

# Linköping University Post Print

## Design and configuration of neuro mechanical networks

Johan Magnusson, Anders Klarbring and Magnus Sethson

N.B.: When citing this work, cite the original article.

The original publication is available at [www.springerlink.com](http://www.springerlink.com):

Johan Magnusson, Anders Klarbring and Magnus Sethson , Design and configuration of neuro mechanical networks, 2009, STRUCTURAL AND MULTIDISCIPLINARY OPTIMIZATION, (37), 4, 335-350.

<http://dx.doi.org/10.1007/s00158-008-0241-8>

Copyright: Springer Science Business Media

<http://www.springerlink.com/>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-16143>

Johan Magnusson · Anders Klarbring · Magnus Sethson

# Design and Configuration of Neuro Mechanical Networks

the date of receipt and acceptance should be inserted later

**Abstract** Neuro Mechanical Network (NMN) is a new concept of adaptronic character. The governing idea is to include geometry, topology, load carrying, energy transfer, actuating, sensing and control of a machine in one single mathematical state model, and thereby enabling a formulation of the design and configuration problem as an optimization problem.

We have focused our attention on a type of NMNs consisting of what we call active trusses. For these we have established a state model and given a design optimization problem from which we have obtained numerical solutions. These solutions show that the approach has the possibility to suggest new families of designs that are superior to those of classical passive trusses. We also indicate how activation may result in singularities, the treatment of which is so far essentially an open problem.

**Keywords** Adaptronics, Actuated structures, Neural networks, Topology optimization

---

## 1 Introduction

Network systems have been the subject of increasing interest during the last decades. One of the more striking features of networks is that complex properties can emerge although the basic elements are very simple. This is particularly true in the case of neural networks that have become a firmly established discipline for signal processing, control systems, and mathematical function

---

Johan Magnusson (✉) · Anders Klarbring  
Division of Mechanics  
E-mail: johan.magnusson@liu.se  
E-mail: anders.klarbring@liu.se

Magnus Sethson  
Division of Fluid and Mechanical Engineering Systems  
E-mail: magnus.sethson@liu.se

Institute of Technology  
Linköping University  
SE-581 83 Linköping, Sweden

approximations. There is a strong element of inspiration from biological systems in this field as there are examples of networks made up of simple elements with spectacular properties in nature. One astonishing feature found in nature is that simple elements have more than one specific role or function to play: they become multi-functional elements. In this work we use a synthesized view of neural networks and adaptive mechanical structures that results in what we call Neuro Mechanical Networks (NMNs). The NMN concept was first introduced in Krus and Karlsson (2002), Sethson et al. (2003a) and Sethson et al. (2003b), and is herein refined and made concrete for an active truss. Examples of NMNs from nature are muscles, especially the heart muscle, given its autonomous character. From a mathematical modeling point of view, NMNs are neural networks superimposed onto mechanical networks, e.g. trusses. This transforms classical passive mechanical structures into active structures that can sense its loading environment and react accordingly.

The task of designing or configuring NMNs from simple but multi-functional elements is a complex problem, where it is difficult to imagine that traditional intuitive, and slowly evolving, design strategies will be very successful. However, for passive structures we have seen the development of topology optimization methods in recent decades, see e.g. Bendsoe and Sigmund (2003), and it is quite reasonable to assume that these methods could be extended to include NMNs. This paper, besides introducing the general idea of NMNs and the NMN element, intends to show a first application of topology optimization methods for configuring NMNs. We superimpose a simple neural network on a classical truss structure and use the fact that both have the same type of network structure to develop a unified topology optimization algorithm for a fully integrated and simultaneous design of structure, actuation, and control. The traditional ground structure approach of topology optimization is extended by including not only a large number of potential bars but also a large number of possible neural connections, i.e. signal scaling weights. By a good choice of design

variables, unwanted bars and neural connections can be removed from the ground structure resulting in an optimal topology.

There is probably no well established theoretical field in which NMNs can be placed. It belongs to an intersection of different fields, where some of the ideas are shared. Building adaptive structures out of structural elements, actuators, and sensors with integrated control electronics is often referred to as adaptronics, smart structures or structronics, Janocha (1999). These ideas were examined already in the late 1980s and early 1990s: a theory for building cranes as adaptive, actuated trusses was presented in Utku et al. (1989). In Tanaka and Hanahara (1991), the concept is developed to include a neural network for control of the motion, but here the design of the structure is given a priori and the design process includes only the motion and the neural network. Much of the work in later years seems to focus on different active and smart materials, used to build adaptronics, see e.g. Chopra (2002) and references therein. In the part of the field focused on structures and their mechanical behavior, much work centers on using piezoelectric force actuators for active vibration control, health monitoring in structures and on using adaptive structures to control the shape of functional surfaces in aerodynamics. In tune with the development of methods in structural and topology optimization these new methods have also been adopted to the design process of adaptronics. In Liu and Begg (2000) and Begg and Liu (2000) they use a method for optimizing the topology of a truss, the placement of actuators and the control system simultaneously. They also discuss the importance of allowing the structures topology to change in the design process and not only designing an actuator and control system for a given structure. Even though they use topology optimization for the structure design problem, they view the actuator placement problem as a discrete problem and develop a method based on a combination of topology optimization and simulated annealing/genetic algorithms for simultaneous optimization. From the point of view of nature as inspiration, the work of Lipson and Pollack (2000) is very interesting. They design adaptronic mechanisms, or robotic life forms to use their own term, as actuated truss mechanisms controlled by neural networks, through an evolutionary process in form of genetic algorithms. They also use rapid prototyping technology to realize the optimized structures as demonstrators. However, having made this review of previous work, it is clear that the general concept of NMN presented in this paper is a novel one. The most interesting contribution from the work presented in this paper is the way in which all functions in an NMN, e.g. the topology, actuating and control, are described in one single mathematical formulation and how this formulation makes it possible to optimize all these parts in one simultaneous process without the structural geometry given a priori or the need for binary optimization of actuator placement.

Practical realization of NMNs may lie years ahead, but with this work we want to provide a readiness for when the enabling technologies arrive on the scene. Developments of materials like electro-active polymers and manufacturing techniques such as 3-D printing are interesting areas from a NMN point of view.

In Section 2 we introduce the NMN concept and the ideas acting as it's foundation, the NMN-element, discuss how a general strategy for configuration might look, and discuss some possible applications. In Section 3 we discuss the special type of NMNs that we call active trusses, followed by the topology optimization problem in Section 4. Section 5 gives numerical results that suggest families of designs.

---

## 2 What is a Neuro Mechanical Network?

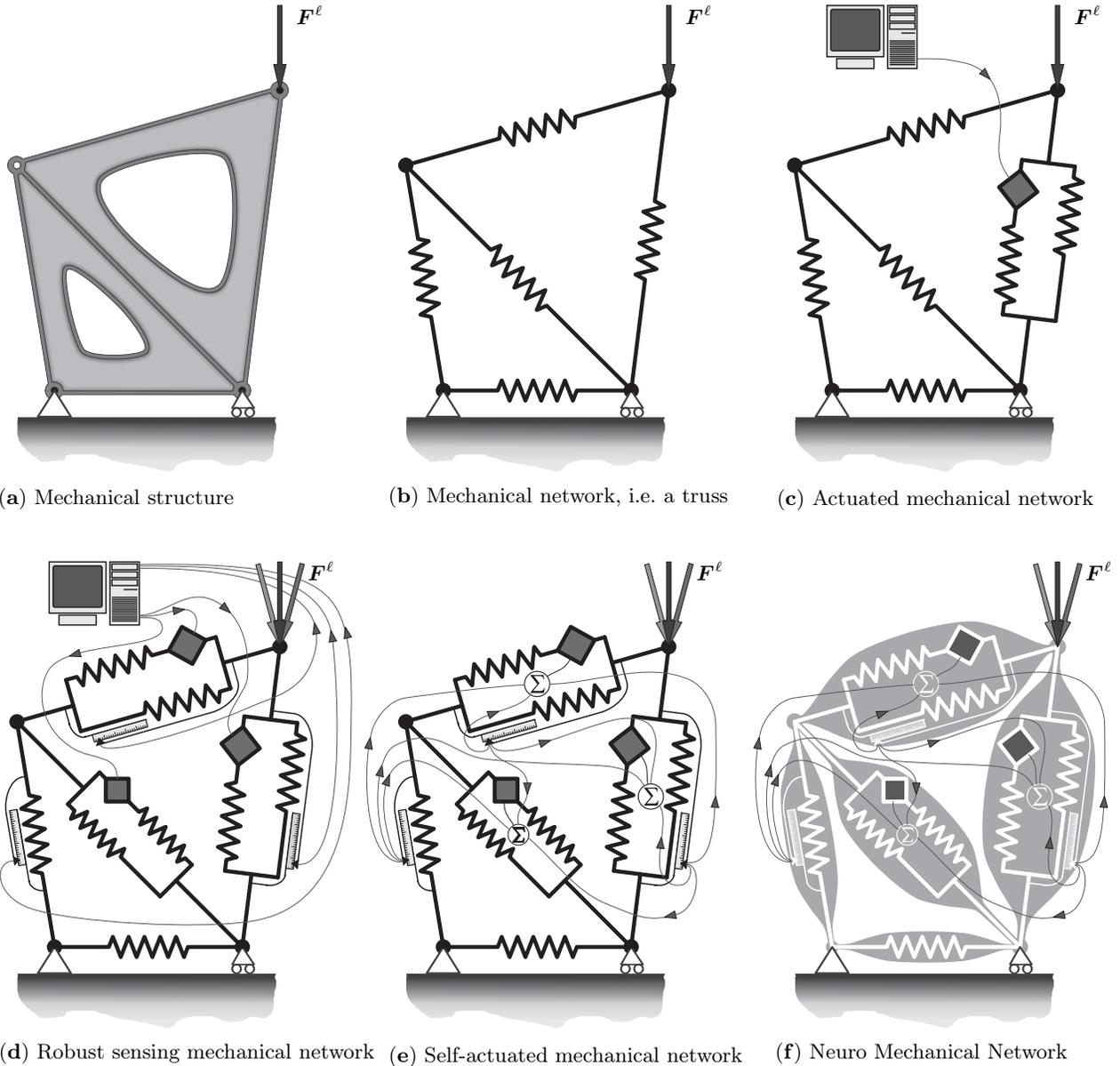
In this section we give a description of the ideas behind the NMN concept. We begin by establishing the concept of NMNs and the NMN element. This is followed by a section on how to design and configure these networks and a section about characteristic properties and future use.

