular with respect to temperature; and non-linear behavior, such as hysteresis, through feedback control [20]. For a designer of mechatronic systems this means that it is possible to work at a relatively high level of abstraction, treating individual components as subsystems/units with well-defined, uncomplicated, and predictable behavior.

One conclusion that can be drawn from the preceding discussion is that, physically, it is reasonable to view an active truss primarily as an actuator network — the actuators make up the bulk of volume and consume most of the supplied energy. Another thing to be noted is that while NMNs may seem to be a futuristic concept, to some extent, enabling technologies exist already today. With a bit of a stretch it could therefore be argued that the primary difficulty is to determine how the parts should be assembled, a problem which in this thesis is addressed by application of numerical optimization methods.

### 3 Theoretical background

In what follows, some background material on nonlinear systems and optimization is presented. Detailed accounts of these topics can be found in [68, 126, 76, 45, 92, 19, 4, 11]. A brief introduction to traditional structural optimization and some comments regarding the problem treated in Paper V are given at the end of the section.

### 3.1 Nonlinear systems

Many physical processes can be modeled as ordinary differential equations (ODEs) of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t), \quad (1)$$

where the  $t$  is time, a superposed dot denotes time-differentiation, and the function  $\mathbf{f} : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$  is nonlinear; in this thesis, the counterparts of  $\mathbf{f}$  are smooth in their first arguments and at least piecewise continuous in the second (time). The system (1) is a *non-autonomous* system as there is an explicit dependency on time in the right-hand side; otherwise the system would have been *autonomous*. Associated with (1) is the initial value problem

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, t) \\ \mathbf{x}(0) &= \mathbf{x}_0. \end{aligned} \quad (2)$$

It is well-known that this problem has a unique solution defined for all  $t \geq 0$  if  $\mathbf{f}$  satisfies a global Lipschitz condition<sup>1</sup>. However, as our models contain quadratic and cubic terms of the state (see Section 4), the Lipschitz property holds only locally. To prove global existence and uniqueness, one can therefore instead make use of, e.g., Theorem 3.3 in [68] which asserts that if every solution starting in a compact subset of  $\mathbb{R}^n$  remains in that set for all  $t \geq 0$ , then (2) has a unique solution defined for all  $t \geq 0$ . To ensure that this condition holds it suffices to show that every solution to (2) is bounded, i.e.,  $\|\mathbf{x}(t)\| < \infty$  for all  $t \geq 0$ .

The word "nonlinear" is important here — nonlinear systems are capable of much more complex behavior than their linear counterparts. For instance, a nonlinear system can have multiple isolated equilibrium points, exhibit finite escape behavior, and have stable limit cycles (self-excited periodic oscillations). To make things worse (or more interesting), it is often impossible to obtain analytical solutions. Fortunately, much can often be said about the qualitative behavior of nonlinear systems without actually solving them.

Some of the most important tools for qualitative studies of dynamical systems, concerning such things as stability of equilibria and boundedness of solutions, are so-called Lyapunov functions<sup>2</sup>. A continuously differentiable function  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  is said to be a Lyapunov function for (1) on an open set  $\mathcal{D} \subseteq \mathbb{R}^n$  if it has the property that, for all  $t \geq 0$ ,

$$\dot{V}(\mathbf{x}, t) = \nabla V(\mathbf{x})^\top \mathbf{f}(\mathbf{x}, t) \leq 0, \quad \forall \mathbf{x} \in \mathcal{D}, \quad (3)$$

where we have used the chain rule and substituted (1).  $\dot{V}$  would typically be referred to as something like "the time-derivative of  $V$  along solutions to (1)". By integration we obtain

$$V(\mathbf{x}(t)) - V(\mathbf{x}(0)) = \int_0^t \dot{V}(\mathbf{x}(\tau), \tau) \, d\tau \leq 0, \quad \forall \mathbf{x} \in \mathcal{D},$$

<sup>1</sup>That is,  $\|\mathbf{f}(\mathbf{x}, t) - \mathbf{f}(\mathbf{y}, t)\| \leq L\|\mathbf{x} - \mathbf{y}\|$  for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  and  $t \in [0, \infty)$ ,  $L$  being a Lipschitz constant.

<sup>2</sup>After the Russian mathematician AM Lyapunov, 1857-1918.

so clearly the evolution of the system on  $\mathcal{D}$  is such that  $V$  is non-increasing.

In order to obtain global results it is often useful to know that a function is *radially unbounded*. A continuous function  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  is said to be radially unbounded or, equivalently, *proper* [77, Proposition 2.17], if

$$\|\mathbf{x}\| \rightarrow \infty \Rightarrow V(\mathbf{x}) \rightarrow \infty, \quad (4)$$

where  $\|\cdot\|$  denotes a vector norm. The defining characteristic of a proper function, or *map*, is that the inverse image of a compact set under such a function is compact [77, p. 45]. This means that if  $V$  is radially unbounded, then the level sets  $\{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) \leq c\}$  are bounded for every  $0 < c < \infty$ <sup>3</sup>.

### 3.1.1 Dependency on parameters

As this thesis deals with optimal design of systems governed by ODEs it is interesting to see how solutions to (2) are affected by changes in the right-hand side. To this end, let  $\mathbf{f}$  be parameterized by a vector  $\mathbf{p}$  of time-invariant parameters. It turns out that if  $\mathbf{f}$  depends smoothly on  $\mathbf{p}$ , then so does the solution  $\mathbf{x} = \mathbf{x}(t, \mathbf{x}_0, \mathbf{p})$  [46, Corollary 4.1, p. 101].

When dealing with so-called *nested* optimization problems (see below) with many parameters and few constraints, the *adjoint method* [47] is best suited for sensitivity analysis. For an example, consider a functional of the form

$$H(\mathbf{x}(\mathbf{p})) = \int_0^T h(\mathbf{x}(t, \mathbf{p})) dt,$$

where  $h$  is a smooth function. It can be shown (see Paper II) that

$$\frac{\partial H}{\partial p_i} = \int_0^T \boldsymbol{\lambda}^\top \frac{\partial \mathbf{f}}{\partial p_i} dt,$$

where  $\boldsymbol{\lambda}$  solves the linear terminal value problem

$$\dot{\boldsymbol{\lambda}} = -\mathbf{J}^\top \boldsymbol{\lambda} + \nabla h \quad (5a)$$

$$\boldsymbol{\lambda}(T) = \mathbf{0}, \quad (5b)$$

with  $\mathbf{J} = \nabla_x \mathbf{f}$ . Assuming  $\mathbf{J}$  is bounded for all  $t \geq 0$  it follows from [68, Theorem 3.2] that (5) has a unique solution which does not escape to infinity for  $t < \infty$ .

## The static case

In many applications one is only interested in equilibria of (1) — often without reference to an underlying dynamical system. An equilibrium solution for (1) is a solution  $\mathbf{x}$  which satisfies

$$\mathbf{0} = \mathbf{f}(\mathbf{x}, \mathbf{p}, t), \quad \forall t, \quad (6)$$

---

<sup>3</sup>It is no restriction to assume, as is usually done, that  $V$  is non-negative; continuity and radial unboundedness ensures that it is bounded below, so a suitable constant can always be added to make it non-negative.

where we have again assumed a dependency of  $\mathbf{f}$  on some parameters. Provided there exists a point  $\mathbf{x}^*$  satisfying (6) for a given  $\mathbf{p} = \mathbf{p}^{*4}$ , it follows from the implicit function theorem [106, Theorem 9.28] that  $\mathbf{x}$  is a function of  $\mathbf{p}$  satisfying (6) in some vicinity of  $\mathbf{p}^*$  if the Jacobian  $\mathbf{J}$  is non-singular at  $(\mathbf{x}^*, \mathbf{p}^*)$ .

Given a point where the Jacobian is non-singular, the derivative of a function  $h = h(\mathbf{x}(\mathbf{p}))$  with respect to a parameter  $p_i$  can be computed using the adjoint method as follows:

$$\frac{\partial h}{\partial p_i} = \boldsymbol{\lambda}^\top \frac{\partial \mathbf{f}}{\partial p_i},$$

where  $\boldsymbol{\lambda}$  solves the linear system

$$\mathbf{J}^\top \boldsymbol{\lambda} = -\nabla h. \quad (7)$$

In the static case, the gradient of a function can generally be obtained at a small fraction of the cost of solving a nonlinear state problem (which is typically the most time-consuming part when solving a nested problem). In the dynamic case on the other hand, such a computation is comparatively expensive as it is necessary to solve an additional terminal value problem, cf. (5), and evaluate one integral for each variable  $p_i$  of interest.

## 3.2 Optimal design

### 3.2.1 General nonlinear non-convex optimization

The optimization problems encountered in this thesis can all be cast in the form

$$\begin{aligned} & \underset{\mathbf{p} \in \mathbb{R}^m}{\text{minimize}} && h(\mathbf{p}) \\ & \text{subject to} && \mathbf{l} \leq \mathbf{g}(\mathbf{p}) \leq \mathbf{u}, \end{aligned} \quad (8)$$

where  $h$  and  $\mathbf{g}$  are smooth non-linear and non-convex functions, referred to as the objective and constraint functions, respectively.  $\mathbf{l}$  and  $\mathbf{u}$  are constant vectors, and equality constraints are modeled by setting  $l_i = u_i$  for some  $i$ 's. Problem (8) is what we refer to as a standard nonlinear optimization problem (NLP). A *local solution* to this problem is a point in  $\mathbb{R}^m$  which satisfies the constraints of the problem and is locally optimal; i.e., there is a vicinity of this point where no smaller function value is attained. A *global solution* is any of the local solutions having the smallest objective function value. Existence of globally optimal solutions to (8) holds under the fairly non-restrictive conditions of Weierstrass' theorem [4, Theorem 4.7].

If the objective function in an NLP is *convex* and the constraints define a convex set, then this problem is said to be convex [19]. This situation is highly desirable since for a convex problem, every local solution is also a global solution [11, Theorem 3.4.2]. For this reason, convex problems are "easy" to solve. Unfortunately, the problems encountered in this thesis are non-convex, meaning that it can be difficult

<sup>4</sup>See [100, 105] for a sufficient condition.

to find globally optimal solutions. Systematic methods for finding global solutions exist (based on e.g. branch-and-bound procedures), but their application are deemed outside the scope of this work. For well-posed problems, the numerical optimization methods used here guarantee convergence to a local solution, or at least a so-called KKT point [92]. They are thus globally convergent *local* methods.

Apart from distinguishing between local and global methods, one way to classify optimization algorithms is to state whether they use derivative information or not (the latter category includes methods such as genetic algorithms and simulated annealing [48]). In our case, the optimization problems are smooth and obtaining, at least first-order, derivative information is relatively cheap in terms of CPU time compared to the cost of solving the state problems (see below). This makes derivative-based optimization methods a sensible choice.

### 3.2.2 Optimal design problems

In one sense, the optimal design problems considered here are just standard NLPs. However, there is some additional structure and notation associated with the term "optimal design". The variables are divided into two groups: *state variables* ( $\mathbf{x}$ ) and *design variables* ( $\mathbf{p}$ ), where the state variables should at some point satisfy a so-called *state problem*. The degree to which a candidate design meets the requirements (the "optimality") set up by a designer is quantified by *performance* or *cost* functions (or functionals) that are used as objective or constraints in the optimization problem together with constraints imposed directly on the design variables.

In the following, a *dynamic problem* is one where performance is quantified using a least one performance functional that depends on the entire solution to an initial value problem of the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{p}, t) \\ \mathbf{x}(0) &= \mathbf{x}_0.\end{aligned}\tag{9}$$

The state problem thus consists of finding a solution to (9) for a given  $\mathbf{p}$ . In a *static problem* on the other hand, performance is evaluated only at *points* that satisfy the algebraic equation

$$\mathbf{0} = \mathbf{f}(\mathbf{x}, \mathbf{p}).\tag{10}$$

In this case the state problem therefore consists in finding a solution to (10) for a given  $\mathbf{p}$ . As discussed in Paper IV, there are at least two ways in which this can be done: by applying a Newton-type solver to (10); or by solving a dynamic system whose equilibria coincide with solutions to (10).

