8th World Congress on Structural and Multidisciplinary Optimization
June 1-5, 2009, Lisbon, Portugal

A Neuro-Mechanical Shape Memory Device

Carl-Johan Thore and Anders Klarbring

Division of Mechanics, Linköping University, SE-581 83, Linköping
carl-johan.thore@liu.se, anders.klarbring@liu.se

1. Abstract

In this paper we present what we call a Neuro-Mechanical Shape Memory Device (NMSMD), together with a method for configuration of such devices. This work builds on previous efforts dealing with the concept of Neuro-Mechanical Networks, and a NMSMD is here defined as a Neuro-Mechanical Network (NMN) which has the ability to take on certain predefined shapes when subject to external (mechanical or other) stimuli.

From a mathematical point of view, the NMSMD is described by a coupled system of equations governing the behavior of an active truss with a neural network superimposed onto it. It is shown that with certain restrictions on the model parameters and an appropriate neuro-mechanical coupling term, a Liapunov function for the complete system can be found.

In order to configure a NMSMD, we begin by setting up a ground structure and introducing a set of stimuli vectors, load cases, and associated shapes to be taken on by the device. We then solve an optimization problem with the objective of minimizing a measure of the sum of deviations from the desired shapes. As design variables we use the element volumes and the neural network weights. The problem is solved using the MMA algorithm, and some numerical examples to illustrate the idea are provided in the paper.

2. Keywords: Neuro-Mechanical Network, Shape Memory, Active Structures

3. Introduction

The term Neuro-Mechanical Network refers to a system in which the basic structural unit is a multifunctional element. In addition to providing basic structural integrity, such elements may possess means for actuation, sensing, rudimentary signal processing and possibly other performance-linked functions. The concept of NMN was first introduced in [1] and further developed in [2]. More recently, methods from structural optimization was applied in [3].

In its current formulation, NMN is probably best described as a, partly biologically inspired, concept within the fields of smart structures [4], adaptronics [5], or mechatronics [6]. More specifically it might fall under optimization of smart structures, see [7] and in particular [8] which provide a survey of the application of optimization methods to smart structures. As for the biological inspiration, we mention the electromechanical coupling in the heart, see for instance [9], and the combined neural and mechanical model of fish swimming presented in [10].

A NMN might be further classified as a network system, a term which is here used for systems consisting of a large number of similar, simple but active, elements. Network systems have been under study in the field of neural networks [11] since at least the 1940s [12]. Examples of mechanical network systems have been studied within the framework of distributed or cooperative control, see for instance [13], [14] and [15]. An imaginative and interesting idea that could also mentioned in this context is the concept of Claytronics [16].

In this paper we present a special type of NMN which we refer to as Neuro-Mechanical Shape Memory Devices. Such devices are designed to take on certain predefined shape in response to external, mechanical or other, stimuli. In the literature, this is often referred to as (static) shape control, see for instance [17] for a review. While the examples presented in this work are simple, shape control in general will probably find real applications in for instance aerospace engineering [18]. As a recent source of inspiration from this field we mention the work presented in [19] on static shape control of plates bonded with piezoelectric laminae. Although the mechanical structure is different, the governing equations have the same form as our active truss-equation, and the optimization problem is similar. In contrast to our work, however, the mechanical structure is not subject to optimization, and there is only a one-way coupling from the electric control system to the mechanical structure.

From a mathematical point of view, the NMSMD is naturally split in two subsystems: a mechanical and a neural subsystem. In this case, the neural subsystem consists of a recurrent (artificial) neural network, known as a Hopfield network. For such a network, it was shown in [20] and for neurons with continuous activation in [21], that, with appropriate restrictions on the network parameters, a Liapunov function can be found. This means that the network has stable limit points, i.e., the state of the network evolves in time until it reaches a stable equilibrium. For a given network, several equilibrium points may exist, and an important fact is that the location of the equilibrium points can be controlled by altering the parameters of the network. This is the basis for so called associative or content-addressable memories [20, 21, 22].