### 2.1 The general idea of Neuro Mechanical Networks

The idea of NMN is inspired by the way complicated actuated mechanical systems function in nature, more precisely by the way in which nature uses networks of huge numbers of multi-functional elements to solve mechanical tasks such as load carrying and motion planing. The network idea is to use many standardized and uniform elements to build systems, where the overall functionality and performance of the system is significantly higher than for each isolated element. With multi-functional elements it is possible to create all necessary functions of the system in an integrated process. One example from nature is our muscles, which are built from millions of muscle myofibers that, apart from the ability to hold loads and contract themselves, also have functions for energy transformation, information storage, and waste and heat removal. In this way the muscle gets all of its functionality from just one type of elements, and the same type of elements, myofibers, can be used for the smallest muscle in a mouse as well as for the largest in an elephant.

The NMN concept is gradually established in Figure 1. In part **(a)** of the figure we have a one-piece mechanical structure, specifically designed to perform some well defined load-carrying assignment, under restrictions on, for example, geometry, total mass or mechanical behavior. A drawback of such a design is that for every new task, a new structure has to be designed and manufactured.

In Figure 1**(b)** the number of structural members is increased and the complexity of each member can thus



**Fig. 1:** The NMN idea and concept are established gradually from a one-piece mechanical structure to a Neuro Mechanical Network (NMN).

be substantially decreased without any loss of system performance. We have a mechanical network where the design task is ideally to configure the network by finding the right topology and then choosing among a set of standard elements. Building the network is also a simple task: just assemble the elements with joints at each node. For a new load-carrying assignment, only a reconfiguration is needed and the elements can be reused in new networks.

In Figure 1(c) an element with actuation capabilities is introduced. This gives the network some motion ability, typically used to compensate for deformation under the applied load,  $F^\ell$ , or to let some point follow a pre-

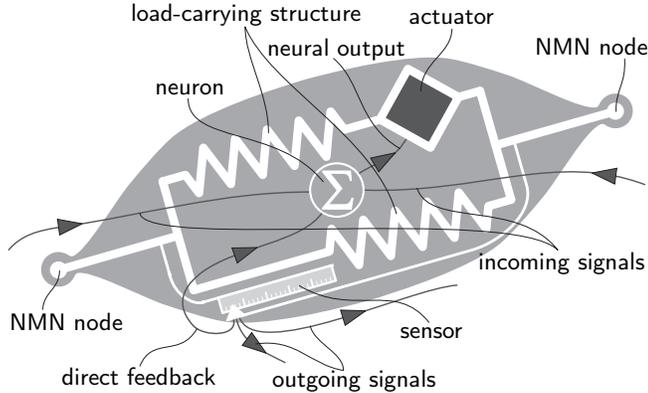
defined motion path. To control the motion some sort of external control system, e.g. a computer or a micro processor, is required.

In Figure 1(d) the number of actuator elements is increased and the elements have been given the ability to sense the load or its state in some way, meaning that the structure can react to a load scenario and become more robust to load variances or other changes in its environment. With more actuated elements and sensors the external control system must be upgraded or changed to handle the cumulative control task.

Figure 1(e) may be the most essential figure for understanding the NMN concept. Here the control system

is constructed as a network, e.g. a neural network, and in that way it can be distributed in the mechanical network. The result is a self-actuating network where the only external supply is the power supply.

In the last part of the figure, (f), the load-carrying, the actuation and the state sensing features are combined with the neuron and form a single sensing and actuating unit, the NMN element. With these multi-functional elements it is possible to build active networks that are autonomous and can sense and react to its mechanical environment, in short NMNs. The present work deals with modeling and configuring such networks.



**Fig. 2:** In a closer view we see the different functionalities of a multi-functional NMN element

In Figure 2 one single NMN element is shown in close-up and the different functions are marked. In addition to what has already been mentioned, important features that we need from the NMN element are that the values of the different properties can be chosen independently of each other and in a very wide range. In that way, the NMN can be used in micromechanisms as well as in huge construction designs like cranes and bridges. Another neat property is that the element interface could be designed with mechanical and signal contacts gathered for easy assembly and disassembly of elements, even by the structure itself - self-configuration.

Figure 3 shows a more complete view of how a neural network is superimposed on a mechanical network resulting in the NMN. Figure 3(a) is the mechanical network, where the elements are structural members, e.g. bars, beams or shafts, and the nodes are joints connecting the elements. The difference between different kinds of mechanical networks is in the way elements and joints transfer loads. Figure 3(b) shows a neural network that has its nodes in layers named input, output, and hidden layer. The nodes in the input layer simply serve to introduce the values of the input variables, often from sensor signals. The number of hidden layers can vary from zero and upwards. The nodes in the hidden and output layers are neurons, having one connection to each unit in the preceding layer. A neuron takes a number of input sig-

nals and returns one output signal based on the sum of the input signals. In mathematical form this reads

$$S_i = f_i \left( \sum_{j=0}^n s_{ij} w_{ij} \right), \quad (1)$$

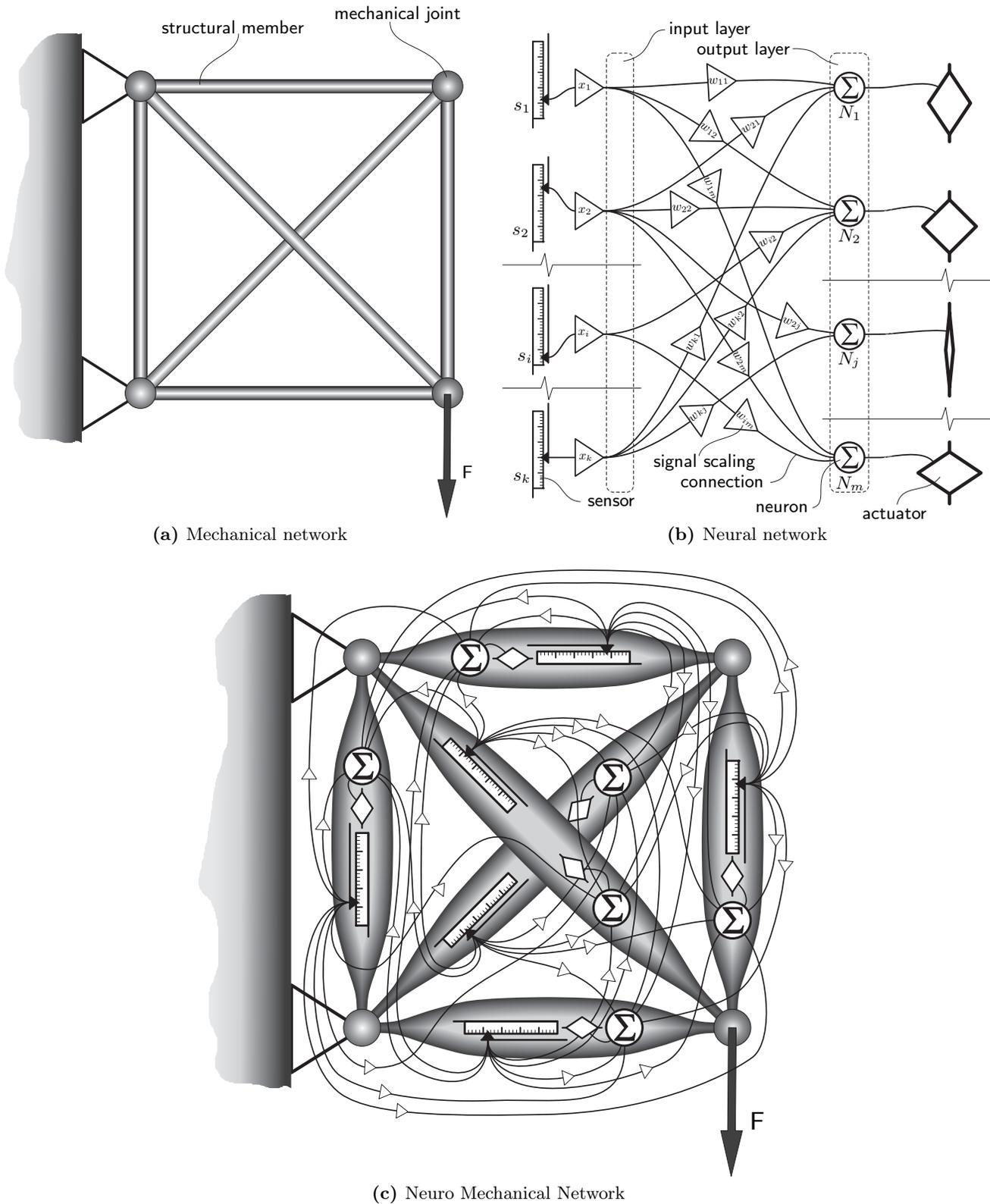
where  $S_i$  is the output signal from neuron  $i$ ,  $f_i$  is the activation function for neuron  $i$ ,  $s_{ij}$  is the signal from node  $j$  to neuron  $i$ , and  $w_{ij}$  is the corresponding neural scaling weight with  $w_{i0}$  being the threshold value for neuron  $i$ . The elements in a neural network are the connections between the neurons (and input nodes) in the different layers. For every connection the signal is scaled by a scaling weight. The response of a neural network is controlled by choosing the scaling weights. Making this choice based on experimental or other input data is usually called training or learning. More about neural networks can be found in Haykin (1999). In this work the training is accomplished through solving an optimization problem. In Figure 3(c) the neural network is superimposed onto the mechanical network, and an input and an output node of the neural network is combined with every element in the mechanical network. Note that herein lies one of the main computational challenges in NMN configuration since the potential number of connections will explode when the number of mechanical nodes is increased: for truss structures with a one layer neural network, explicitly treated in this work, we have approximately  $n^4/4$  design variables for a network of  $n$  mechanical nodes.