### 3.2.3 Nested versus simultaneous

Regardless of whether the problem is static or dynamic, there are two commonly seen ways in which an optimal design problem can be formulated [7]: as a *simultaneous* problem where both the design and state variables are used as variables in the resulting NLP; or as a *nested* problem wherein the state variables are treated as functions of the design variables (by solution of the state problem) and therefore do not appear

explicitly in the NLP. For a given problem, the nested form is smaller in terms of the number of variables. However, it is hampered by a number of drawbacks, including the facts that

- in order to obtain useful sensitivity information it may be necessary to solve the state problem to a high level of accuracy for each set of design variables;
- the design-to-state map  $\mathbf{p} \rightarrow \mathbf{x}$  may be highly nonlinear (see Appendix II); and
- simple constraints, such as box constraints, involving the state variables become nonlinear and solution of additional "state problems", of the form (5) or (7), is required to obtain sensitivities.

These problems can be avoided by using a simultaneous formulation at the cost of increasing the number of variables and the possibility that intermediate designs are useless since they do not satisfy the equations of state.

### 3.2.4 Dynamic problems

In the dynamic case, the continuous-time, simultaneous optimization problem is written as

$$\begin{aligned} & \underset{\mathbf{x}(t) \in \mathbb{R}^n, \mathbf{p} \in \mathbb{R}^m}{\text{minimize}} && h(\mathbf{x}, \mathbf{p}) \\ & \text{subject to} && \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{p}, t), & t \in [0, T] \\ \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{l} \leq \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{u}, \end{cases} \end{aligned} \quad (11)$$

where  $h$  and  $\mathbf{g}$  are smooth non-linear and non-convex functionals. To treat this problem numerically, it must be converted into a standard NLP. In the direct transcription method of optimal control [17] this is done by dividing the interval  $[0, T]$  into a grid with  $N$  segments, introducing a set of variables  $\mathbf{x}_1, \dots, \mathbf{x}_{N+1}$  that approximates  $\mathbf{x}(t)$  at the grid points and using these to replace the ODE-constraint, and the performance functionals, by a discrete approximation (e.g. using the trapezoidal rule). Usually  $N$  must be quite large in order for the solution to (9) to be approximated with good accuracy, so the number of state variables need not be particularly large for (11) to become a large-scale problem.

To avoid a large number of variables one may consider a nested version of (11), namely

$$\begin{aligned} & \underset{\mathbf{p} \in \mathbb{R}^m}{\text{minimize}} && h(\mathbf{p}) \\ & \text{subject to} && \mathbf{l} \leq \mathbf{g}(\mathbf{p}) \leq \mathbf{u}, \end{aligned} \quad (12)$$

where we have used the shorthand notation  $h(\mathbf{p}) = h(\mathbf{x}(\mathbf{p}), \mathbf{p})$  and  $\mathbf{g}(\mathbf{p}) = \mathbf{g}(\mathbf{x}(\mathbf{p}), \mathbf{p})$ . Here  $\mathbf{x}(\mathbf{p})$  denotes the solution to the initial value problem (9) for a given  $\mathbf{p}$ . The number of variables in (12) is usually much smaller than in (11), but solving an initial value problem to high accuracy can be costly, and adding to that we also need to solve a terminal value problem, of the type (5), to obtain sensitivity information for each state dependent functional.

### 3.2.5 Static problems

For static problems, (11) reduces to

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n, \mathbf{p} \in \mathbb{R}^m}{\text{minimize}} && h(\mathbf{x}, \mathbf{p}) \\ & \text{subject to} && \begin{cases} \mathbf{0} = \mathbf{f}(\mathbf{x}, \mathbf{p}) \\ \mathbf{l} \leq \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{u}, \end{cases} \end{aligned} \quad (13)$$

where  $\mathbf{x}$  is now independent of time. Again, a nested version can be obtained:

$$\begin{aligned} & \underset{\mathbf{p} \in \mathbb{R}^m}{\text{minimize}} && h(\mathbf{p}) \\ & \text{subject to} && \mathbf{l} \leq \mathbf{g}(\mathbf{p}) \leq \mathbf{u}, \end{aligned} \quad (14)$$

where  $h(\mathbf{p}) = h(\mathbf{x}(\mathbf{p}), \mathbf{p})$  and  $\mathbf{g}(\mathbf{p}) = \mathbf{g}(\mathbf{x}(\mathbf{p}), \mathbf{p})$  and  $\mathbf{x}(\mathbf{p})$  satisfies (10). The difference between this problem and (12) is that in the latter, the objective and constraints depend on the entire solution to an initial value problem, whereas in (14) performance is evaluated only at a single point. This implies that a numerical solution to (10) can be sought using some variant of Newton's method, leading to what in Paper IV is referred to as a *static nested* problem.

As discussed in section 3.1.1, the map  $\mathbf{p} \rightarrow \mathbf{x}$  is well-defined if the Jacobian  $\nabla_{\mathbf{x}} \mathbf{f}$  is non-singular. For optimal design using large-displacement models of trusses, this requirement has been found sometimes to be quite restrictive by the author. The reason for this is that buckling occurs, leading to singular Jacobians and hence non-convergence of the Newton methods tested<sup>5</sup>. As a possible remedy for these difficulties, one may consider applying an ODE-solver to find an equilibrium point for (9)<sup>6</sup>; using the terminology of Paper IV this approach leads to a *dynamic nested* problem.

## Structural optimization

As an example of a class of static problems we consider optimal design of (passive) structures for maximum stiffness. The state problem is to find a displacement vector  $\mathbf{u} \in \mathbb{R}^n$  which satisfies the small deformation quasi-static equilibrium equation

$$\mathbf{K}(\boldsymbol{\xi})\mathbf{u} = \mathbf{F}, \quad (15)$$

where  $\mathbf{K}(\boldsymbol{\xi})$  is the (symmetric, positive semi-definite) stiffness matrix,  $\boldsymbol{\xi} \in \mathbb{R}^m$  is a vector of design variables, and  $\mathbf{F}$  collects external, dead loads. Equation (15) is valid for a large class of structures, including a-priori discrete structures such as trusses, as well as discretized (by some finite element method) continuum structures. The

<sup>5</sup>Note that singular Jacobians is a potential issue even in the simultaneous formulation (13). Some NLP-solvers, however, will take measures to handle this [92, p. 572] so that the user never sees such problem directly, but rather by observing slow convergence.

<sup>6</sup>Convergence to an equilibrium point holds for instance if  $\mathbf{f}$  defines a *gradient system* (and all equilibria are isolated), i.e.  $\mathbf{f}$  is the gradient of some scalar function [51, Chapter 9].

problem discussed in this section is thus quite general. To maximize the stiffness of a structure, one tries to minimize its compliance  $\mathbf{F}^\top \mathbf{u}$ , which can be interpreted as the displacement of the structure in the direction of the load [2]. Presently, the so-called *ground structure* approach<sup>7</sup> is used almost exclusively. In the case of truss optimization, a ground structure consists of a number of nodes distributed in the design domain (a 2- or 3-dimensional subset of space) connected by a set of potential bars. A "structural universe" is thus defined, consisting of all designs compatible with the given ground structure, and the task is to pick among those that design which best suits our need. Choosing the bar volumes as design variables in a ground structure with  $m$  potential elements, we now pose the following optimization problem for truss design:

$$\begin{aligned} & \underset{\boldsymbol{\xi} \in \mathbb{R}^m, \mathbf{u} \in \mathbb{R}^n}{\text{minimize}} && \mathbf{F}^\top \mathbf{u} \\ & \text{subject to} && \begin{cases} \mathbf{K}(\boldsymbol{\xi}) \mathbf{u} = \mathbf{F} \\ \sum_{i=1}^m \xi_i \leq V, \end{cases} \end{aligned} \quad (16)$$

where  $V$  is the maximum allowed volume. With appropriate support conditions,  $\xi_i > 0$  for all  $i$  ensures that the stiffness matrix is positive definite, so  $\mathbf{u}$  can be eliminated from (16) by solving the equilibrium equation (15) for each  $\boldsymbol{\xi}$ <sup>8</sup>. We thereby obtain the following nested version of (16):

$$\begin{aligned} & \underset{\boldsymbol{\xi} \in \mathbb{R}^m}{\text{minimize}} && \mathbf{F}^\top \mathbf{u}(\boldsymbol{\xi}) \\ & \text{subject to} && \begin{cases} \sum_{i=1}^m \xi_i \leq V \\ \epsilon \leq \xi_i, \quad i = 1, \dots, m, \end{cases} \end{aligned} \quad (17)$$

where  $\epsilon$  is a small positive number. Typically  $n \ll m$  so it might seem as if the nested formulation would offer no significant advantage over the simultaneous. However, since  $\mathbf{K}$  depends linearly on  $\boldsymbol{\xi}$ , it can be shown that (17) is in fact a convex problem [2, 118], while due to the equilibrium constraint, problem (16) is non-convex [24, p. 87]. This illustrates that there may sometimes be additional points to consider besides those listed in Section 3.2.3 when formulating an optimal design problem.

Optimal design of structures using numerical methods is a vast subject, with research being initiated as early as the 1960s [31]. While this work has borrowed ideas and terminology from this field, a more comprehensive review is outside the present scope; see [16, 2] and [14, Chapter 4] for reviews focused on optimization of trusses. We will, however, mention an interesting extension to the problem discussed in the preceding paragraph that was treated in Paper V.

<sup>7</sup>A concept apparently due to Dorn [31] in the early 1960s. Alternatives to the ground structure approach are growth methods [87], wherein new bars and nodes may be introduced during the optimization process. This idea appears so far not to have gained much interest, however.

<sup>8</sup>With special care it is possible to let elements vanish even in a nested formulation, see [21] (which deals with discretized continuum structures), but this is not a common approach.

When solving, say (17), it may turn out that the optimal design is unstable in the sense that the structure is prone to global buckling; i.e., buckling of the structure as a whole rather than of individual bars. It can be shown [71] that under the *linear buckling assumption*, the condition

$$\mathbf{K}(\boldsymbol{\xi}) + \mathbf{G}(\boldsymbol{\xi}, \mathbf{u}) \quad \text{positive semi-definite,} \quad (18)$$

where  $\mathbf{G}(\boldsymbol{\xi}, \mathbf{u})$  is the *geometric stiffness matrix*, is sufficient to guarantee a stable structure for loads  $\tau \mathbf{F}$  with  $\tau \in [0, 1)$ ,  $\mathbf{F}$  being the load in (17) and (16). Adding this as a constraint to (17), or (16), results in a problem which is not a standard NLP. However, as described in Paper V<sup>9</sup>, it is quite straight-forward to put the problem into standard form should one wish to. Unfortunately (18) constitutes a non-convex (due to the bilinearity of the geometric stiffness matrix) and large-scale constraint, and therefore the alternative use of multiple load cases to enforce stability will be preferable in many cases.

## 4 The mathematical model

When establishing a mathematical model for NMNs (or any physical system) there are some important choices that need to be made. First, the level of detail. For instance, there exists a vast body of work concerned with very detailed models of actuators, taking into account such things as friction, heat production, and flow of fluid and chemicals. In this work, however, we have opted for a simple generic actuator model in the form of a spring with variable spring constant depending on the state of the neuron in the same element. One rationale for this is that it is not clear what type of actuators would be used to implement NMNs. Another is that, as discussed in section 2.1.1, a given actuator can often be made to behave in a desired way through integration of sensors and feedback control. Second, using a large displacement (geometrically nonlinear) model rather than a small displacement (linear) model can be very important to accurately predict the behavior of mechanical systems; see, e.g., results in [113]. Since the type of NMNs considered in this thesis are intended for generation of large displacements, the mathematical model takes into account geometrically nonlinear effects.

The remainder of this section presents a detailed derivation of the mathematical model, a set of nonlinear ODEs, used throughout this thesis. The mechanical and neural subsystems are treated separately in section 4.1 and 4.2, respectively.

### 4.1 The mechanical subsystem

The state variables in the model of the mechanical subsystem are the nodal displacements and velocities, collected in  $\mathbf{u} \in \mathbb{R}^n$  and  $\dot{\mathbf{u}} \in \mathbb{R}^n$ , respectively. The number of

---

<sup>9</sup>In that paper, the weight is minimized subject to an upper bound on the compliance. An advantage of that formulation is that existence of solutions is trivial to guarantee since (18) can always be satisfied by adding more material.

mechanical degrees of freedom is  $n = dN - n_f$ , where  $d$  denotes the number of spatial dimensions,  $N$  is the number of nodes in the truss, and  $n_f$  is the number of fixed degrees of freedom. The  $N$  nodes in the truss are connected by  $n_{el}$  elements. The state of the neural subsystem is given by  $\mathbf{v} \in \mathbb{R}^{n_{el}}$ , and enter into the model of the mechanical subsystem through the so-called active forces defined below.