The remainder of this paper is organized as follows: In section 3.1 we present the state equations and show that a Liapunov function can be found. We then formulate an optimization problem for configuration of the NMSMD in section 3.2. This is followed by two numerical examples in section 4, and the paper ends with some concluding remarks and suggestions for future work in section 5.

3. Theory

3.1. Mathematical model

From a mathematical point of view, the NMSMD is described by a strongly coupled system of non-linear equations governing the behavior an active truss and a superimposed neural network. The truss has m potential elements and n mechanical degrees of freedom. The state variables in our model are the nodal displacements and the neural network control signals, collected in the vectors $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^m$, respectively. The state variables are assumed to be functions of time, and a superposed dot is used to denote time differentiation. For the mechanical model we assume small deformations and linear elasticity. The equations of state for the NMSMD are, see [24] and cf. [25, 11],

$$\boldsymbol{F} = \boldsymbol{A}(\boldsymbol{x})\dot{\boldsymbol{u}} + \boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{B}^T \boldsymbol{f}^a(\boldsymbol{x}, \boldsymbol{v}), \tag{1}$$

$$\boldsymbol{C}\dot{\boldsymbol{v}} = \boldsymbol{\phi}(\boldsymbol{\varepsilon}) + \boldsymbol{W}\boldsymbol{s}(\boldsymbol{x}, \boldsymbol{v}) - \boldsymbol{R}\boldsymbol{v} + \boldsymbol{I}. \tag{2}$$

In Eq. (1), $\boldsymbol{A}(\boldsymbol{x})$ is a damping matrix, $\boldsymbol{x}$ is the vector of element volumes, $\boldsymbol{K}(\boldsymbol{x}) = \boldsymbol{B}^T\boldsymbol{D}(\boldsymbol{x})\boldsymbol{B}$ is the stiffness matrix, in which $\boldsymbol{B}$ is a matrix which relate strains to displacements through $\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{u}$, and the constitutive matrix is given by $\boldsymbol{D}(\boldsymbol{x}) = \operatorname{diag}\{x_i k_i\}$, where k_i are stiffness constants. The vector $\boldsymbol{f}^a(\boldsymbol{x}, \boldsymbol{v})$ contains the active forces, which are projected onto the appropriate degrees of freedom by multiplication by $\boldsymbol{B}^T$ in Eq. (1).

In the neural network equation, i.e. Eq. (2), $\boldsymbol{C}$ is a diagonal matrix with constant, positive entries, $\boldsymbol{\phi}(\boldsymbol{\varepsilon})$ is a vector of outputs from strain sensors mounted on the elements, $\boldsymbol{W}$ is the weight matrix*, $\boldsymbol{s}(\boldsymbol{x}, \boldsymbol{v})$ is the vector of outputs from the artificial neurons, and $\boldsymbol{R} = \operatorname{diag}\{1/R_i\}$, where R_i are positive constants. External inputs to the combined neuro-mechanical system are $\boldsymbol{F}$, a vector of forces applied to the nodes, and $\boldsymbol{I}$, a vector of non-mechanical stimuli. An interpretation of Eqs. (1) and (2) on the element level is given in Fig. 1.

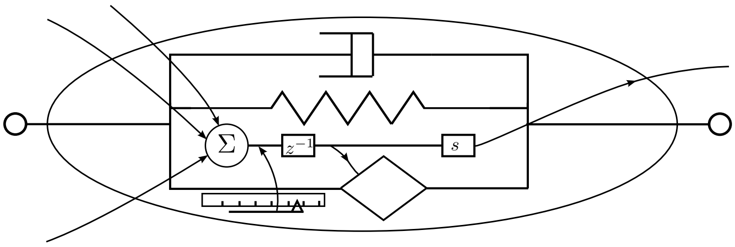

Figure 1: Schematic view of a typical element. z^{-1} denotes a unit time delay. When the elements are assembled to form a truss, the resulting network is governed by Eqs. (1) and (2).