## 2.2 Design and configuration of NMNs

In this subsection we discuss the general nature of the NMN design problem and what would be a suitable method for solving such a problem.

The overall function and performance of a network is defined by the local properties of the nodes and the elements, the topology of the network and, in case of a mechanical network, also by the global geometry. Many of the properties are decided a priori by what type of network is used, e.g. should the mechanical network be a truss or a frame? Left to decide in the actual design process is the topology and the values of some local size-type variables for each node and element, and for some types of networks also geometry. This design stage is often referred to as configuration of the network, and this is what is dealt with in this paper.

From the configuration point of view, the NMN is a mechanical network with a neural network superimposed on it, see Figure 3. Therefore, the configuration process of the NMN is a simultaneous process of configuring two different networks. The fact that the number of possible neural connections explodes when the number of mechanical nodes is increased makes this problem



**Fig. 3:** A schematic picture of the NMN as a neural network superimposed on a mechanical network. Figure (a) shows an ordinary mechanical network with bars and joints as elements and nodes. Figure (b) shows a neural network with connections and neurons as elements and nodes. It also shows how the neural network receive the input signals from sensors and how the output signals are fed to the actuators. Figure (c) shows how the neural network is superimposed onto the mechanical network.

into a large scale problem and some automatic, or at least semi-automatic, configuration method is needed.

Optimization methods in general may be used to find a large number of local variables in order to achieve the best overall performance for a system. Topology optimization in particular solves just the type of problem we have at hand. It helps us choose values of local variables for each node and element in an optimal way, and at the same time it lets us change the topology of the network. Topology optimization, and in a wider perspective structural optimization, are widely used for different design purposes as seen for example in Bendsoe and Sigmund (2003). In this paper we use topology optimization to configure the NMNs.

Topology optimization of discrete structures uses a ground structure approach. In the domain of the physical space where the structure is to be placed the number and positions of connecting nodes are set. The start solution, called the ground structure, is then all possible elements connecting the nodes, or at least a large number of these possible elements. Then, by allowing optimal values of the size-type design variables to be zero, some elements disappear from the solution and the topology is changeable. A similar strategy may be used for the neural network: for each input node a neighborhood of neurons or output nodes is defined, and scaling weights for each connection between an input node and one of its neighbors are introduced. By allowing these scaling weights to be zero, connections disappear from the optimal network and the topology of the neural network is defined. Obviously, some sort of compatibility between the topologies of the mechanical and the neural network needs to be enforced, e.g. an element that is not present in the mechanical network should not be allowed to send a signal affecting the neural network. Further, an element from which there are no outgoing neural connections has no sensor, and an element with no incoming connections has no neuron or actuator.

Summarizing the above discussion, the following steps are involved in designing NMNs:

- Choose the type of mechanical network, e.g. truss or beam structure, small or large deformation setting, type of actuation and neural network, e.g. with or without hidden layers.
- Model the behavior of the combined network and the control of the actuation.
- Choose proper design variables, typically some element stiffness associated variables for the mechanical network and some signal processing variables for the neural network.
- Establish an objective function that in some way quantifies the property we want to optimize, typically some measure of overall stiffness, mass or the deviation from some pre-described motion.
- Establish constraints adequate for the chosen design variables and objective function.

- Set the design domain, number of mechanical nodes and their placements.
- Define the possible neural connections, i.e. the neural neighborhoods.
- Specify the load scenario.
- Use an optimization algorithm to find an optimal topology and local sizes.

The output from the optimization algorithm is a list of values of design variables, where a zero means no element or no connection. Together with general data about the ground structures, this describes an optimal configuration of the NMN.

### 2.3 Characteristic properties and areas of use of NMNs

In this section we will look at the applications and properties we want to achieve by use of NMNs. We will then mainly be concerned with structures in a small displacement range and the objective is to create structures with high artificial or adaptive effective stiffness.

A structure with high effective stiffness has small deformations at points and directions of applied loads, but could have large deformations elsewhere. One reason why the effective stiffness is an interesting property is that the point where a structure is loaded is in many cases the point where the structure is in contact with other structures or systems.

Artificial or adaptive effective stiffness is the term used when actuation and active control is used in a structure to increase its effective stiffness. The NMN may be configured so that actuation compensates for the deformation in the direction of loads. In other words, instead of using more and stiffer material, energy and information are used to achieve an increased effective stiffness. The fact that it is possible to control energy supply and information transfer in the structure makes these stiffness properties adaptive: it is possible to create structures with high effective stiffness just where and when it is needed.

One application of structures with almost infinite adaptive stiffness is when extremely high precision is needed, for example in adaptive manufacturing rigs for the aerospace and automotive industry. When the load is changing, due to changes in the weight of the workpiece along the production line, different product configurations or different production operations, the rig automatically compensates for deformations and the position of the workpiece can be predicted with high accuracy. Another case where increased and adaptive stiffness could lead to increased precision is when these structures are used as load-carrying elements in industrial robots.

Yet another conceivable usage is to save weight and volume in structures where the load is applied only in a small region of the structure at each time. Think of a long structure with the load moving along it, such as a

railway bridge or a portal crane. In such cases the possibility of moving the stiffness, in form of energy and information, along the structure may be used. Instead of using a large amount of material and building a structure that always has sufficient stiffness for the maximum load at all parts, NMNs can be used to significantly save weight and volume.

The fact that NMNs are networks built from standardized multi-functional elements simply joined together at the nodes, gives several useful properties. One is the modularity: from a set of NMN elements with different lengths and stiffnesses it is easy and fast to run the configuration algorithm for some specific load carrying purpose and then to build the structure as an NMN and choose appropriate neuron scaling weights. When the structure has fulfilled its task, the elements can easily be reused in new structures for a different load-carrying purpose. Another useful property is the multi-functionality of the elements: with NMN elements there is no need to design and build a subsystem for each function, and then assemble at the end for full functionality. This makes it possibly to design and build NMNs with all their functions very quickly.

The proposed method for configuration of active structure may also be used for concept generation at early stages of the development of other more complicated products, where actuation, sensing, and control are part of their function. By configuring an NMN for the load scenarios the product is exposed to, the designer gets an idea of where and how to use ordinary material stiffness, actuated elements, and where to collect information for the control system.

In a longer perspective, when the NMN concept is used not only on structures in a small deformation setting, but also in a large deformation setting and with dynamic effects taken into consideration, mechanisms can be designed in a fully automated way and built as NMNs. These actuated mechanisms could be used as simpler types of industrial robots and reconfigurable machines for motion generation.

---

### 3 The state problem

In this paper the focus is on a specific type of NMNs called active trusses. An active truss is an NMN where the mechanical network is a truss, the actuation is a linear motion, the sensors measure length changes, and the neural network is a simple one-layer network.

#### 3.1 The mechanical truss

A ground structure consists of  $m$  NMN elements, which, from a passive mechanical point of view, are essentially bars that have the possibility of length actuation. These bars are connected at nodes that carry a displacement

state having  $n$  degrees of freedom. The geometry of the structure is defined by the matrix  $\mathbf{B}$  given as

$$\mathbf{B} = \begin{bmatrix} \gamma_1^\top \\ \vdots \\ \gamma_m^\top \end{bmatrix},$$

where  $\gamma_i = \bar{\gamma}_i/L_i$  in which  $\bar{\gamma}_i$  is a vector<sup>1</sup> containing direction cosines for the  $i$ :th bar or element,  $L_i$  is the initial length of that element and  $^\top$  denotes the transpose of a vector or matrix. The state of the structure is described by a vector  $\mathbf{u} \in \mathbb{R}^n$  of nodal displacements. Let the strain of element  $i$  be denoted by  $\varepsilon_i$  and collect all of those in a vector  $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ . In a small displacement setting it then holds that

$$\varepsilon_i = \gamma_i^\top \mathbf{u}, \quad i = 1, \dots, m \iff \boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}. \quad (2)$$

The stress in element  $i$ , denoted  $\sigma_i$ , relates to the strain and an actuation  $\alpha_i$  according to

$$\sigma_i = E_i (\varepsilon_i - \alpha_i), \quad (3)$$

where  $E_i$  is element  $i$ 's material stiffness, i.e. a Young's modulus-type constant. Thus, the actuation is taken as an additional strain, which can be thought of as a change of the natural stress free length of the bar. One may compare this to a classical additional strain due to temperature.