As a starting point in the modeling we assume that the total energy of the mechanical subsystem satisfies the so-called *dissipation inequality*<sup>1</sup>

$$\overline{K + \Psi} \leq P_{ext}, \quad (19)$$

where  $K = K(\dot{\mathbf{u}})$  is the kinetic energy,  $\Psi = \Psi(\mathbf{u})$  is the potential energy, and  $P_{ext}$  is the external power. (A minor note is that in a non-inertial frame, the kinetic energy may also depend on  $\mathbf{u}$ .) Integrating both sides of (19) with respect to time over some interval  $[t_1, t_2]$  gives the interpretation that the energy stored in the system at  $t_2$  is less than or equal to the sum of the energy stored at  $t_1$  and the energy supplied to the system during this interval. Governing equations for the system at hand are now obtained by specifying all quantities in (19) and ensuring that this inequality holds for all evolutions of the system.

The mechanical properties of the elements are uniform along their respective lengths, meaning that for the  $i$ :th element, the cross-sectional area  $A_i$ , Young's modulus  $E$  (assumed to be the same for all elements), and the strain and stress,  $\varepsilon_i = \varepsilon_i(\mathbf{u})$  and  $\sigma_i = \sigma_i(\mathbf{u})$ , respectively, do not vary throughout the element. Using the constitutive assumption  $\sigma_i(\mathbf{u}) = E\varepsilon_i(\mathbf{u})$ , the stored elastic energy in element  $i$  is given by

$$\Psi_i(\mathbf{u}) = \int_0^{l_i} A_i \sigma_i(\mathbf{u}) \varepsilon_i(\mathbf{u}) dl = \frac{1}{2} \mathcal{V}_i E \varepsilon_i(\mathbf{u})^2,$$

where the volume  $\mathcal{V}_i = A_i l_i$ ,  $l_i$  being the undeformed length. Summing over all elements in the truss and accounting for linear springs attached to some of the nodes we get

$$\Psi(\mathbf{u}) = \frac{1}{2} \sum_{i=1}^{n_{el}} \mathcal{V}_i E \varepsilon_i(\mathbf{u})^2 + \frac{1}{2} \mathbf{u}^\top \mathbf{K}_0 \mathbf{u}, \quad (20)$$

where the stiffness matrix of the linear springs,  $\mathbf{K}_0$ , is symmetric and positive semi-definite.

To find an expression for kinetic energy, let the nodal velocities of element  $i$  be denoted by  $\dot{\mathbf{u}}_{i1}$  and  $\dot{\mathbf{u}}_{i2}$ , respectively. Assuming that the mass of each element is concentrated at its end points (i.e. each element is treated as two point masses connected by a massless spring), the kinetic energy of the truss is then simply

$$\sum_{i=1}^{n_{el}} \left( \frac{1}{2} m_i \|\dot{\mathbf{u}}_{i1}\|^2 + \frac{1}{2} m_i \|\dot{\mathbf{u}}_{i2}\|^2 \right),$$

---

<sup>1</sup>Note that (19) suggests the total energy as natural Lyapunov function candidate for physical systems (the time derivative should be interpreted as the time derivative along system trajectories). Perhaps not surprisingly, inequalities of this type play an important role in control theory [32], where, following Willems [127], systems satisfying a dissipation inequality are often referred to as *dissipative*.

where  $m_i \geq 0$  denotes mass. From this we obtain

$$K(\dot{\mathbf{u}}) = \frac{1}{2} \dot{\mathbf{u}}^\top \mathbf{M} \dot{\mathbf{u}}, \quad (21)$$

where the mass matrix

$$\mathbf{M} = \sum_{i=1}^{n_{el}} m_i \mathbb{I}_i,$$

in which  $\mathbb{I}_i$  is a diagonal matrix with ones in the places corresponding to the components of  $\dot{\mathbf{u}}$  associated with element  $i$ .

Having specified the quantities on the left-hand side of (19), the remaining task is to define the external power. To this end, we first consider an external force acting along an element to change its length; this is the effect of a linear actuator. We refer to this force<sup>2</sup> as an *active force* and note that it is power-conjugate to the rate of change of strain in the element. The active force in element  $i$  is taken to be a function of the state  $v_i$  of the neuron in the same element, and is defined as

$$f_i^a(v_i) = \mathcal{V}_i \beta_i \hat{f}_i(v_i), \quad (22)$$

where the parameter  $\beta_i$  is referred to as the *actuator gain*, and  $\hat{f}_i$  is a function of sigmoidal character; i.e., bounded and monotone increasing. Scaling with the element volume in (22) ensures that elements with zero volume produce no actuation. In addition to the active forces we also have external, dead loads applied to the nodes; these are collected in  $\mathbf{F}$ . Defining  $\mathbf{f}^a(\mathbf{v}) = [f_i^a(v_i)]$ , the external power can now be written as

$$P_{ext} = \mathbf{F}^\top \dot{\mathbf{u}} - \mathbf{f}^a(\mathbf{v})^\top \overline{\dot{\boldsymbol{\varepsilon}}(\dot{\mathbf{u}})} = (\mathbf{F} - \nabla \boldsymbol{\varepsilon}(\mathbf{u})^\top \mathbf{f}^a(\mathbf{v}))^\top \dot{\mathbf{u}}, \quad (23)$$

where the minus sign in front of the active force term indicates that positive active forces are contractive.

Using the expressions for the kinetic energy and the external power given above, the dissipation inequality (19) becomes, with aid of the chain rule,

$$(\mathbf{M} \ddot{\mathbf{u}} + \nabla \Psi(\mathbf{u}) - \mathbf{F} + \nabla \boldsymbol{\varepsilon}(\mathbf{u})^\top \mathbf{f}^a(\mathbf{v}))^\top \dot{\mathbf{u}} \leq 0. \quad (24)$$

A sufficient condition that yields satisfaction of this inequality, for all  $\dot{\mathbf{u}}$ , is that the term inside the parenthesis equals the negative of the gradient of a convex function of  $\dot{\mathbf{u}}$ , say  $D = D(\dot{\mathbf{u}})$ , such that  $\nabla D(\mathbf{0}) = \mathbf{0}$ <sup>3</sup>. If this is the case, then (24) reduces to

$$\nabla D(\dot{\mathbf{u}})^\top \dot{\mathbf{u}} \geq 0,$$

where the term on the left-hand side, which has dimension power, is the *dissipation*. The function  $D$  is an example of a *dissipation potential* [89]. With the dissipation potential

$$D(\dot{\mathbf{u}}) = \frac{1}{2} \dot{\mathbf{u}}^\top \mathbf{A}^\top \dot{\mathbf{u}},$$

<sup>2</sup>Strictly speaking it is a *force-like* quantity as its dimension is length times force.

<sup>3</sup>A simple proof can be obtained using the fact that a differentiable function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex if and only if  $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x})$  for all  $\mathbf{x}$  and  $\mathbf{y}$  in  $\mathbb{R}^n$  [19, p. 69].

where the matrix  $\mathbf{A}$  is positive definite<sup>4</sup>, (24) yields the governing equation

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{A}\dot{\mathbf{u}} + \nabla\Psi(\mathbf{u}) + \nabla\varepsilon(\mathbf{u})^\top \mathbf{f}^a(\mathbf{v}) = \mathbf{F}, \quad (25)$$

which after substitution of (20) and (21) and a slight rearrangement becomes

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{A}\dot{\mathbf{u}} + \sum_{i=1}^{n_{el}} \left[ \gamma_i E \varepsilon_i(\mathbf{u}) + f_i^a(v_i) \right] \nabla\varepsilon_i(\mathbf{u}) + \mathbf{K}_0 \mathbf{u} = \mathbf{F}. \quad (26)$$

To complete the model, it remains to specify how the strains relate to the nodal displacements; this is done below.

**Remark.** Assuming a static situation and no active forces, (25) reduces to

$$\nabla\Psi(\mathbf{u}) = \mathbf{F}. \quad (27)$$

By the global inverse function theorem proved in [129] this problem has a unique solution for every  $\mathbf{F}$  if and only if  $\|\nabla\Psi(\mathbf{u})\| \rightarrow \infty$  as  $\|\mathbf{u}\| \rightarrow \infty$  and  $\nabla^2\Psi$  is positive definite. From this we conclude that  $\Psi$  must be strictly convex [19, Section 3.1.4.] in order for (27) to have a unique solution. Thus, in order not to preclude buckling,  $\Psi$  must *not* be strictly convex; see [86] for additional discussion.

## Kinematics

The NMNs encountered in this thesis are expected to undergo large deformations, so as a strain measure we have chosen the Green-Lagrange strain. For a truss element this means that

$$\varepsilon_i(\mathbf{u}) = \frac{\hat{l}_i(\mathbf{u})^2 - l_i^2}{2l_i^2}, \quad (28)$$

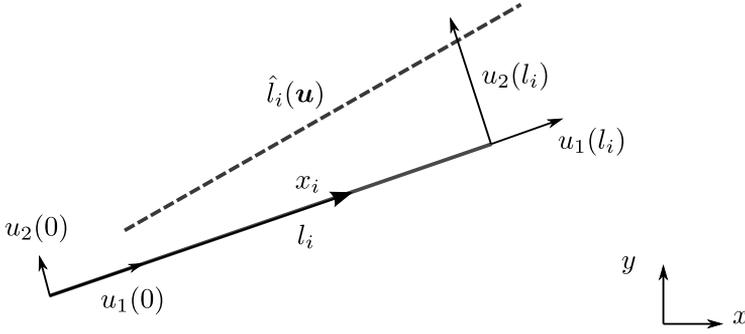
where  $l_i$  and  $\hat{l}_i(\mathbf{u})$  are the initial and current length, respectively, of the element shown in Fig. 5. To see that (28) really is the Green-Lagrange strain, let  $x_i \in [0, l_i]$  denote the position of a cross-section along element  $i$  in its initial configuration and notice that the component of the Green-Lagrange strain tensor in the axial direction of the undeformed bar is given by [27, p. 117]

$$\frac{\partial u_1}{\partial x_i} + \frac{1}{2} \left( \frac{\partial u_1}{\partial x_i} \right)^2 + \frac{1}{2} \left( \frac{\partial u_2}{\partial x_i} \right)^2, \quad (29)$$

where  $u_1$  and  $u_2$  are displacements along and orthogonal, respectively, to the bar in its initial configuration; see Fig. 5. Since the strain distribution in the bar is uniform, the derivatives in (29) are given by

$$\frac{\partial u_1}{\partial x_i} = \frac{u_1(l_i) - u_1(0)}{l_i}, \quad \text{and} \quad \frac{\partial u_2}{\partial x_i} = \frac{u_2(l_i) - u_2(0)}{l_i}. \quad (30)$$

<sup>4</sup>This makes  $D$  (strictly) convex, since  $\nabla^2 D(\dot{\mathbf{u}}) = \frac{1}{2}(\mathbf{A} + \mathbf{A}^\top)$  is positive definite.



**Figure 5:** Element  $i$  shown in its initial and current (dashed) configuration.

Pythagoras' theorem yields

$$\hat{l}_i(\mathbf{u})^2 = (l_i + u_1(l_i) - u_1(0))^2 + (u_2(l_i) - u_2(0))^2.$$

Dividing both sides by  $l_i^2$ , substituting (30), and rearranging leads to the conclusion that

$$\frac{\hat{l}_i(\mathbf{u})^2 - l_i^2}{2l_i^2} = \frac{\partial u_1}{\partial x_i} + \frac{1}{2} \left( \frac{\partial u_1}{\partial x_i} \right)^2 + \frac{1}{2} \left( \frac{\partial u_2}{\partial x_i} \right)^2,$$

showing that (28) is the Green-Lagrange strain.

In practice it is convenient to express the strain in terms of the vector  $\mathbf{u}$  represented in the global  $xy$ -system indicated in Fig. 5. As shown in Paper I, one such expression is

$$\varepsilon_i(\mathbf{u}) = \mathbf{b}_i^\top \mathbf{u} + \frac{1}{2} \mathbf{u}^\top \mathbf{B}_i \mathbf{u}, \quad (31)$$

where  $\mathbf{b}_i \in \mathbb{R}^n$  contains direction cosines of the element in its initial configuration and  $\mathbf{B}_i \in \mathbb{R}^{n \times n}$  is a symmetric, positive semi-definite matrix.

**Remark.** A model based on a linear elastic constitutive relation together with the strain measure defined in (28) can display curious behavior. Consider a bar of unit cross-sectional area with length  $l = 1$ , clamped at one end and constrained to move in the axial direction in the other; cf. Appendix II. Expressed in terms of the displacement  $u$ , the equilibrium equation is  $u^3 + 3u^2 + 2u = 2F/E$ . The interesting thing to note is that the left-hand side vanishes for  $u = -1$  — i.e., the internal force in a bar compressed to a point is zero! This unnatural behavior, which should not be confused with buckling, has not caused numerical difficulties in any of the problems treated herein. An exception was Paper IV, but there it could readily be tackled by adding box constraints on the displacements.