Let  $x_i$  denote the material volume of element  $i$  and collect all such volumes in the vector  $\mathbf{x} \in \mathbb{R}^m$ . For obvious physical reasons we have the constraint

$$x_i \geq 0, \quad i = 1, \dots, m \iff \mathbf{x} \geq \mathbf{0}.$$

In the optimization problem to be formulated below,  $\mathbf{x}$  will be taken as one of the vectors of design variables and if an optimal solution shows  $x_i = 0$ , we interpret that as if bar  $i$  has been removed from the ground structure: herein lies a topology optimization feature of the problem formulation.

Next, a force-like variable, i.e. force times element length,

$$s_i = x_i \sigma_i, \quad (4)$$

is defined. Now (3) and (4) give

$$\begin{aligned} s_i = x_i E_i (\varepsilon_i - \alpha_i), \quad i = 1, \dots, m &\iff \\ \iff \mathbf{s} = \mathbf{D}(\mathbf{x})(\boldsymbol{\varepsilon} - \mathbf{a}), \end{aligned} \quad (5)$$

where  $\mathbf{D}(\mathbf{x}) = \text{diag}\{x_i E_i\}$  and  $\mathbf{a} \in \mathbb{R}^m$  is a vector formed from the element actuations. Furthermore, assuming a static or quasi static situation, the force equilibrium reads

$$\mathbf{F} = \sum_{i=1}^m s_i \gamma_i \iff \mathbf{F} = \mathbf{B}^\top \mathbf{s}, \quad (6)$$

---

<sup>1</sup> We use the term vector as an abbreviated form of column vector.

where  $\mathbf{F} \in \mathbb{R}^n$  is the external forces at the nodes, collected in a vector.

Finally, with (2), (5) and (6), the mechanical part of the state equation for the system reads

$$\mathbf{F} = \sum_{i=1}^m \gamma_i x_i E_i (\gamma_i^\top \mathbf{u} - \alpha_i) = \mathbf{B}^\top \mathbf{D}(\mathbf{x}) (\mathbf{B}\mathbf{u} - \mathbf{a}). \quad (7)$$

### 3.2 The neural network

As indicated in Section ??, every bar  $i$  has a sensor that measures its change of length, i.e. essentially the strain  $\varepsilon_i$ . This value is multiplied by the bar volume, and the resulting signal, i.e.  $x_i \varepsilon_i$ , is feed to the neurons of all bars in a neighborhood of bar  $i$ . Such a neighborhood is defined by a set of indices, denoted  $\mathcal{N}_i$ , that contains indices of all bars that are connected to bar  $i$  by the neural network. The specification of all such neighborhoods is part of setting up the optimization problem, equivalent to specifying the ground structure for the mechanical part of the problem. The multiplication of the sensor signal by  $x_i$  is essential for the topological feature of the problem, since we do not want non-existing elements ( $x_i = 0$ ) to affect the activation.

The signal from the sensor in element  $j$  to the neuron in element  $i$  is sent through the neural scaling weight  $w_{ij}$ . The neurons sum up all such incoming signals and send the result to actuators, which change the lengths of the bars in proportion to the signal. Summarizing, the actuation of element  $i$  is

$$\alpha_i = \sum_{j \in \mathcal{N}_i} w_{ij} x_j \varepsilon_j, \quad i = 1, \dots, m \iff \iff \mathbf{a} = \mathbf{W}(\mathbf{x}, \mathbf{w}) \boldsymbol{\varepsilon} \quad (8)$$

where  $\mathbf{w}$  is a vector of scaling weights  $w_{ij}$  and  $\mathbf{W}(\mathbf{x}, \mathbf{w})$  is a matrix of elements  $w_{ij} x_j$  and zeros at places  $(i, j)$  where  $j \notin \mathcal{N}_i$ .

From (2), (7) and (8) we finally get the state equation for an NMN, in the form of an active truss, as

$$\begin{aligned} \mathbf{F} &= \sum_{i=1}^m \gamma_i x_i E_i \left( \gamma_i^\top \mathbf{u} - \sum_{j \in \mathcal{N}_i} w_{ij} x_j \varepsilon_j \right) = \\ &= \mathbf{B}^\top \mathbf{D}(\mathbf{x}) (\mathbf{I} - \mathbf{W}(\mathbf{x}, \mathbf{w})) \mathbf{B}\mathbf{u}, \end{aligned}$$

where  $\mathbf{I}$  is the identity matrix. This state equation can be written more compactly as

$$\mathbf{F} = \mathbf{H}(\mathbf{x}, \mathbf{w}) \mathbf{u}, \quad (9)$$

by introducing the scaling weight dependent stiffness-like matrix

$$\mathbf{H}(\mathbf{x}, \mathbf{w}) = \mathbf{B}^\top \mathbf{D}(\mathbf{x}) (\mathbf{I} - \mathbf{W}(\mathbf{x}, \mathbf{w})) \mathbf{B}, \quad (10)$$

to be called the adaptive stiffness matrix in the following. Note, however, that  $\mathbf{H}(\mathbf{x}, \mathbf{w})$  will, in general, be

a nonsymmetric matrix so the analogy with a standard mechanical stiffness matrix is not entirely appropriate.

In the following we will formulate an optimization problem where  $\mathbf{x}$  and  $\mathbf{w}$  will be taken as design variables. If  $\mathbf{H}(\mathbf{x}, \mathbf{w})$  is non-singular, (9) can be used to obtain the state, i.e. the nodal displacements, for every load scenario and set of design variables.

### 3.3 Remarks on work and energy

From (2) and (6) we obtain the standard work equivalence of linear structures:

$$\mathbf{F}^\top \mathbf{u} = \mathbf{s}^\top \boldsymbol{\varepsilon}. \quad (11)$$

Moreover, (5) can be rewritten as

$$\mathbf{s} + \mathbf{s}^a = \mathbf{D}(\mathbf{x}) \boldsymbol{\varepsilon}, \quad (12)$$

where

$$\mathbf{s}^a = \mathbf{D}(\mathbf{x}) \mathbf{a} \iff s_i^a = x_i E_i \alpha_i, \quad i = 1, \dots, m \quad (13)$$

is the actuation force (times element length  $L_i$ ).

Inserting (12) into (11) we get

$$\mathbf{F}^\top \mathbf{u} + \mathbf{s}^{a\top} \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^\top \mathbf{D}(\mathbf{x}) \boldsymbol{\varepsilon}.$$

This equation shows how two types of external work are in balance with the stored energy of the structure.

From (13) and (8) we find that the actuation force can be written

$$s_i^a = \sum_{j \in \mathcal{N}(i)} x_i E_i w_{ij} x_j \varepsilon_j, \quad i = 1, \dots, m. \quad (14)$$

The physical dimension of  $E_i w_{ij}$  is work  $[m^2 kg/s^2]$  per volume squared  $[m^6]$ . The factor  $x_i E_i w_{ij} x_j$  is the energy spent in transforming a unit strain in element  $j$ , with volume  $x_j$ , to a force (times length) in element  $i$ , with volume  $x_i$ . When formulating the optimization problem in the next section it will be natural to introduce a constraint on the sum of such energies.

---

## 4 The optimization problem

This section establishes the optimization problem for active truss configuration. We choose design variables, state a proper objective function, and find adequate constraints to restrict the solution. Then a modification of the state equation is made in order to adapt it to the topology optimization context. This is followed by a discussion on the singularity or non-singularity of the adaptive stiffness matrix.

#### 4.1 Design variables, objective and constraints

As design variables we use the material volume of the elements,  $x_i$ , to control the mechanical configuration and the neural scaling weight factors,  $w_{ij}$ , to control the actuation. This seems to be a natural choice: the first one is classical in structural topology optimization and the second one is what is chosen when a neural network is trained. Moreover, these variables occur explicitly in the state equation, they describe the local element properties in a direct way, and they make it possible to include topological features in the design: if  $x_i$  is found to be zero, element  $i$  has no volume and, thus, has been removed from the design. In the same way, if  $w_{ij}$  is set to zero, there is no signal from element  $j$  to element  $i$  and, thus, this connection has been removed from the design.

Our goal is to design a structure with as high effective stiffness as possible. As mentioned in Section ??, one measure of such a property is the compliance, which is the sum of all displacements at loaded nodes, parallel to the loads, and scaled by them. The compliance  $C(\mathbf{x}, \mathbf{w})$  for a certain design  $(\mathbf{x}, \mathbf{w})$  is written

$$C(\mathbf{x}, \mathbf{w}) = \frac{1}{2} \mathbf{F}^\top \mathbf{u}(\mathbf{x}, \mathbf{w}),$$

where  $\mathbf{u}(\mathbf{x}, \mathbf{w})$  is the solution of the state equation (9). We are here assuming that this solution is uniquely defined, a property that will be discussed below. Maximizing the effective stiffness is taken as equivalent to minimizing the compliance. However, for an active truss it cannot be assumed that the adaptive stiffness matrix is always positive semi-definite, as is the case for the stiffness matrix of a non-active structure. That means that  $\mathbf{F}^\top \mathbf{u} \stackrel{(9)}{=} \mathbf{u}^\top \mathbf{H}^\top \mathbf{u}$  can in principle take negative values and therefore the squared value of the compliance is used as the objective function. The objective function for one load case is thus

$$f^1(\mathbf{x}, \mathbf{w}) = (\mathbf{F}^\top \mathbf{u}(\mathbf{x}, \mathbf{w}))^2. \quad (15)$$

In many applications, and in particular in the NMN case, it is preferable to optimize for a load scenario with more than one load case. Otherwise, there is a risk for mechanically unstable solutions or solutions that are very non-robust with respect to variations in the load scenarios. Moreover, the thought of any adaptive effective stiffness is worthless with only one single load case present. Therefore, we assume  $L$  different load cases, represented by load vectors  $\mathbf{F}^\ell$ ,  $\ell = 1, \dots, L$ . Solving the state equation (9) for each load case, i.e.

$$\mathbf{F}^\ell = \mathbf{H}(\mathbf{x}, \mathbf{w}) \mathbf{u}^\ell, \quad \ell = 1, \dots, L,$$

gives  $L$  displacement vectors  $\mathbf{u}^\ell = \mathbf{u}^\ell(\mathbf{x}, \mathbf{w})$ . For a set of weighting factors,  $\alpha^\ell$ ,  $\ell = 1, \dots, L$ , the following objective function is used:

$$f^L(\mathbf{x}, \mathbf{w}) = \sum_{\ell=1}^L \alpha^\ell (\mathbf{F}^{\ell\top} \mathbf{u}^\ell(\mathbf{x}, \mathbf{w}))^2.$$

To restrict the solution, some constraints must be used. A natural constraint, classical in topology optimization, is

$$\sum_{i=1}^m x_i \leq V, \quad (16)$$

where  $V$  is a constant representing the total available material volume. Of course we also use the constraint of non-negative element volumes,  $x_i$ . In analogy with (16) we can state a constraint for the scaling weights:

$$\sum_{i=1}^m \sum_{j \in \mathcal{N}_i} |w_{ij}| \leq A, \quad (17)$$

where  $A$  is a constant representing the total available signal scaling. The absolute value is used because the scaling factors,  $w_{ij}$ , can take both positive and negative values. An alternative to the 1-norm of (17) would be the 2-norm

$$\sum_{i=1}^m \sum_{j \in \mathcal{N}_i} w_{ij}^2 \leq \hat{A}, \quad (18)$$

for some constant  $\hat{A}$ . Yet another alternative would be to bound the  $\infty$ -norm by some constant  $\tilde{A}$ , i.e.,

$$|w_{ij}| \leq \tilde{A} \text{ for all } i, j. \quad (19)$$

The inequality (16) together with any of the inequalities (17) through (19) place a restriction on the total energy per unit strain spent in actuation, cf. Section 3.3. Using (16) and (17) we find

$$\begin{aligned} \sum_{i=1}^m \sum_{j \in \mathcal{N}_i} x_i E_i w_{ij} x_j &\leq \\ &\leq \left[ \max_{1 \leq i \leq m} E_i \right] \left( \sum_{i=1}^m \sum_{j \in \mathcal{N}_i} |w_{ij}| \right) \left( \sum_{i=1}^m x_i \right)^2 \leq \\ &\leq \left[ \max_{1 \leq i \leq m} E_i \right] V^2 A. \end{aligned}$$

Similar inequalities follows on using (18) or (19) instead of (17).

At this stage of development of the NMN concept and active trusses, arguments for making a choice between the three possible constraints (17) through (19) may not be obvious. Therefore, we have made numerical tests, comparing three different formulations involving the three different constraints. A small example consisting of a ground structure of 5 bars and 25 neural connections was considered. The constants  $A$ ,  $\hat{A}$  and  $\tilde{A}$  were chosen so that (17) through (19) defined hypervolumes of the same size. In all three cases 3 out of the 5 bars are present in the optimal structure and the best objective value was obtained when (17) was used. However, what we considered important was the fact that for problems based on (18) and (19) all 9 neural couplings possible for

the 3 bars were present in the optimal structure, while for the problem based on (17) only two such couplings remained. This result holds true for all  $A$ s tested. Since simplicity in this sense is preferable and since we believe this trend to be generally true (confirmed also by the example in Section 5) we have chosen to base our further studies in this paper on (17).

We end this section by summarizing what may be called the unperturbed optimization problem:

$$(\mathbb{P}^L) \quad \min_{(\mathbf{x}, \mathbf{w}) \in \mathcal{D}} \sum_{\ell=1}^L \alpha^\ell \left( \mathbf{F}^{\ell\top} \mathbf{u}^\ell(\mathbf{x}, \mathbf{w}) \right)^2,$$

where

$$\mathcal{D} = \left\{ (\mathbf{x}, \mathbf{w}) : x_i \geq 0, \sum_{i=1}^m x_i \leq V, \sum_{i=1}^m \sum_{j \in \mathcal{N}_i} |w_{ij}| \leq A \right\}$$

is the set of admissible designs. In the next section the possible singularity of the adaptive stiffness matrix leads us to introduce a perturbation of this problem.

#### 4.2 On the possible singularity of the adaptive stiffness matrix

Due to the strategy of topology optimization, where elements removed from the design are represented as existing elements with zero volume, i.e. with no stiffness, the stiffness matrix becomes singular for many designs. The standard approach for avoiding this difficulty is to replace the constraints  $x_i \geq 0$  by  $x_i \geq \epsilon$  for some conveniently small number  $\epsilon > 0$ . However, this approach may be inconvenient in case of an active truss since the variables  $x_i$  have an effect not only on the passive mechanical properties but also on activation levels. Therefore, we chose an approach where we add the passive stiffness corresponding to element material volumes  $\epsilon$  directly to the adaptive stiffness matrix. For a truss without activation, this is equivalent to the classical approach: the two approaches are connected by the change of variables  $x_i \leftrightarrow x_i - \epsilon$ . Thus, we replace  $\mathbf{H}(\mathbf{x}, \mathbf{w})$  by

$$\begin{aligned} \mathbf{H}_\epsilon(\mathbf{x}, \mathbf{w}) &= \\ &= \mathbf{B}^\top (\mathbf{D}(\mathbf{x}) (\mathbf{I} - \mathbf{W}(\mathbf{x}, \mathbf{w})) + \mathbf{D}(\mathbf{1}\epsilon)) \mathbf{B}, \end{aligned} \quad (20)$$

where  $\mathbf{1} \in \mathbb{R}^m$  is a vector of 1:s and  $\epsilon > 0$  is a small number representing a perturbation. The perturbed optimization problem is

$$(\mathbb{P}_\epsilon^L) \quad \min_{(\mathbf{x}, \mathbf{w}) \in \mathcal{D}} \sum_{\ell=1}^L \alpha^\ell \left( \mathbf{F}^{\ell\top} \mathbf{u}_\epsilon^\ell(\mathbf{x}, \mathbf{w}) \right)^2,$$

where  $\mathbf{u}_\epsilon^\ell(\mathbf{x}, \mathbf{w})$  are the solutions of the perturbed state equations

$$\mathbf{F}^\ell = \mathbf{H}_\epsilon(\mathbf{x}, \mathbf{w}) \mathbf{u}_\epsilon^\ell, \quad \ell = 1, \dots, L. \quad (21)$$

For a passive truss, i.e., when  $\mathbf{W}(\mathbf{x}, \mathbf{w})$  is the zero matrix, (20) is a sum of a positive semi-definite matrix and a positive definite matrix and it is thus non-singular. However, for an active truss the perturbation does not guarantee non-singularity. Therefore, we will develop a constraint that implies this property. We note that, since  $\mathbf{B}$  has full column rank,  $\mathbf{H}_\epsilon(\mathbf{x}, \mathbf{w})$  is non-singular if  $\mathbf{D}(\mathbf{x}) + \mathbf{D}(\mathbf{1}\epsilon) - \mathbf{D}(\mathbf{x}) \mathbf{W}(\mathbf{x}, \mathbf{w})$  is so, and this matrix is, in turn, non-singular if and only if

$$\mathbf{I} - (\mathbf{D}(\mathbf{x}) + \mathbf{D}(\mathbf{1}\epsilon))^{-1} \mathbf{D}(\mathbf{x}) \mathbf{W}(\mathbf{x}, \mathbf{w}) \quad (22)$$

is non-singular. Combining, e.g., Propositions 2.2.11 and 2.2.14 in Cottle et al. (1992) one finds that (22) is non-singular if

$$\|\mathbf{W}(\mathbf{x})\|_p < 1, \quad (23)$$

for some matrix norm  $\|\cdot\|_p$ . For a general  $n$  by  $n$  matrix  $\mathbf{A}$  with elements  $a_{ij}$  it holds that

$$\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$$

so (23) holds if

$$x_j \sum_{i \in \mathcal{I}_j} \frac{x_i}{x_i + \epsilon} |w_{ij}| < 1, \quad j = 1, \dots, m,$$

where  $\mathcal{I}_j$  is the set of indices  $i$  for which  $j \in \mathcal{N}_i$ . Since the quotient in (24) is less than 1, we conclude that  $\mathbf{H}_\epsilon(\mathbf{x}, \mathbf{w})$  is non-singularity if

$$x_j \sum_{i \in \mathcal{I}_j} |w_{ij}| < 1, \quad j = 1, \dots, m, \quad (24)$$

This inequality can be guaranteed if we note that it always holds that

$$x_j \sum_{i \in \mathcal{I}_j} |w_{ij}| \leq x_j A, \quad j = 1, \dots, m. \quad (25)$$

Therefore, (24) is satisfied if

$$x_i \leq \frac{1}{A}, \quad i = 1, \dots, m. \quad (26)$$

The box constraints (26) do not noticeably make the problem more expensive to solve, but it will restrict the element volumes severely, at least for large  $A$ , which will result in designs with a large number of thin elements. Therefore, it is a question whether (26) should be added to  $(\mathbb{P}_\epsilon^L)$  or not. This is discussed below.

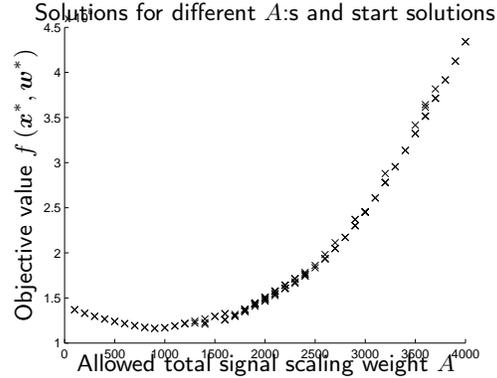
To further understand the nature of singularities, small scale problems can be investigated. Looking at a two bar problem it is clear that singularities appears due to signal feedback. Such insight may be used in future investigations of NMN, but is not considered further in this initial study.