#### 4.1.1 A theoretical note on displacement control

In a quasi-static small deformation setting with no external forces, the equilibrium equation for an actuated truss is given by (cf. (46) and (15))

$$\mathbf{K} \mathbf{u} + \mathbf{B} \tilde{\mathbf{f}} = \mathbf{0}, \quad (32)$$

where  $\tilde{\mathbf{f}}$  is the control force (corresponding to  $\mathbf{f}^a(\mathbf{v})$  in (23)), and the  $i$ :th column of the matrix  $\mathbf{B}$  is  $\mathbf{b}_i$  found in (31). The question is how the control should be chosen if a certain displacement is desired? This can be formulated as an optimization problem:

$$\begin{aligned} & \underset{\tilde{\mathbf{f}} \in \mathbb{R}^m, \mathbf{u} \in \mathbb{R}^n}{\text{minimize}} && \|\mathbf{u} - \mathbf{u}^*\| \\ & \text{subject to} && \mathbf{K}\mathbf{u} + \mathbf{B}\tilde{\mathbf{f}} = \mathbf{0}, \end{aligned} \quad (33)$$

where  $\mathbf{u}$  and  $\mathbf{u}^*$  are the actual and desired displacements, respectively. Note that it is obviously unrealistic to assume that the control forces can be chosen arbitrarily as in (33), but the general point of this section does not depend crucially on this fact.

Now, if  $\mathbf{K}$  is positive definite, then  $\mathbf{u} = -\mathbf{K}^{-1}\mathbf{B}\tilde{\mathbf{f}}$  and (33) can be reduced to a problem in  $\tilde{\mathbf{f}}$  only:

$$\underset{\tilde{\mathbf{f}} \in \mathbb{R}^m}{\text{minimize}} \quad \|\mathbf{K}^{-1}\mathbf{B}\tilde{\mathbf{f}} - \mathbf{u}^*\|,$$

with solutions satisfying

$$\mathbf{B}\tilde{\mathbf{f}} = -\mathbf{K}\mathbf{u}^*. \quad (34)$$

This equation is solvable for every right-hand side if and only if the columns of  $\mathbf{B}$  span  $\mathbb{R}^n$  (this will typically be the case for a truss with many bars). Furthermore, if  $\mathbf{B}$  has full rank, then  $\tilde{\mathbf{f}}$  is uniquely determined by (34); otherwise  $\tilde{\mathbf{f}}$  is not unique and can be decomposed as

$$\tilde{\mathbf{f}} = \mathbf{f}_{\text{nil}} + \mathbf{f}_{\text{pot}},$$

where  $\mathbf{f}_{\text{nil}}$  lies in the nullspace of  $\mathbf{B}$  and causes stresses but no displacements (here "nil" and "pot" stands for nilpotent and impotent, respectively, a terminology borrowed from some recent papers on shape control [132, 93, 78]). In general,  $\mathbf{B}$  will not have full rank and therefore  $\tilde{\mathbf{f}}$  will not be unique. In the present work, however, the control force is prescribed to be a function of the neural state, hence unique. The important lesson from this discussion is that, for statically indeterminate structures, there is always a risk of wasting energy on actuation that does not contribute to completion of the task at hand.

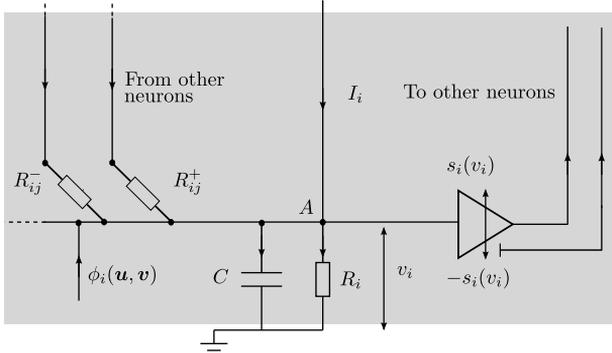
**Remark.** Having discussed the minimum compliance problem for passive structures at the end of section 3.2.5, it is interesting to see how  $\tilde{\mathbf{f}}$  should be chosen to minimize the compliance of an active structure. To this end, we consider the following problem:

$$\underset{\tilde{\mathbf{f}} \in \mathbb{R}^m}{\text{minimize}} \quad \left( \mathbf{F}^\top \mathbf{u}(\tilde{\mathbf{f}}) \right)^2,$$

where  $\mathbf{u}(\tilde{\mathbf{f}})$  solves (32) with  $\mathbf{F}$  on the right-hand side (the compliance of an active structure may actually become negative, hence the square). It is readily verified that the objective function is minimized for every  $\tilde{\mathbf{f}}$  satisfying  $\mathbf{B}\tilde{\mathbf{f}} = \mathbf{F}$ . Since the stiffness matrix can be written as  $\mathbf{K} = \mathbf{B}^\top \mathbf{D} \mathbf{B}$ , where  $\mathbf{D}$  is a diagonal matrix, and the strain as  $\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}$ , it is natural to set  $\tilde{\mathbf{f}} = \mathbf{D}\mathbf{B}\mathbf{u}_0$ , where  $\mathbf{u}_0$  solves  $\mathbf{K}\mathbf{u}_0 = \mathbf{F}$ . In practice this means, which should not come as a surprise, that each element in the truss should measure its strain and then produce an active force which is proportional to the measured value; the resulting compliance is thereby zero.

## 4.2 The neural subsystem

As already mentioned, the neural subsystem constitutes an artificial nonlinear recurrent<sup>5</sup> neural network. A network of this type can be implemented in a variety of ways, including as a software program running on some general purpose CPU; directly in analog [90] or digital [40] electronics hardware; or even in more exotic media [95]. To give some physical motivation for the mathematical model of the neural subsystem, we present a derivation of the governing equations for an implementation in analog electronics [55] illustrated in Fig. 6. The state variable for the depicted neuron is the voltage  $v_i$  across the capacitor.



**Figure 6:** Neuron  $i$  implemented using analog electronics. The output amplifier, indicated by a triangle, has two outputs: one normal, providing the voltage  $s_i(v_i)$ , and one inverted. The gray background alludes to the integration of the neuron inside one of the elements depicted in Fig. 3.

Although it is possible to write down a dissipation inequality for the neural subsystem resembling the one in section 4.1 and derive governing equations from it, this is unnecessarily complicated for the simple model used here. Instead, we note that Kirchhoff's current law applied to the node marked with an A in Fig. 6 gives

$$-C\dot{v}_i - \frac{v_i}{R_i} + I_i + \sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}^+} (s_j(v_j) - v_i) + \sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}^-} (-s_j(v_j) - v_i) + \phi_i(\mathbf{u}, \mathbf{v}) = 0, \quad (35)$$

where  $C$  denotes capacitance; and  $R_i$ ,  $R_{ij}^+$  and  $R_{ij}^-$  resistance. The index set  $\mathcal{N}_i$  specifies which of the  $n_{el}$  neurons in the network provide input to neuron  $i$ , and  $\phi_i(\mathbf{u}, \mathbf{v})$  is a current providing "sensory information"; see (43) below. The functions  $s_j$ ,  $j = 1, \dots, n_{el}$ , are taken to be of sigmoidal character. Following established terminology [49], the  $s_j$ 's are referred to as *activation functions*. Two things may be noted regarding (35): i) the output amplifier is assumed to have infinite input

<sup>5</sup>"Recurrent" implies that the neurons in the network are connected to each other in feedback loops. A more common architecture, particularly for machine learning applications, is the so-called *feed-forward* network [49].

impedance, hence there is no flow of charge through the amplifier; and ii) the implicit constitutive assumptions regarding the behavior of the resistors and the capacitor are suitable for small signals; cf. small deformations in mechanics.

Equation (35) can be expressed more succinctly as

$$\dot{v}_i = \tilde{\phi}_i(\mathbf{u}, \mathbf{v}) + \sum_{j \in \mathcal{N}_i} W_{ij} s_j(v_j) - \frac{1}{\tau_i} v_i + \tilde{I}_i, \quad (36)$$

where  $\tilde{\phi}_i(\mathbf{u}, \mathbf{v}) = \phi_i(\mathbf{u}, \mathbf{v})/C$ ,

$$W_{ij} = \frac{1}{C} \sum_{j \in \mathcal{N}_i} \left( \frac{1}{R_{ij}^+} - \frac{1}{R_{ij}^-} \right), \quad \frac{1}{\tau_i} = \frac{1}{C} \left( \frac{1}{R_i} + \sum_{j \in \mathcal{N}_i} \left( \frac{1}{R_{ij}^+} + \frac{1}{R_{ij}^-} \right) \right),$$

and  $\tilde{I}_i = I_i/C$ . Using matrix-vector notation, the governing equation for a network with  $n_{el}$  neurons of the type (36) can be written as

$$\dot{\mathbf{v}} = \boldsymbol{\phi}(\mathbf{u}, \mathbf{v}) + \mathbf{W} \mathbf{s}(\mathbf{v}) - \mathbf{C} \mathbf{v} + \mathbf{I}, \quad (37)$$

where  $\mathbf{C} = \text{diag}\{c_i\}$ , with  $c_i = 1/\tau_i$  for all  $i$ . The matrix  $\mathbf{W}$  is referred to as the *weight matrix* and defines the connectivity of the neural network and the weight of each neural network connection; cf. Fig. 3.

Omitting  $\boldsymbol{\phi}(\mathbf{u}, \mathbf{v})$  on the right-hand side and assuming  $\mathbf{W}$  is symmetric, (37) describes what is generally known as the Hopfield model [54, 55]<sup>6</sup>

$$\dot{\mathbf{v}} = \mathbf{W} \mathbf{s}(\mathbf{v}) - \mathbf{C} \mathbf{v} + \mathbf{I}, \quad (38)$$

originally envisioned to be used for so-called associative memories [54]. To study the behavior of (38) for constant  $\mathbf{I}$ , Hopfield used a Lyapunov function of the form

$$V(\mathbf{v}) = -\frac{1}{2} \mathbf{s}(\mathbf{v})^T \mathbf{W} \mathbf{s}(\mathbf{v}) + \sum_{i=1}^m c_i \int_0^{s_i(v_i)} s_i^{-1}(x) dx - \mathbf{s}(\mathbf{v})^T \mathbf{I}, \quad (39)$$

whose time derivative along solutions to (38) is given by

$$\dot{V} = -\dot{\mathbf{v}}^T \text{diag} \left\{ \frac{\partial s_i}{\partial v_i} \right\} \dot{\mathbf{v}} < 0, \quad \forall \dot{\mathbf{v}} \neq \mathbf{0},$$

where the inequality follows since each  $s_i$  is a monotone increasing function. Since  $V$  is radially unbounded, the invariance principle discussed in Appendix I applies, and it follows that every solution to (38) will converge to some equilibrium point, provided all equilibria are isolated. It turns out that in the presence of non-isolated equilibria, it is sufficient that the activation functions are analytic — the often used hyperbolic tangent being an example of such a function — to guarantee convergence to an equilibrium point [36].

<sup>6</sup>So called after John Hopfield, 1933- who was a pioneer in the study of RNNs. Some would argue that the term "Hopfield model" is inappropriate as a similar model was introduced in the 1960s but in a different field [41, p. 23].

Another conclusion that can be drawn from this discussion is that a non-symmetric weight matrix is a necessary condition for (38) to exhibit limit cycle behavior. As shown in Paper IV it is not sufficient, however.

**Remark.** Since the early 1980s, the Hopfield model and variants of it have received a great amount of interest. A comprehensive review would probably require hundreds of pages, but one can at least mention a few references covering the use of RNNs for solving combinatorial optimization problem [116]; networks with delays [35]; networks with non-symmetric weight matrices [117, 85], and algorithms for optimization — or in this context, "training" — of such networks [8]. The relation between artificial neural networks and their biological counterparts is covered in [65, 108], and spatially continuous approximations of RNNs are treated in [33].

## 5 Optimal design problems

This work introduces two new concepts based on the active truss model: *Neuro-Mechanical Shape Memory Devices* (NMSMDs), discussed Papers I and IV; and *Neuro-Mechanical Oscillators* (NMOs), treated in Papers II and III. The task of designing an NMSMD or an NMO is cast as an optimal design problem to be treated numerically. In this section we present versions of these problems that are slightly simplified compared to those given in the appended papers, together with some examples of the type of systems that can be obtained.