### 4.3 Optimization algorithm

Given the success in classical structural optimization of explicit approximation methods like MMA (Method of Moving Asymptotes), Svanberg (1987), such algorithms would be a natural choice also for the the present problem. There are two points that needs to be addressed, however.

The first point concerns the non-differentiability of the constraint on the scaling weights (the absolute value). Since MMA does not directly handle this property, to use it we need to rewrite the constraint so that it becomes differentiable. This requirers introducing auxiliary variables  $v_{ij}$  such that  $|w_{ij}| \leq v_{ij}$ , i.e., a large number of additional variables and constraints. Another alternative in handling the absolute value is to develop an MMA-like algorithm that handles the non-differentiability directly. It is possible to go through the derivation of MMA and modify those sub-steps where the absolute value appears: this turns out to be when establishing the primal-dual relation. We plan to publish such a derivation in a separate paper. We have tested both of these possibilities with similar results. The numerical experiment presented in Section 5 uses our modified version of MMA.

The second point that should be addressed concerns the singularities of the adaptive stiffness matrix. For singular points the objective function may become infinite. Thus, unless we add constraints that guaranty that such points are not in the admissible set, we cannot expect even globally convergent versions of MMA, Svanberg (1995), to work in all situations. In the previous subsection we derived (26) that restricts the admissible set to non-singular points. To test the usefulness of this constraint it was added to  $(\mathbb{P}_\epsilon^L)$  and the truss bridge problem used for numerical experimentation in Section 5 was solved. Our modified MMA algorithm had no difficulties solving this problem and the objective function value as a function of the constant  $A$  is shown in Figure 4. The problem is solved for several starting points for the same  $A$ , which due to non-convexity sometimes results in slightly different solutions. This is shown as several crosses for the same  $A$  in Figure 4. What is seen in Figure 4 is that the constraints (26) restrict the gain in stiffness from actuation severely. Actually, for large allowed signal scaling weights the objective function value is worse than the value for a classical non-actuated truss. As a consequence of this we have thought it useful to make numerical experiments on the original version of  $(\mathbb{P}_\epsilon^L)$ , even though singularities are present. As is seen in the next section, such experiments makes it possible to obtain families of NMN structures with substantially better stiffness performance than passive structures. However, from a mathematical and numerical point of view we are dealing with a problem and an algorithm that, at least for large actuations, sometimes results in a non-convergent behavior. From the engineering viewpoint of examining the potential possibilities in



**Fig. 4:** The 5 by 2 node truss bridge problem treated in detail in Section 5 is here solved for a modified form of  $(\mathbb{P}_\epsilon^L)$ , where (26) is added to prevent singularities in the adaptive stiffness matrix. The extra constraints turn out to restrict the solution space so that the gain in stiffness from adaptivity is small and occurs only for a limited range of total allowed signal scaling.

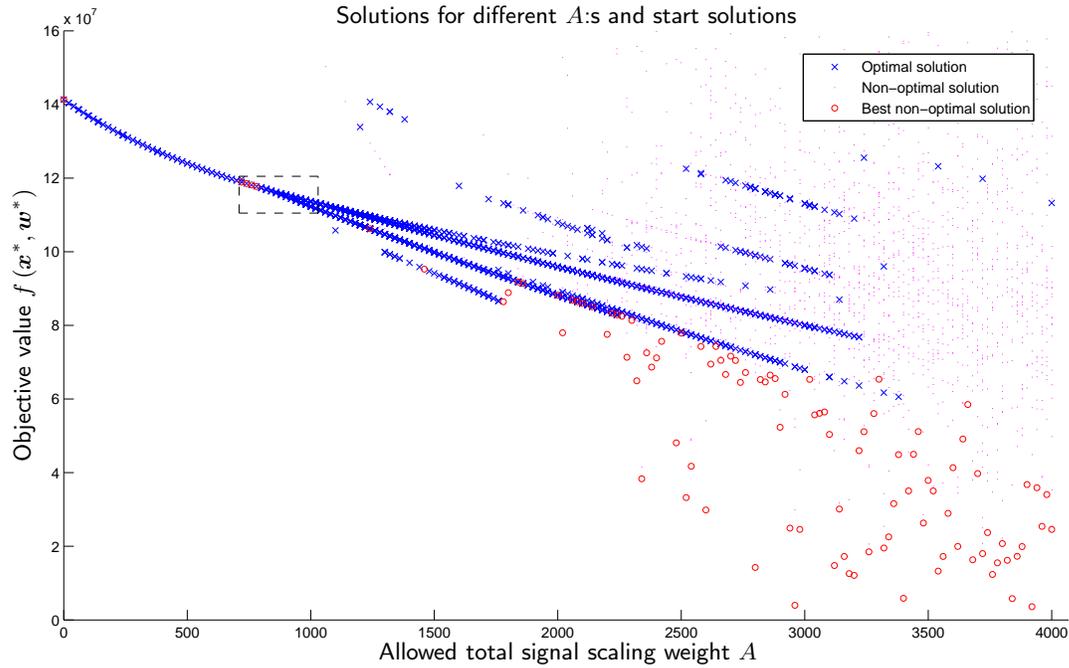
the NMN concept, on the other hand, these numerical experiments give useful information.

## 5 Numerical experiment

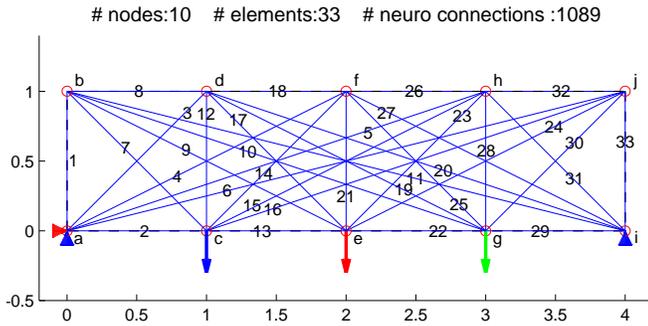
In this example we want to design a 5 by 2 node truss bridge. This example is large enough to show how an active truss can have adaptive stiffness properties and react to a variable load environment, but still small enough to be solvable by Matlab on an ordinary PC in no more than a few minutes. Here we model a load moving along the bridge as a series of load cases in a semi-static way.

The problem setup, i.e. the ground structure, load scenario, and mechanical constraints, is shown in Figure 6. The ground structure consists of all possible non-overlapping elements and neural connections between all pairs of elements, i.e. the neural neighborhood is the whole truss. This gives 33 elements and 1089 neural connections in the ground structure. Further data of the problem are  $V = 0.01$ ,  $E = 1$  and all three load cases have unit magnitudes.

We want to know what happens with the optimal truss when we allow actuation, and then how a further increase in the total signal scaling  $A$  affects the solution. Therefore, we solve a series of problems, where the problem setup is the same except for the upper bound on signal scaling  $A$ , which is gradually increased. Moreover, to deal with the non-convex nature of the problem at hand it is solved several times for each  $A$ , but with different initial solutions  $(\mathbf{x}^{(0)}, \mathbf{w}^{(0)})$ . The initial solutions are randomly chosen but such that they always fulfill the constraints as equalities, i.e.,  $(\sum_{i=1}^m x_i = V$  and  $\sum_{i=1}^m \sum_{j \in \mathcal{N}_i} |w_{ij}| = A)$ . The solutions obtained are checked against optimality criteria to determine if they are at least local optima or if the algorithm has failed to



**Fig. 5:** The 5 by 2 node truss bridge problem has been solved for  $A$  increasing from 0 to 4000 in steps of 20. For each  $A$  the problem is solved 50 times with different start solutions. Thus, the problem is solved 10050 times. The required solution time was around 250 hours on an ordinary PC. Solutions marked by  $\times$  fulfil optimality conditions. Red  $\circ$  and pink  $\cdot$  denote non-optimal solutions, with  $\circ$  marking the best objective value for each  $A$ .



**Fig. 6:** The ground structure used in this test consists of all possible non-overlapping elements (33) connecting the 10 nodes and neural connections between all pairs of elements and between the sensor and neuron in each single element (1089). In the picture the neural network is not plotted because it would make a very indistinct plot.

converge. The results are shown in Figure 5, where the value of the objective function for each solution is plotted against the total signal scaling  $A$ . Each mark in the figure is the result from one single run. For  $A$  above 720, non-convergent solutions starts appearing. For some of these the iteration history was investigated: a typical behavior is that a local monotone descent of the objective function value is suddenly interrupted by a substantial increase after which it is again descending; this behavior

is repeating it self. It was found that the adaptive stiffness matrix was singular close to where the descent was interrupted.

To point out some further properties of the results, in the following subsections we present some specific designs, starting at  $A = 0$  and continuing with increasing  $A$ . The focus is on changes in the neural coupling since the mechanical design is essentially the same for all convergent solutions. The extreme change in a non-zero cross-sectional area that occurs between  $A = 0$  and  $A = 1760$  is 19%, but for most bars this change is only a few percent

### 5.0.1 Optimal trusses for $0 \leq A \leq 700$

For reasons of comparison we begin by looking at an unactuated classical truss, corresponding to  $A=0$ . The optimal design is shown in Figure 7(a), which is a classical solution for a passive truss. Figure 7(b) shows what happens when some actuation is allowed, i.e.  $A > 0$ . The first observation is that the mechanical design is essentially the same for the active truss as for the unactuated truss. Second, the signal power is symmetrically concentrated to two bars which act as both sensors and actuators. We call this design *type I*. To get some understanding of this behavior, study the energy relations in Section 3.3. We see from (14) that the energy spent in actuation depends on the material volumes of both the sensing and actuating elements. Thus, by using the largest element both

as sensing and actuating element, maximal actuation energy can be used for a given  $A$ . This design is the only one we found to be an optimal solution for  $A \leq 700$ . It is also clear that for increasing  $A$ s, the objective value decreases along a smooth curve, i.e. the truss become stiffer.

### 5.0.2 Optimal trusses for $700 < A \leq 1000$

As the available signal scaling  $A$  increases, other types of designs begin to show up as local optimum solutions. In Figure 8 a part of Figure 5 is zoomed in to clearly show the details.

Three phenomena can be observed. First, at  $A = 720$  a new line, related to a new local optimal solution is found. This new design, called *type II* is worse than type I designs, but with increasing  $A$  it becomes a more efficient way to use the available signal scaling. Second, the type I designs transform to a new design, *type Ib*, still based on type I design, but new neural connections have been added. These new connections are, as the existing, direct feedback of a type having negative scaling factor. Third, after the transition from type I to type Ib, the line is divided in two lines, where a slightly different design, *type Ic*, appears. Figure 9 shows the four different designs discussed here.

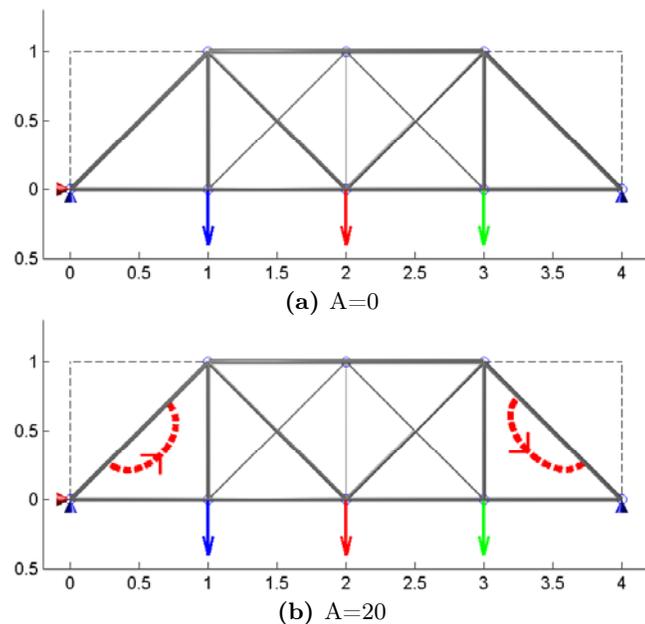
### 5.0.3 Optimal trusses for $1000 < A \leq 3000$

As  $A$  increases, the different designs become better with more available actuation energy and the difference in per-

formance between different designs becomes more pronounced. One interesting observation is the new best design appearing first at  $A=1100$  and then disappearing again at  $A = 1760$ , which can be seen in Figure 5. We call this design *type III* and it is plotted in Figure 10, where it can be seen that also this design undergoes a transition with new connections appearing as  $A$  is increasing. This is the first design where positive scaling appears. Instead of counteracting the natural deformation in an element it is amplified, and in that way larger signals from that element are possible. Furthermore, we see a new design, called *type IV*, rather similar to the type II design. For the same  $A$  it is slightly inferior than type II designs, but it is found for considerable larger  $A$ , so it is the overall best performing optimal design we found in the test. So far none of the designs studied here have a mechanical design considerably different from the unactuated truss.

### 5.0.4 Optimal trusses for $3000 < A$

As  $A$  increases the algorithm more often does not converge to optimal designs and for  $A \gtrsim 3200$  almost no runs manage to converge. An investigation of the iteration history of these runs indicate that this is due to singularities in the stiffness matrices for some designs. This problem is discussed in Section 4.2. However, an interesting observation is that in this region there exist designs that fulfil the constraints, but have neither converged nor fulfil optimality conditions, and still have objective values far better than the locally optimal designs shown earlier in this section.

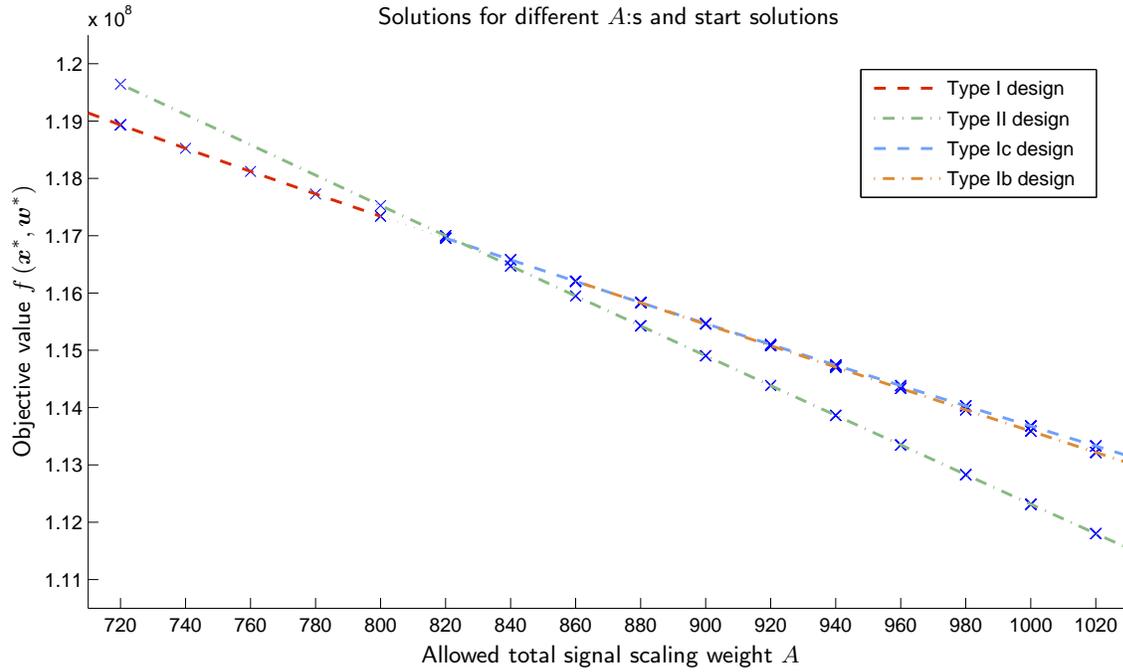


**Fig. 7:** Figure (a) shows the solution for an ordinary unactuated truss exposed to the same load scenario as the active truss designed in this example. Figure (b) shows the solution when some actuation energy is allowed.

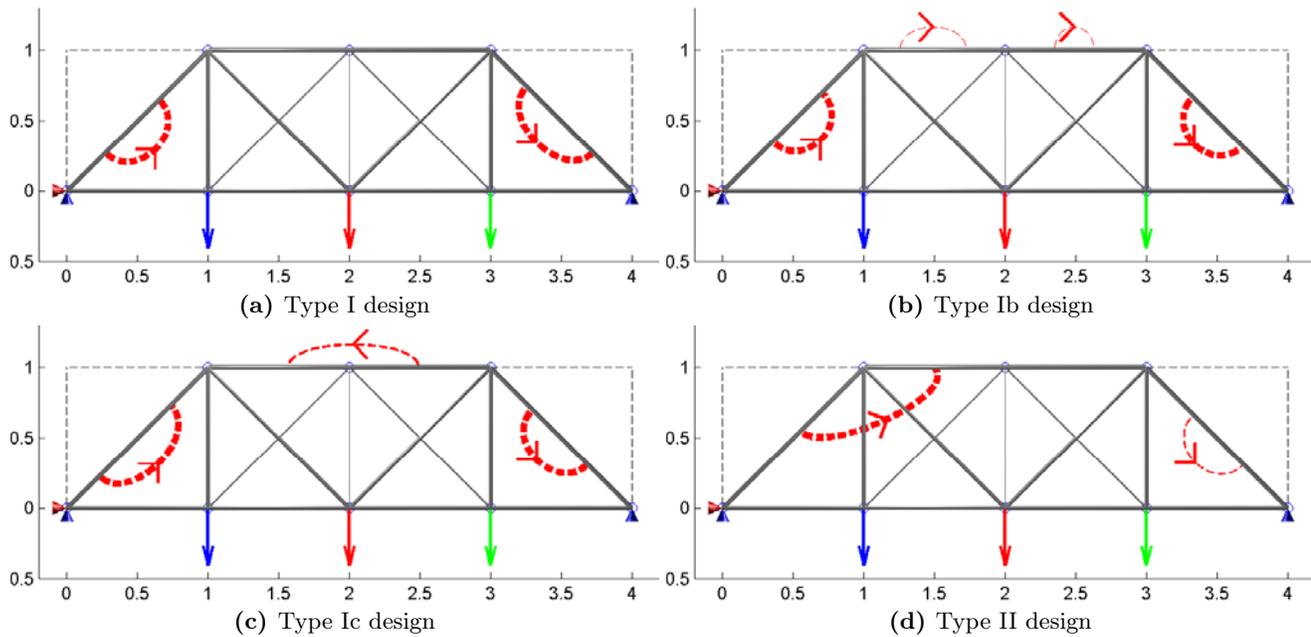
## 6 Summary and conclusions

In this paper we present the general concept of Neuro Mechanical Networks (NMNs). In an NMN a mechanical network, e.g., a truss or a frame, is provided with actuators and sensors to affect and measure its mechanical state. The actuators are controlled by a neural network, receiving input from the sensors. In this way, an NMN can autonomously react to changes in its environment. After a survey of the ideas behind the NMN concept, the function of an NMN and possible future use of the technique, we focused our attention on a type of NMN consisting of what can be called *active trusses*. These are trusses with a one-layer neural network superimposed onto it.