In all of the design problems, the neural network connection weights are used as design variables; this is quite natural since the neural subsystem constitutes the control system. In some of the problems stated below and in the appended papers, the actuator gains (see (22) above) and element volumes have also been subject to optimization. For notational convenience we introduce the set of feasible designs  $\mathcal{H} \subseteq \mathbb{R}^{n_{des}}$  defined by all constraints that do not depend on the state variables<sup>1</sup>.

The optimal design problems are based on a ground structure approach with ground structures of the type shown in Fig. 7. A subset of all possible neural network connections are shown as (green) curved lines in the figure, with an arrow indicating the direction of each connection. Also shown are support conditions, an applied force, a linear spring attached to one of the nodes, and non-mechanical input stimuli, drawn as curved (red) lines connected to the elements on the left side of the structure. Compared to those used in traditional structural optimization, the ground structures used herein also include neural network connections, so the number of design variables is potentially much larger. With no restriction on element lengths, a ground structure for an NMN with  $N$  nodes where overlapping elements are excluded will contain  $N(N+1)/2$  elements and  $n_w = n_{el}^2$  neural network connections, provided each neuron has direct connection to all other neurons in the system. Although in most cases the neural subsystem will probably not be fully connected,  $n_w$  will typically be substantially greater than  $n_{el}$ .

---

<sup>1</sup> $\mathcal{H}$  is assumed to be such that a particular design variable is restricted to a compact set whenever needed to ensure a well-posed problem.

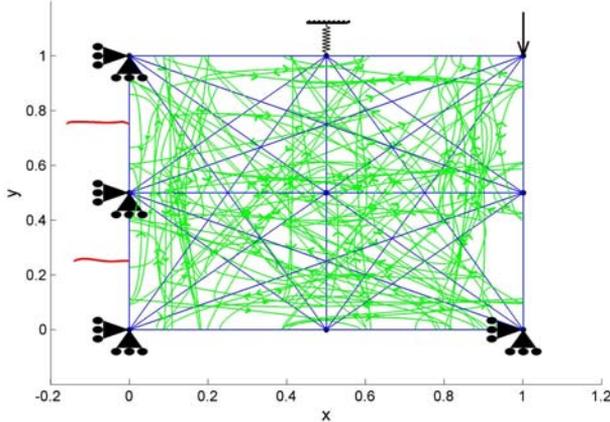


Figure 7: A ground structure for design of an NMO or NMSMD.

## 5.1 Neuro-Mechanical Shape Memory Devices

NMSMDs are designed to take on static configurations, *shapes*, in response to constant input signals. To this end, they are, as in Fig. 2, equipped with strain sensors. This allows them to sense applied loads through induced strain. By means of actuation, an NMSMD can thus by itself take active measures to, for instance, amplify or invert the response of a corresponding passive structure (which is to "give in" to the applied load).

A "shape" is characterized by target values for  $n_c$  of the displacement components associated with nodes in the system. The difference between the actual and desired shape can thus be quantified using the functions  $h_i$ ,  $i = 1, \dots, n_c$ , defined by

$$h_i(\mathbf{u}) = (\mathbf{c}_i^\top \mathbf{u} - (u_{tar})_i)^2, \quad i = 1, \dots, n_c,$$

where  $\mathbf{c}_i^\top \mathbf{u}$  and  $(u_{tar})_i$ , is the actual and desired value, respectively, for the  $i$ :th displacement component. Here,  $\mathbf{c}_i$  is a vector of zeros except for a one in the  $i$ :th position. An optimization problem to find a design such that the maximum value attained by any of the  $h_i$ :s for a given load is minimized subject to a constraint on the actuation energy can now be defined as follows:

$$\begin{aligned} & \text{minimize}_{\mathbf{p} \in \mathcal{H}} \left\{ \max_{i=1, \dots, n_c} h_i(\mathbf{u}(\mathbf{p})) \right\} + \gamma \Phi(\boldsymbol{\omega}) \\ & \text{subject to } \|\mathbf{f}^a(\mathbf{v}(\mathbf{p}))\|^2 \leq E_{max}, \end{aligned} \quad (40)$$

where  $\gamma > 0$ , the positive function  $\Phi$  is used to penalize non-zero neural network weights, and  $E_{max}$  is an upper bound on the size of the active force vector. The max-function is non-differentiable at the origin, but a standard trick turns (40) into

a standard NLP; see Paper I. The state variables are assumed to satisfy the following state problem:

Given  $\mathbf{F}$ ,  $\mathbf{I}$  and  $\mathbf{p}$ , find  $\mathbf{u}$  and  $\mathbf{v}$  such that

$$\sum_{i=1}^{n_{el}} \left[ \gamma_i E \varepsilon_i(\mathbf{u}) + f_i^a(v_i) \right] \nabla \varepsilon_i(\mathbf{u}) + \mathbf{K}_0 \mathbf{u} = \mathbf{F} \quad (41a)$$

$$\mathbf{0} = \boldsymbol{\phi}(\mathbf{u}, \mathbf{v}) + \mathbf{W} \mathbf{s}(\mathbf{v}) - \mathbf{C} \mathbf{v} + \mathbf{I}. \quad (41b)$$

Since (41) is a set of nonlinear algebraic equations, it is natural to seek numerical solutions using Newton's method or some variant thereof<sup>2</sup>. This, however, turned out to be difficult — too frequently the solvers tested would not converge to points where the residual vanished. Therefore, we chose instead to solve the dynamic system

$$\mathbf{A} \dot{\mathbf{u}} + \sum_{i=1}^{n_{el}} \left[ \gamma_i E \varepsilon_i(\mathbf{u}) + f_i^a(v_i) \right] \nabla \varepsilon_i(\mathbf{u}) + \mathbf{K}_0 \mathbf{u} = \mathbf{F} \quad (42a)$$

$$\dot{\mathbf{v}} = \boldsymbol{\phi}(\mathbf{u}, \mathbf{v}) + \mathbf{W} \mathbf{s}(\mathbf{v}) - \mathbf{C} \mathbf{v} + \mathbf{I}, \quad (42b)$$

where  $\mathbf{A}$  is positive definite, whose equilibria coincide with solutions to (41). In Paper I, to ensure that solutions to (42) actually converged to equilibrium points, the weight matrix was required to be symmetric and the term  $\boldsymbol{\phi}(\mathbf{u}, \mathbf{v})$  in (42b), referred to as the *sensor term*, was defined as

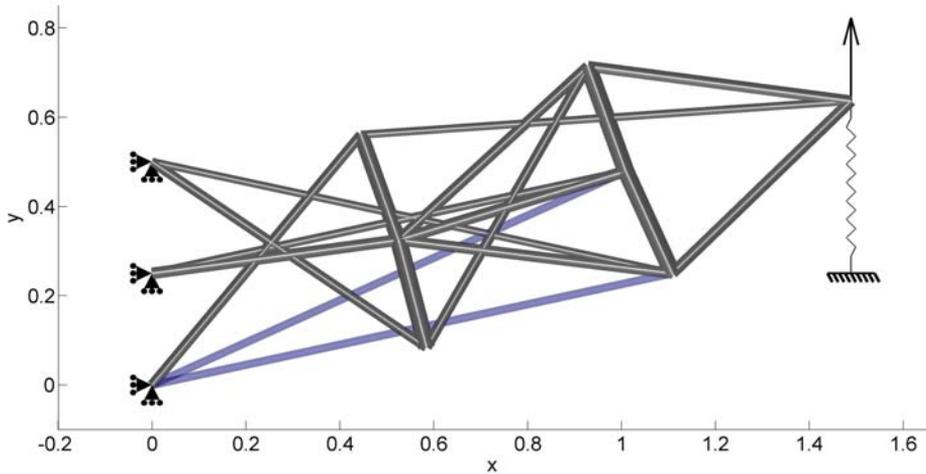
$$\boldsymbol{\phi}(\mathbf{u}, \mathbf{v}) = -\text{diag} \{ \beta_i \} \boldsymbol{\varepsilon}(\mathbf{u}). \quad (43)$$

This choice resulted in the existence of a Lyapunov function for (42) such that the invariance principle in Appendix I could be applied to show convergence. It may be noted from (43) that elements which are not capable of actuation, i.e. for which  $\beta = 0$ , should not measure their own state of strain.

An example of an optimized system taken from Paper I is shown in Fig. 8. The NMSMD is clamped at the left end and a stiff spring is attached to the rightmost node, to which a vertically directed force is also applied. By itself, this force is unable to induce any significant displacement, but through the strain sensors in the structure the elements shown in blue are activated and increase their respective length to move the controlled node upwards. For moderately obvious reasons, this type of mechanism is referred to as a *force amplifier*. An interesting type of force amplifiers are artificial exoskeletons, and with further development it is possible that NMSMDs could be used as such.

When designing the system depicted in Fig. 8, both the neural network connections and the element volumes was subject to optimization. The problem in Paper I included a cost function depending on the number of elements, and to minimize the value of this function, 28 of the 45 elements in the ground structure (not shown) have been removed. Noteworthy is also the absence of direct neural network connections;

<sup>2</sup>Line search or trust-region methods based on the Newton step [92, chap. 3-4], for instance.



**Figure 8:** An NMSMD functioning as a force amplifier.

i.e.,  $\mathbf{W} = \mathbf{0}$ . The fact that no direct connections — the neurons are still indirectly connected through the sensor term and the mechanical structure — were needed was discovered by solving (40) for different values of  $\gamma$  and noting that systems showing small differences between their the actual and desired shapes could be obtained independently of the degree of penalization of the neural network weights. For the other example studied in Paper I (and Paper IV) however, the so-called *force inverter*, non-zero neural network weights were found to be necessary for any system that were to complete the prescribed task.

### 5.1.1 A simultaneous formulation

Numerical experiments with the nested problem in Paper I (some of which are described therein), resembling (40), showed that finding acceptable solutions can be difficult due to frequent convergence to spurious local minima. Hoping to alleviate this issue, specifically by avoiding the nonlinear design-to-state mapping (see section 3.2.3), we therefore considered a different formulation of the problem in Paper IV,

resembling the following simultaneous version of (40):

$$\begin{aligned}
 & \underset{\mathbf{p} \in \mathcal{H}, \mathbf{u} \in \mathbb{R}^n, \mathbf{v} \in \mathbb{R}^m, \boldsymbol{\varepsilon} \in \mathbb{R}^m, \mathbf{f} \in \mathbb{R}^m}{\text{minimize}} \quad \left\{ \max_{i=1, \dots, n_c} h_i(\mathbf{u}) \right\} + \gamma \Phi(\boldsymbol{\omega}) \\
 & \text{subject to} \quad \begin{cases} \|\mathbf{f}^a(\mathbf{v})\|^2 \leq E_{max} \\ \sum_{i=1}^m [f_i + f_i^a(v_i)] (\mathbf{b}_i + \mathbf{B}_i \mathbf{u}) + \mathbf{K}_0 \mathbf{u} = \mathbf{F} \\ f_i = E \mathcal{V}_i \varepsilon_i, & i = 1, \dots, m \\ \varepsilon_i = \mathbf{b}_i^\top \mathbf{u} + \frac{1}{2} \mathbf{u}^\top \mathbf{B}_i \mathbf{u}, & i = 1, \dots, m \\ \mathbf{0} = \boldsymbol{\phi}(\mathbf{u}, \mathbf{v}) + \mathbf{W} \mathbf{s}(\mathbf{v}) - \mathbf{C} \mathbf{v} + \mathbf{I} \\ \underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}}, \end{cases} \quad (44)
 \end{aligned}$$

where box constraints on the displacements have been added primarily to avoid certain pathological solutions (see Paper IV). The mechanical equation of state (41a) has here been split in three parts to reduce the degree of nonlinearity slightly. To this end, two new sets of variables which were previously dependent quantities have been introduced: the bar forces  $f_i, i = 1, \dots, m$ , and the strains  $\varepsilon_i, i = 1, \dots, m$ .

Although problem (44) is much larger in terms of both number of constraints and variables than (40), it turned out to be easier to solve in the sense that systems with small shape errors could be obtained from almost any starting point. However, in Paper IV, when carrying out dynamic simulations to validate the optimized systems, it was found that the target configurations attained by optimized system — i.e., the optimal values of  $\mathbf{u}$ ,  $\boldsymbol{\varepsilon}$ ,  $\mathbf{f}$ , and  $\mathbf{v}$  — could not be reached except when launching the ODE-solver<sup>3</sup> with initial states very close to their respective optimal values. In practice, however, one expects an NMSMD to take on the prescribed shape for a wider range of initial conditions, in particular with states initialized to zero. The conclusion is therefore that (44) may not be a suitable problem formulation for design of NMSMDs.