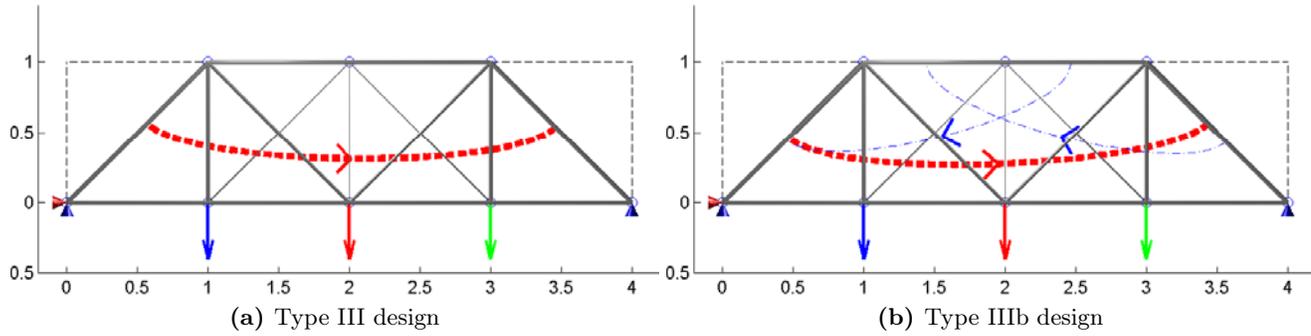
We have established a state model for active trusses, which includes both mechanical and control variables. Using the state equation the active truss configuration problem is stated as a topology optimization problem. This formulation makes it possible, in a simultaneous and automatic process, to design the mechanical topology, actuator, and sensor placement, and perform neural network training for active trusses.



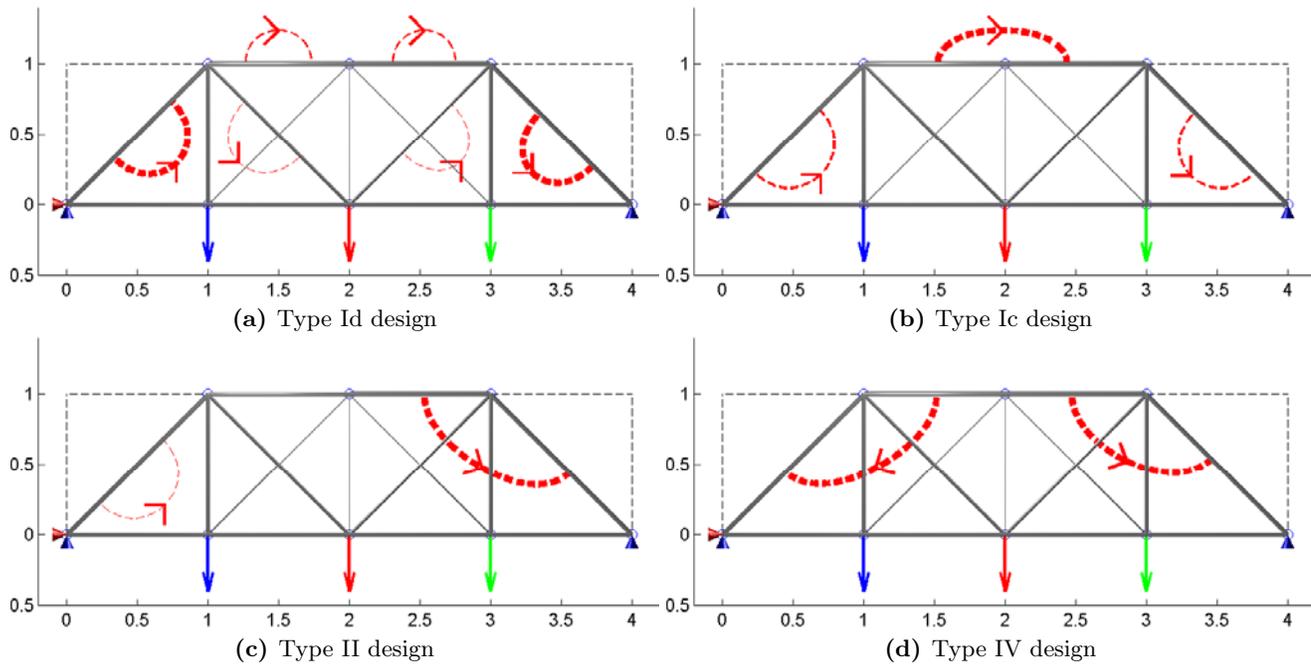
**Fig. 8:** This is a part of Figure 5 zoomed in. We can clearly see how different design families are represented by different lines. Furthermore, it is clear that a type of design that is the most effective for one  $A$  may not be the most effective for another  $A$ . It is also possible to see how a design transforms and divides into two rather similar designs.



**Fig. 9:** These are the four different designs found in the neighborhood of  $A = 800$ . Figure (a) shows a type I design, which is the only design found for  $20 < A < 700$ . Between  $A = 800$  and  $A = 820$ , a type I design transforms to types Ib and Ic, shown in Figure (b) and (c). The original neural connections are still intact but one or two connections have been added. Figure (d) shows a completely new design, first found for  $A = 720$ .



**Fig. 10:** This is the type III design that suddenly appears at  $A = 1100$  and then equally suddenly disappears at  $A = 1760$ . In between, it has undergone a transition where two more connections have appeared. This is the first design with positive signal scaling found in this test. When a compressed element sends a signal with positive scaling it means that the natural deformation in the receiving element is amplified instead of counteracted.



**Fig. 11:** These are four different local optimal designs found for  $A = 1760$ . In Figure (a) it is seen that type I designs have two additional connections, still just as direct feedbacks. Figure (b) shows how the ratio between the different scaling weights changes with increasing  $A$ . Figure (d) shows the design representing the best optimal solution found in this test.

To test the NMN concept, the model and the optimization formulation, some numerical examples have been solved. The results show that NMNs have improved stiffness properties compared to classical passive structures. It also shows how families of active trusses, corresponding to different local optima and with different properties, are suggested. However, convergence problems occurred for large available actuation energy. The algorithm failed to converge to optimal solutions. The problem was traced to singularities in the adaptive stiffness matrix. Improved formulations that take this into account are needed.

Tanaka M., Hanahara K., 1991, Motion Planning of Truss-Type Robot: Acquisition and Representation of Criterion by Modular Neural Network, Proceedings  
 Utku S., Ramesh A. V., Das S. K., Wada B. K., Chen G. S., 1989, Control of a slow moving space crane as an adaptive structure, Proceedings of AIAA/ASME/ASCE/AHS/ASC 30th Structures, Structural Dynamics and Materials Conference. pp. 1119-1126.

---

## References

- Bazaraa M. S., Sherali H. D., Shetty C. M. (1993) *Nonlinear Programming, Theory and Algorithms*, second edition, John Wiley and Sons, Inc.
- Begg D. W., Liu X., 1998, On simultaneous optimization of smart structures – Part II: Algorithms and examples, *Comput. Methods Appl. Mech. Engrg.* 184:25–37
- Bendsøe M. P., Sigmund O., 2003, *Topology Optimization Theory, Methods and Applications*, Springer
- Bendsøe M. P., Olhoff N., Sigmund O., editors, 2006, *IUTAM symposium on Topological Design of Structures, Machines and Materials*, Springer
- Chopra I., 2002, Review of State of Art of Smart Structures and Integrated Systems, *AIAA Journal*, Vol. 40, No. 11, pp 2145–2187
- Cottle R. W., Pang J.-S., Stone R. E., *The Linear Complementarity Problem*, Academic Press
- Fleury C., 1989, CONLIN: an efficient dual optimizer based on convex approximation concepts, *Structural and Multidisciplinary Optimization* 1:81–89
- Haykin S., 1999, *Neural Networks, A comprehensive foundation*, Prentice Hall.
- Janocha H. (Ed.), 1999, *Adaptronics and Smart Structures: Basics, Materials, Design and applications*. Springer ISBN 3540614842
- Klarbring A., Petersson J., Torstenfelt B., Karlsson M., 2003, Topology optimization of flow networks, *Computer methods in applied mechanics and engineering*, 192:3909–3932
- Krus P., Karlsson M., 2002, Neuro-mechanical Networks – self-organising multifunctional systems, *PTCM2002* 1:22–44
- Lipson H., Pollack J.B., 2000, Automatic design and manufacture of robotic lifeforms, *Nature*, Vol 406, pp. 974–978
- Liu X., Begg D. W., 1998, On simultaneous optimization of smart structures – Part I: Theory, *Comput. Methods Appl. Mech. Engrg.* 184:15–24
- Sethson M., Krus P., Karlsson M., 2003, Simulation of self organizing structures using Neuro Mechanical Networks, Proceedings of 17th European Simulation Multi-conference
- Sethson M., Karlsson M., Krus P., 2003, Neuro-mechanical networks as architecture for system design, In Proceedings of AAAI-Synthesis symposium at Stanford university, USA. AAAI, March 2003
- Svanberg K., 1987, The method of moving asymptotes—A new method for structural optimization, *International journal for numerical methods in engineering* 24:359–373
- Svanberg K., 1995, A globally convergent version of MMA without linesearch, Proceedings of the First World Congress of Structural and Multidisciplinary Optimization 9–16