## 5.2 Neuro-Mechanical Oscillators

In contrast to NMSMDs, NMOs are designed for dynamic applications, specifically to generate oscillatory, periodic motion. The work on NMOs described herein was partly inspired by reading about biological system, where oscillatory motion driven by neural networks play a crucial role [57]. Locomotor activity such as walking or swimming are good examples of this. Not surprisingly, robots controlled by artificial neural networks have received interest recently [97, 109, 64], and the NMOs considered here could be viewed as examples of such bio-inspired robots or machines.

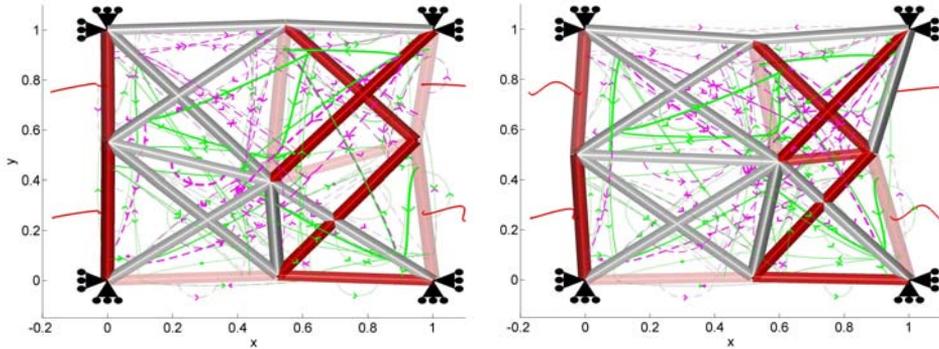
Two different principles underlying the functioning of an NMO are encountered in this thesis: In Paper II, periodic motion is generated as the neural subsystem

<sup>3</sup>ode15s from the Matlab ODE suite [112] has been used throughout this work.

converges into a limit cycle after application of a constant input signal. In Paper III on the other hand, limit cycle behavior is considered undesirable, and motion results instead from the application of periodic input signals to the neural subsystem. The former principle allows for a certain autonomy of the system, while the latter enables a greater degree of external control — the frequency of the periodic motion, for instance, is eventually the same as the frequency of the input signal. Which of these approaches is preferable depends on the application.

Inspired by the way muscle work, the active forces [recall (22)] have been restricted to be non-negative in the NMO-model. This means that the elements are only capable of one-way actuation, namely contraction. Borrowing an idea from the field of binary robots [23], they are also modeled as binary in the sense that, in principle, either they take on the value zero, or they take on their respective maximum value. These restrictions imply that individual elements can be made very simple, which should facilitate robust implementations since no special mechanisms for two-way actuation or precision control are needed.

An example of an NMO in action is shown in Fig. 9. The system is fixed at all



**Figure 9:** An NMO shown at two instants in time. Left:  $t = 1.3$ . Right:  $t = 1.5$ .

four corners and has been designed such that the center node moves counterclockwise in a circle of a given radius when the system is subjected to certain non-mechanical input stimuli. Elements that are actively contracting are shown (in red) with various levels of transparency, where fully opaque elements are those generating the largest active contraction forces. Unlike what was the case for the force amplifier shown in Fig. 8, a weight matrix with at least some non-zero entries is necessary for the functioning of an NMO. Each curved plain (green) or dashed (magenta) line in Fig. 9 therefore represents a neural network connection. The thickness and style of each line is determined by the weight of the connection times the instantaneous output from the neuron at which it originates, i.e.  $W_{ij}s_j(v_j)$  for the line connecting element  $j$  to element  $i$ . The curved lines can thus also be said to represent the "activity" of the neural subsystem.

### 5.2.1 The design problem

The goal of attacking the optimal design problem is to obtain an NMO with the property that some of its nodes follow prescribed, periodic, trajectories given a certain input stimulus  $\mathbf{I}$  to the system. In addition, the system should also use a minimal amount of energy for actuation. Collecting the target values for the displacement components to be controlled in  $\mathbf{u}^* = \mathbf{u}^*(t)$ , a nested version of the optimal design problem can be stated as follows:

$$\begin{aligned} & \underset{\mathbf{p} \in \mathcal{H}}{\text{minimize}} && \int_{T_0}^T \mathbf{f}^a(\mathbf{v}(\mathbf{p}))^\top \overline{\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{p}))} \, dt \\ & \text{subject to} && \int_{T_0}^T \|\mathbf{L}\mathbf{u}(\mathbf{p}) - \mathbf{u}^*\|^2 \, dt = 0, \end{aligned} \quad (45)$$

where  $\overline{\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{p}))} = \nabla \boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{p}))^\top \dot{\mathbf{u}}(\mathbf{p})$ ,  $0 < T_0 < T$ , and the matrix  $\mathbf{L}$  picks out the displacement components of interest. That the integral in the objective really is the energy spent on actuation, over the interval  $[T_0, T]$ , is clear from (19) and (23). Time histories of the state variables in (45) are obtained by solving state problems of the form

Given  $\mathbf{I}$ ,  $\mathbf{p}$ , and  $\mathbf{x}_0$ , find  $\mathbf{x} = (\dot{\mathbf{u}}, \mathbf{u}, \mathbf{v})$ ,  $t \in [0, T]$ , such that

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{A}\dot{\mathbf{u}} + \sum_{i=1}^{n_{el}} \left[ \mathcal{Y}_i E \boldsymbol{\varepsilon}_i(\mathbf{u}) + f_i^a(v_i) \right] \nabla \boldsymbol{\varepsilon}_i(\mathbf{u}) + \mathbf{K}_0 \mathbf{u} = \mathbf{F} \quad (46a)$$

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{W}\mathbf{s}(\mathbf{v}) - \mathbf{C}\mathbf{v} + \mathbf{I} \\ \mathbf{x}(0) &= \mathbf{x}_0. \end{aligned} \quad (46b)$$

Figure 9 shows an example of the type of systems that can be obtained, using a suitable ground structure, from problem (45). This problem is non-convex in general, so one should expect different designs depending on the initial guess.

Apart from the addition of the inertial term in (46a), the difference between (46) and (42) is that the sensor term is missing in (46b). This omission was made because strain sensors play no immediately obvious role in NMOs. It has been shown, however, that feedback from strain sensors is important for efficient motor control in animals [91], and an interesting future study could therefore be to investigate in what situations NMOs would benefit from the addition of such sensors.

Another thing that may be noted is that, depending on fixed system parameters, such as initial nodal positions, problem (45) may lack solutions due to the constraint on the displacement trajectories. An obvious reformulation that results in a well-posed problem is to remove this constraint and add the displacement error functional, call it  $H$ , to the objective (possibly times some weight factor). However, in practice it turned out that for such a formulation, the optimization solvers<sup>4</sup> were much more likely to converge to points where the fit to the target trajectories was unsatisfactory.

<sup>4</sup>In this case interior-point and SQP algorithms available in `fmincon` [121] and `KNITRO` [22].

A possible explanation for this behavior is that there are, in general, many more points satisfying  $\nabla_{\mathbf{p}} H(\mathbf{p}) = \mathbf{0}$  than  $H(\mathbf{p}) = 0$ .

Finally, it should also be mentioned that a version of (45) of the simultaneous form (11) has been tested. However, since a relatively fine time-discretization was needed to ensure that state trajectories were accurately represented by solutions (this was checked by comparing solutions to results from simulations using an ODE-solver), the resulting problems became rather large and could not be solved as efficiently as their nested counterparts. This observation is based on preliminary results, however, so we refrain from drawing a definite conclusion of how the problem should best be formulated, but note that this is an interesting avenue for future investigations.

## 5.2.2 Design with stability constraints

As indicated above, a drawback of letting the neural subsystem converge into a limit cycle is that the NMO may become difficult to control. Responses to variations in the external stimuli may be unpredictable, and the motion of the system may be highly dependent on initial conditions. If external control of the system is desired, limit cycle behavior must be prevented and the system should have the property that it eventually "forgets" its initial state, meaning that after some time, the motion is determined by the input stimulus. Following [96], we refer to a system having this property as *convergent*. More precisely, an NMO is locally convergent if it has a locally attractive solution associated with a given input stimulus  $\mathbf{I} = \mathbf{I}(t)$  and an initial state  $\mathbf{x}_0$  such that for every  $\mathbf{x}'_0$  sufficiently close to  $\mathbf{x}_0$ ,

$$\lim_{t \rightarrow \infty} \|\mathbf{x}(t, \mathbf{x}_0, \mathbf{I}) - \mathbf{x}'(t, \mathbf{x}'_0, \mathbf{I})\| = 0.$$

Since NMOs are designed for generating periodic motion it is interesting to note that if the input signal to a convergent system is periodic with period  $\tau$ , then so is the motion of the system [96]<sup>5</sup>.

The possibility of large deformations means that is probably not possible to obtain global results, but in Paper III it is shown that an NMO is locally convergent if

1. there exists a positive semi-definite, symmetric matrix  $\mathbf{P}$  such that the matrix

$$\begin{pmatrix} -\mathbf{C}\mathbf{P} - \mathbf{P}\mathbf{C} & \mathbf{P}\mathbf{W} + \mathbb{I} \\ \mathbf{W}^{\top}\mathbf{P} + \mathbb{I} & -2\text{diag}\{L_i\} \end{pmatrix}$$

is negative definite (here  $L_i$ ,  $i = 1, \dots, m$ , are Lipschitz constants for the activation functions); or

2. there exists positive constants  $\lambda_i$ ,  $i = 1, \dots, m$ , such that

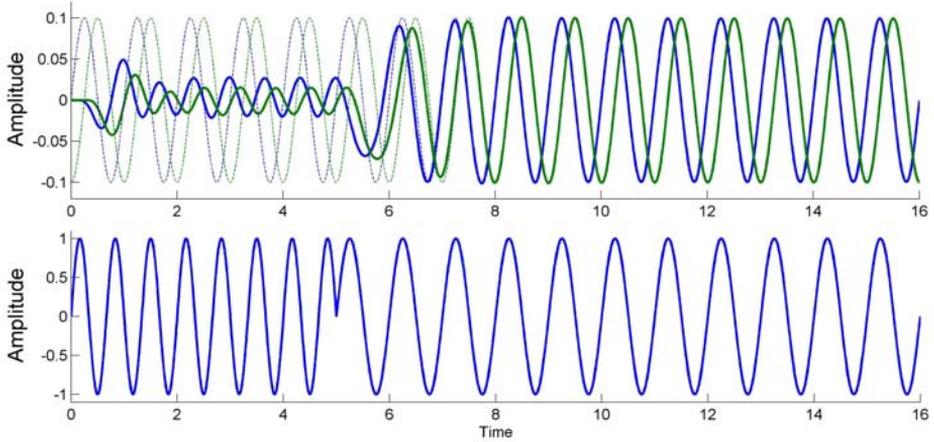
$$-\lambda_i(c_i - L_i W_{ii}) + L_i \sum_{j \in \mathcal{N}_i, j \neq i} \lambda_j |W_{ji}| < 0, \quad i = 1, \dots, m.$$

---

<sup>5</sup>This is not true for general nonlinear systems as demonstrated by  $\ddot{x} + x = (x^2 + \dot{x}^2 - 1) \sin(\sqrt{2}t)$ , which has a periodic solution  $x = \sin(t)$  despite the fact that the input signal has period  $2\pi/\sqrt{2}$  [44, p. 30].

Solving problem (45) augmented with either of these conditions — possibly reformulated to obtain a standard NLP as in Paper III — as constraints results in a, at least locally, convergent system.

The top plot in Fig. 10 shows the response of a convergent NMO of the type depicted in Fig. 9 in terms of time histories for the displacement components of the center node. The system has been optimized such that this node follows the trajectory



**Figure 10:** Top: Time histories for the displacement components of the center node. Bottom: A non-zero component of the input stimulus  $\mathbf{I}$ .

represented by thin dashed lines in the top plot given a time-varying input stimulus  $\mathbf{I}$  whose non-zero components  $\sin(2\pi t)$  apply to the sides of the NMO, as in Fig. 9. The desired behavior is represented in the plots for  $t \gtrsim 8$ , where an excellent fit of the actual trajectory to the target can be seen. Initially, however, an input stimulus with non-zero components  $\sin(3\pi t)$  is applied. As seen in the top plot in Fig. 10, for  $t < 5$  the actual trajectory clearly differs from the desired, both in terms of frequency and magnitude. Once the original stimulus is applied at  $t = 5$ , however, we see rapid convergence towards the target trajectory — precisely the type of behavior expected from a convergent system.

## 6 Concluding remarks and future work

This thesis has been concerned with the concept of Neuro-Mechanical Networks — active mechanical systems made up from simple building blocks capable of actuation and rudimentary signal processing/control that interact with their nearest neighbors. A specific type of NMNs, referred to as active trusses, has been the focus of study, and a mathematical model for systems of this type including geometrical nonlinearities, actuator saturation, and nonlinear, additive neurons has been established. The notions of Neuro-Mechanical Shape Memory Devices and Neuro-Mechanical Oscilla-

---

tors, based on the active truss model, have been introduced, exemplifying the use of NMNs in both static and dynamic situations.

Future research on NMNs could go in many different directions. Continuing with the active truss model, this could be extended to include phenomena such as friction in the joints, communication delays in the neural subsystem, and nonlinear material behavior. Flow of heat and energy within the system could also be taken into account. Another direction of future work is towards implementation of physical prototypes. Such efforts would probably benefit from studying work done on modular self-reconfigurable robots [119]. Formulation of optimal design problems that guarantee systems which degrade gracefully, as mentioned in section 2.1, could also be considered.

In the author's opinion, Neuro-Mechanical Oscillators is the most interesting concept introduced in this thesis; in fact, it could be argued that NMSMDs are just special cases of NMOs. In addition to the suggestions for studying the incorporation of strain sensors and simultaneous formulations of the design problem given in section 5.2, a natural extension of the work on NMOs presented herein is also to consider optimization of element cross-sectional areas, or volumes, and nodal positions. Another idea is to use NMOs as basis for explanatory models of biological systems [72].

Finally, non-convexity has been an issue throughout this work. Unless one resorts to small displacement models this is probably unavoidable, but there may be other ways to alleviate the problems associated with non-convexity. One possibility could be to switch from nested to simultaneous formulations, another to apply algorithms for global optimization.

# I An invariance principle

An invariant set  $M$  for the autonomous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{47}$$

is a set such that if  $\mathbf{x}(0) \in M$ , then  $\mathbf{x}(t) \in M$  for all  $t \geq 0$ . Let  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  be a continuously differentiable, radially unbounded (Lyapunov) function such that  $\dot{V}(\mathbf{x}) \leq 0$  for all  $\mathbf{x}$ . Now, consider the set  $E = \{\mathbf{x} \in \mathbb{R}^n \mid \dot{V}(\mathbf{x}) = 0\}$  and let  $M$  be the largest invariant set in  $E$ . From LaSalle's invariance principle [74, Theorem 1]<sup>1</sup> it then follows that every solution to (47) will tend to  $M$  as  $t \rightarrow \infty$ . Here it is interesting to consider the case when

$$\dot{\mathbf{x}} = \mathbf{0} \Leftrightarrow \dot{V}(\mathbf{x}) = 0,$$

i.e.  $M$  is the set of all equilibrium points for (47). Since the set of points to which a bounded solution tends as  $t \rightarrow \infty$  (the  $\omega$ -limit set) is connected [98, Theorem 1, p. 188], convergence to an equilibrium point follows if every point in  $M$  is isolated (two or more isolated points do not form a connected set). It may be the case, however, that  $M$  contains connected sets of equilibria, and in this case the invariance principle only guarantees so-called quasi-convergence [50]. Furthermore, note that even if all equilibria are isolated, the arguments above shows only convergence to an equilibrium point; unless  $V$  is positive definite in some vicinity, this point need not be stable.

# II A design-to-state mapping

Consider the bar shown in Fig. 11 below, and let the cross-sectional area  $A > 0$  be the design variable. If the length of the bar  $l = 1$ , and the Green-Lagrange strain measure



**Figure 11:** A single bar subject to a force.

is used (see section 4.1), the equilibrium equation in terms of the displacement  $u$  is given by

$$u^3 + 3u^2 + 2u = \frac{2F}{AE},$$

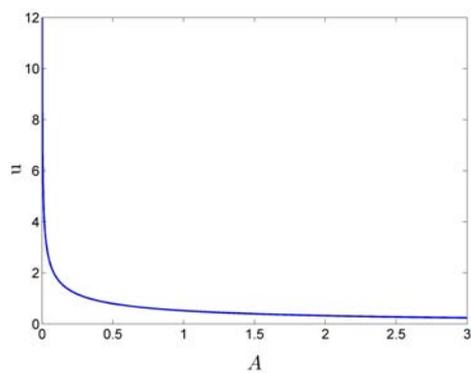
where  $F$  and  $E$  is the applied force and Young's modulus, respectively. Assuming  $F/E = 1$  and solving for  $u$  we get the real root

$$u(A) = \frac{1}{3((1/A^2 - 1/27)^{1/2} + 1/A)^{1/3}} + ((1/A^2 - 1/27)^{1/2} + 1/A)^{1/3} - 1,$$

<sup>1</sup>See [75] for some historical notes on the invariance principle.

---

which is plotted as a function of  $A$  in Fig. 12.



**Figure 12:** Displacement  $u$  as a function of cross-sectional area  $A$ .

## References

- [1] Abdalla M, Frecker M, Gurdal Z, Johnson T, Lindner D (2005) Design of a piezoelectric actuator and compliant mechanism combination for maximum energy efficiency. *Smart materials and structures* 14:1421–1430
- [2] Achtziger W (1997) Topology optimization of discrete structures - an introduction in view of computational and nonsmooth aspects. In: Rozwany G (ed) *Topology Optimization in Structural Mechanics, CISM Courses and Lectures*, Springer, pp 57–100
- [3] Alkhatib R, Golnaraghi M (2003) Active structural vibration control: A review. *The Shock and Vibration Digest* 35:367–383
- [4] Andréasson N, Evgrafov A, Patriksson M (2005) *An Introduction to Continuous Optimization*. Studentlitteratur
- [5] Angeles J (2007) *Fundamentals of Robotic Mechanical Systems*. Springer
- [6] Anton S, Sodano H (2007) A review of power harvesting using piezoelectric materials (2003–2006). *Smart Materials and Structures* 16:1–21
- [7] Arora J, Wang Q (2005) Review of formulations for structural and mechanical system optimization. *Structural and Multidisciplinary Optimization* 30:251–272
- [8] Atiya A, Parlos A (2000) New results on recurrent network training: Unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks* 11:697–709
- [9] Atmel Corporation (2011) Datasheet for ATtiny48/88
- [10] Bar-Cohen Y (2006) Biomimetics - using nature to inspire human innovation. *Bioinspiration & Biomimetics* 1:1–12
- [11] Bazaraa M, Sherali H, Shetty C (1993) *Nonlinear programming - Theory and algorithms*, 2nd edn. John Wiley & Sons
- [12] Begg D, Liu X (2000) On simultaneous optimization of smart structures – Part I: Theory. *Computer methods in applied mechanics and engineering* 184:25–37
- [13] Begg D, Liu X (2000) On simultaneous optimization of smart structures – Part II: Algorithms and examples. *Computer methods in applied mechanics and engineering* 184:25–37
- [14] Bendsøe M, Sigmund O (2003) *Topology Optimization: Theory, Methods and Applications*. Springer-Verlag
- [15] Bendsøe M, Olhoff N, Taylor J (1987) On the design of structure and controls for optimal performance of actively controlled flexible structures. *Mechanics of Structures and Machines* 15:265–295
- [16] Bendsøe M, Ben-Tal A, Zowe J (1994) Optimization methods for truss geometry and topology design. *Structural Optimization* 7:141–159
- [17] Betts J (2001) *Practical Methods for Optimal Control Using Nonlinear Programming*. Advances in Design and Control, SIAM
- [18] Bishop H (ed) (2008) *Mechatronic systems, sensors and actuators*. CRC Press
- [19] Boyd S, Vandenberghe L (2004) *Convex Optimization*. Cambridge University Press

- 
- [20] Brignell J (1987) Digital compensation of sensors. *Journal of Physics E: Scientific Instruments* 20:1097–1102
- [21] Bruns T (2006) Zero density lower bounds in topology optimization. *Computer Methods in Applied Mechanics and Engineering* 196:566–578
- [22] Byrd R, Nocedal J, Waltz R (2006) KNITRO: An integrated package for nonlinear optimization. In: *Large-Scale Nonlinear Optimization*, Springer, pp 35–59
- [23] Chirikjian G (1994) A binary paradigm for robotic manipulators. In: *IEEE Int. Conference on Robotics and Automotion*, San Diego USA
- [24] Christensen P, Klarbring A (2009) *An Introduction to Structural Optimization*. Springer
- [25] Chua L, Roska T (1993) The CNN paradigm. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications* 40:147–156
- [26] Crawley E (1994) Intelligent structures for aerospace: A technology overview and assessment. *AIAA Journal* 32:1689–1699
- [27] Crisfield M (1991) *Non-linear Finite Element Analysis of Solids and Structures*. John Wiley & Sons
- [28] Culshaw B (1996) *Smart Structures and Materials*. Artech house, inc
- [29] D’Andrea R, Dullerud G (2003) Distributed control design for spatially interconnected systems. *IEEE transactions on automatic control* 48:1478–1495
- [30] Donoso A, Sigmund O (2009) Optimization of piezoelectric bimorph actuators with active damping for static and dynamic loads. *Structural and Multidisciplinary Optimization* 38:171–183
- [31] Dorn W (1964) Automatic design of optimal structures. *Journal de Mecanique* 6:25–52
- [32] Ebenbauer C, Raff T, Allgower F (2009) Dissipation inequalities in systems theory: An introduction and recent results. In: Jeltsch R, Wanner G (eds) *Invited Lectures of the International Congress on Industrial and Applied Mathematics 2007, CISM Courses and Lectures*, European Mathematical Society Publishing House, pp 23–42
- [33] Edwards R (1996) Approximation of neural network dynamics by reaction-diffusion equations. *Mathematical Methods in the Applied Sciences* 19:651–677
- [34] Fathy H, Reyer J, Papalambros P, Ulsoy A (2001) On the coupling between the plant and controller optimization problems. In: *Proceedings of the American Control Conference*, pp 1864–1869
- [35] Feng C, Plamondon R, O’Reilly C (2010) On some necessary and sufficient conditions for a recurrent neural network model with time delays to generate oscillations. *IEEE Transactions on neural networks* 21
- [36] Forti M, Tesi A (2004) Absolute stability of analytic neural networks: An approach based on finite trajectory length. *IEEE Transactions on Circuits and Systems - I: Regular Papers* 51
- [37] Frecker M (2003) Recent advances in optimization of smart structures and actuators. *Journal of Intelligent Material Systems and Structures* 14:207–216
- [38] Giurgiuti V (2001) Actuators and smart structures. In: Ewins D, Braun S, Rao S (eds) *Encyclopedia of Vibrations*, Elsevier, pp 58–81

- [39] Giurgiutui V (2006) Concepts of adaptronic structures. *Automatisierungstechnik* 54:251–258
- [40] Granado J, Vega M, Perez R, Sanchez J, Gomez J (2006) Using FPGAs to implement artificial neural networks. In: *IEEE International Conference on Electronics, Circuits and Systems*, 2006. ICECS '06. 13th, pp 934–937
- [41] Grossberg S (1988) Nonlinear neural networks: principles, mechanisms, and architecture. *Neural Networks* 1:17–61
- [42] Guran A (ed) (1998) *Structronic Systems: Smart Structures, Devices and Systems*. World Scientific
- [43] Haftka R, Adelman H (1985) An analytical investigation of shape control of large space structures by applied temperatures. *AIAA Journal* 23:450–457
- [44] Hagedorn P (1988) *Non-linear Oscillations*, 2nd edn. Oxford University press
- [45] Hahn W (1967) *Stability of Motion*. Springer
- [46] Hartman P (1964) *Ordinary Differential Equations*. John Wiley & Sons
- [47] Haug E, Arora J (1978) Design sensitivity analysis of elastic mechanical systems. *Computer Methods in Applied Mechanics and Engineering* 15:35–62
- [48] Haupt R, Haupt S (2004) *Practical Genetic Algorithms*. Wiley-Interscience
- [49] Haykin S (2009) *Neural Networks and Learning Machines*. Prentice Hall
- [50] Hirsch M (1989) Convergent activation dynamics in continuous time networks. *Neural Networks* 2:331–349
- [51] Hirsch M, Smale S (1974) *Differential Equations, Dynamical Systems, and Linear algebra*. Academic press
- [52] Hollerbach J, Hunter I, Ballantyne J (1992) A comparative analysis of actuator technologies for robotics. In: Khatib O, Craig J, Lozano-Pérez T (eds) *The Robotics Review 2*, Mit Press
- [53] Holnicki-Szulc J (ed) (2008) *Smart Technologies for Safety Engineering*. John Wiley & Sons
- [54] Hopfield J (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academic Science, USA* 79:2554–2558
- [55] Hopfield J (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of National Academic Science, USA* 81:3088–3092
- [56] Huber J, Fleck N, Ashby M (1997) The selection of mechanical actuators based on performance indices. *Proceeding of the Royal Society A: Mathematical, Physical and Engineering Sciences* 453:2185–2205
- [57] Ijspeert A (2008) Central pattern generators for locomotion controls in animals and robots: A review. *Neural Networks* 21:642–653
- [58] Iorga L, Baruh H, Ursu I (2008) A review of  $H_\infty$  robust control of piezoelectric smart structures. *Applied Mechanics Reviews* 61
- [59] Irschik H (2002) A review on static and dynamic shape control of structures by piezoelectric actuation. *Engineering Structures* 24:5–11

- 
- [60] Isermann R (2003) *Mechatronic Systems: Fundamentals*. Springer-Verlag
- [61] ITRS (2011) International technology roadmap for semiconductors - executive summary. <http://www.itrs.net/Links/2011ITRS/Home2011.htm>, Accessed March 2012
- [62] Janocha H (ed) (1999) *Adaptronics and Smart Structures*. Springer-Verlag
- [63] Jantapremjit P, Austin D (2001) Design of a modular self-reconfigurable robot. In: Proceedings of Australian conference on robotics and automation
- [64] Jin HL, Zacksenhouse M (2003) Oscillatory neural networks for robotic yo-yo control. *IEEE Transactions on Neural Networks* 14:317–325
- [65] Kamps M, Velde F (2001) From artificial neural networks to spiking neuron populations and back again. *Neural Networks* 14:941–953
- [66] Kang Z, Tong L (2008) Integrated optimization of material layout and control voltage for piezoelectric laminated plates. *Journal of Intelligent Material Systems and Structures* 19:889–904
- [67] Kang Z, Tong L (2008) Topology optimization-based distribution of actuation voltage in static shape control of plates. *Computers and Structures* 86:1885–1893
- [68] Khalil H (2002) *Nonlinear systems*. Pearson Education
- [69] Kitano H (2002) Computational systems biology. *Nature* 420:206–210
- [70] Kögl M, Silva E (2005) Topology optimization of smart structures: design of piezoelectric plate and shell actuators. *Smart Materials and Structures* 14:387–399
- [71] Kočvara M (2002) On the modeling and solving of the truss design problem with global stability constraints. *Structural and Multidisciplinary Optimization* 23:189–203
- [72] Krus P, Karlsson M (2002) Neuro-mechanical networks - self-organizing multifunctional systems. In: KA E (ed) *Ninth Bath International Fluid Power Workshop*
- [73] Kyura N, Oho H (1996) *Mechatronics - an industrial perspective*. *IEEE/ASME Transactions on Mechatronics* 1:10–15
- [74] LaSalle J (1960) Some extensions of Liapunov's second method. *IRE Transactions on Circuit Theory* 7:520–527
- [75] LaSalle J (1967) An invariance principle in the theory of stability. In: *Differential Equations and Dynamical Systems*, Academic Press, pp 277–286
- [76] LaSalle J, Lefschetz S (1961) *Stability by Liapunov's Direct Method with Applications*. Academic Press
- [77] Lee J (2003) *Introduction to smooth manifolds*. Springer
- [78] Lokhov V, Nyashin Y, Ziegler F (2009) Statement and solution of optimal problems for independent stress and deformation control by eigenstrain. *Journal of Applied Mathematics and Mechanics* 89:320–332
- [79] MacLennan B (2007) A review of analog computing. Tech. rep., Department of Electrical Engineering & Computer Science University of Tennessee, Knoxville

- [80] Madden J (2004) Actuator selection for variable camber foils. In: Proc. SPIE - Int. Soc. Opt. Eng. (USA), vol 5385, pp 442–448
- [81] Madden J, Vandesteeg N, Anquetil P, Madden P, Takshi A, Pytel R, Lafontaine S, Wieringa P, Hunter I (2004) Artificial muscle technology: Physical principles and naval prospects. *IEEE Journal of Oceanic Engineering* 29:706–728
- [82] Magnusson J (2009) Neuro Mechanical Networks - The concept and optimal design. Linköping University, Licentiate thesis
- [83] Magnusson J, Klarbring A, Sethson M (2009) Design and configuration of neuro mechanical networks. *Structural and Multidisciplinary Optimization* 37:335–350
- [84] Magnusson J, Klarbring A, Sethson M (2009) Design and configuration of neuro mechanical networks. *Structural and Multidisciplinary Optimization* 37:335–350
- [85] Marco M, Forti M, Tesi A (2002) Existence and characterization of limit cycles in nearly symmetric neural networks. *IEEE transactions on circuits and systems – 1: Fundamental theory and applications* 49:979–992
- [86] Marsden J, Hughes T (1994) *Mathematical Foundations of Elasticity*. Dover
- [87] Martinez P, Marti P, Querin O (2007) Growth method for size, topology, and geometry optimization of truss structures. *Structural and Multidisciplinary Optimization* 33:13–26
- [88] Mitchell M (1996) Computation in cellular automata: A selected review. In: *Nonstandard Computation*, Wiley-VCH, pp 95–140
- [89] Moreau J (1974) On unilateral constraints, friction and plasticity. In: Capriz G, Stampacchia G (eds) *New Variational Techniques in Mathematical Physics*, Edizioni Cremonese, pp 58–81
- [90] Newcomb R, Lohn J (1995) Analog VLSI for neural networks. In: Arbib M (ed) *Handbook of Brain Theory and Neural Networks*, MIT Press
- [91] Nishikawa K, Bienwener A, Aerts P, Ahn A, Chiel H, Daley M, Daniel T, Full R, Hale M, Hedrick T, Lappin A, Nichols T, Quinn R, Satterlie R, Szymik B (2007) *Neuromechanics: an integrative approach for understanding motor control*. *Integrative and Comparative Biology* 47:16–54
- [92] Nocedal J, Wright S (2006) *Numerical Optimization*, 2nd edn. Springer
- [93] Nyashin Y, Likhov V, Ziegler F (2005) Decomposition method in linear elastic problems with eigenstrain. *Journal of Applied Mathematics and Mechanics* 85:557–570
- [94] Padula S, Kincaid R (1999) Optimization strategies for sensor and actuator placement. Tech. rep., NASA, Langley Research
- [95] Palm R, Korenivski V (2007) A ferrofluid based neural network: design of an analogue associative memory. pre-print [ArXiv:0705.3402v1](https://arxiv.org/abs/0705.3402v1)
- [96] Pavlov A, Wouw N, Nijmeijer H (2005) Convergent systems: Analysis and synthesis. In: Meurer T, Graichen K, Gilles E (eds) *Control and Observer Design for Nonlinear Finite and Infinite Dimensional Systems*, Lecture Notes in Control and Information Sciences, vol 322, Springer, pp 131–146
- [97] Pelc E, Daley M, Ferris D (2008) Resonant hopping of a robot controlled by an artificial neural oscillator. *Bioinspiration & Biomimetics* 3:1–11

- 
- [98] Perko L (2001) *Differential equations and dynamical systems*. Third Edition. Springer
- [99] Pons J (2005) *Emerging actuator technologies: a micromechatronic approach*. John Wiley and Sons Ltd
- [100] Prasad V, Sharma R (1987) Sufficient conditions for existence and diffeomorphism of a nonlinear vector function. *Proceedings of the IEEE* 75:846–847
- [101] Preumont A (1997) *Vibration control of active structures: An introduction*. Kluwer Academic Publishers
- [102] Preumont A (2006) *Mechatronics: Dynamics of Electromechanical and Piezoelectric Systems*. Springer
- [103] Rauli M, Maute K (2005) Topology optimization of electrostatically actuated microsystems. *Structural and Multidisciplinary Optimization* 30:342–359
- [104] Reyer J (2000) *Combined embodiment design and control optimization: Effect of cross-disciplinary coupling*. PhD thesis, The University of Michigan
- [105] Richardson W (1989) Comments on "Sufficient conditions for existence and diffeomorphism of a nonlinear vector function". *Proceedings of the IEEE* 77:496–497
- [106] Rudin W (1976) *Principles of Mathematical Analysis*. McGraw-Hill
- [107] Salama M, Garba J, Demsetz L, Udawadia F (1988) Simultaneous optimization of controlled structures. *Computational Mechanics* 3:275–282
- [108] Sandberg A, Bostrom N (2008) *Whole brain emulation – A roadmap*. Tech. rep., Future of Humanity Institute, Faculty of Philosophy & James Martin 21st Century School Oxford University
- [109] Sato T, Kano T, Ishiguro A (2011) On the applicability of the decentralized control mechanism extracted from the true slime mold: a robotic case study with a serpentine robot. *Bioinspiration & Biomimetics* 6
- [110] Sethson M, Krus P, Karlsson M (2002) Neuro-mechanical networks, self-organizing multifunctional systems. In: *Proceedings of 17th European Simulation Multi-Conference*.
- [111] Sethson M, Karlsson M, Krus P (2003) Neuro-mechanical networks as an architecture for system design. In: *Proceedings of AAAI-Synthesis symposium at Stanford University, USA AAAI*
- [112] Shampine L, Reichelt M (1997) The MATLAB ODE suite. *SIAM Journal on Scientific Computing* 18:1–22
- [113] Sigmund O (2001) Design of multiphysics actuators using topology optimization - part 1: One-material structures. *Computer Methods in Applied Mechanics and Engineering* 190:6577–6604
- [114] Sigmund O (2001) Design of multiphysics actuators using topology optimization - part 2: Two-material structures. *Computer Methods in Applied Mechanics and Engineering* 190:6605–6627
- [115] Simpson T, Martins J (2011) *Multidisciplinary design optimization for complex engineered systems: Report from a national science foundation workshop*. *Journal of Mechanical Design* 133
- [116] Smith K (1999) Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing* 11:15–34

- [117] Steil J (2000) Input-output stability of recurrent neural networks with time-varying parameters
- [118] Stolpe M, Svanberg K (2001) An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary optimization* 22:116–124
- [119] Stoy K, Brandt D, Christensen D (2010) *Self-Reconfigurable Robots: An Introduction (Intelligent Robotics and Autonomous Agents)*. The MIT Press
- [120] Takahashi M, Kobori T, Nasu T, Niwa N, Kurata N (1998) Active response control of buildings for large earthquakes - seismic response control system with variable structural characteristics. *Smart Materials and Structures* 7:522–529
- [121] The MathWorks Inc (2011) *Optimization Toolbox - User's Guide*. Natick, MA, 2000
- [122] Thore CJ (2011) Optimal design of neuro-mechanical oscillators. Submitted
- [123] Thore CJ (2011) Optimal design of neuro-mechanical oscillators with stability constraints. Submitted
- [124] Utku S (1998) *Theory of adaptive structures: incorporating intelligence into engineered products*. CRC Press
- [125] Varadan V, Vinoy K, Gopalakrishnan S (2006) *Smart Material Systems and MEMS*. John Wiley & Sons
- [126] Vidyasagar M (1993) *Nonlinear Systems Analysis*, 2nd edn. Prentice-Hall
- [127] Willems J (1972) Dissipative dynamical systems – Part 1: General theory. *Archive of Rational Mechanics and Analysis* 45:351–393
- [128] Witt G (1974) The electromechanical properties of thin films and the thin film strain gauge. *Thin Solid Films* 22:133–156
- [129] Wu F, Desoer C (1972) Global inverse function theorem. *IEEE Transactions on Circuit Theory* 19
- [130] Yim M, Shen WM, Salemi B, Rus D, Moll M, Lipson H, Klavins E, Chirikjan G (2007) Modular self-reconfigurable robot systems. *IEEE Robotics & Automotion Magazine* 14:43–52
- [131] Zhu Y, Qiu J, Du H, Tani J (2002) Simultaneous optimal design of structural topology, actuator locations and control parameters for a plate structure. *Computational Mechanics* 19:89–97
- [132] Ziegler F (2005) Computational aspects of structural shape control. *Computers and Structures* 83:1191–1204



**Linköping University**  
**INSTITUTE OF TECHNOLOGY**

Division of Mechanics  
Department of Management and Engineering

ISBN 978-91-7519-900-9 ISSN 0345-7524