*Note that the neural network weights should not be interpreted as passive components, e.g. resistors. If so, then the non-zero entries of the matrix $\boldsymbol{R}$ would be functions of the weights. For an implementation where the weights are active components, see [23]. Another possibility is of course to interpret the neurons as software entities implemented on microcontrollers attached to the elements.

The entries of the vectors $\boldsymbol{f}^a(\boldsymbol{x},\boldsymbol{v})$ and $\boldsymbol{s}(\boldsymbol{x},\boldsymbol{v})$ have the following structure:

$$\begin{aligned}f_i^a(x_i,v_i) &= x_i\beta_i\tanh(a_iv_i)\\ s_i(x_i,v_i) &= x_i\varphi_i(v_i) = x_i\tanh(a_iv_i),\end{aligned}$$

where β_i and a_i are constants. Scaling with the element volume ensures that elements that vanish in the optimization produce no output. The hyperbolic tangent limits the output from the neurons and the active forces, respectively, since $\tanh(\cdot)\in(-1,1)$. In particular, the free stroke of element i is limited by β_i/k_i.

Inspired by the work of Hopfield [21], we like to find a Liapunov function for Eqs. (1) and (2). The following is a candidate Liapunov function:

$$L(\boldsymbol{u},\boldsymbol{v}) = \alpha(\Pi(\boldsymbol{u}) + \boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{f}^a(\boldsymbol{x},\boldsymbol{v})) + E(\boldsymbol{v}),$$

where α is a positive constant, and

$$\begin{aligned}\Pi(\boldsymbol{u}) &= \frac{1}{2}\boldsymbol{u}^T\boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} - \boldsymbol{u}^T\boldsymbol{F},\\ E(\boldsymbol{v}) &= -\frac{1}{2}\boldsymbol{s}^T(\boldsymbol{x},\boldsymbol{v})\boldsymbol{W}\boldsymbol{s}(\boldsymbol{x},\boldsymbol{v}) + \sum_{i=1}^{m}\frac{1}{R_i}\int_0^{s_i(x_i,v_i)}\varphi_i^{-1}\Big(\frac{s}{x_i}\Big)\mathrm{d}s - \boldsymbol{s}^T(\boldsymbol{x},\boldsymbol{v})\boldsymbol{I}.\end{aligned}$$

Given that $\boldsymbol{K}(\boldsymbol{x})$ and $\boldsymbol{W}$ are symmetric, $\boldsymbol{A}(\boldsymbol{x})$ is positive definite, $\boldsymbol{C}$ and $\partial\boldsymbol{s}/\partial\boldsymbol{v}$ are positive definite diagonal matrices, and

$$\boldsymbol{\phi}(\boldsymbol{\varepsilon}) = -\alpha\frac{\partial\boldsymbol{f}^a}{\partial\boldsymbol{s}}\boldsymbol{\varepsilon} = -\alpha\,\mathrm{diag}\,\{\beta_i\}\,\boldsymbol{\varepsilon},$$

it is readily shown that

$$\frac{\partial L(\boldsymbol{u},\boldsymbol{v})}{\partial t} = -\dot{\boldsymbol{u}}^T\boldsymbol{A}(\boldsymbol{x})\dot{\boldsymbol{u}} - \dot{\boldsymbol{v}}^T\boldsymbol{C}\frac{\partial\boldsymbol{s}}{\partial\boldsymbol{v}}\dot{\boldsymbol{v}} \le 0$$

holds, with equality only if $\dot{\boldsymbol{u}}=\boldsymbol{0}$ and $\dot{\boldsymbol{v}}=\boldsymbol{0}$. Together with the fact that $\boldsymbol{K}(\boldsymbol{x})$ is positive definite so that L is coercive, stability in the Liapunov sense is thus guaranteed.

3.2. Optimization problem

The goal of the optimization is to create a device that for a given external stimuli will take on a certain shape, here defined as a set of nodal coordinates. To this end, we introduce a set of stimuli vectors $\boldsymbol{I}^\theta$, $\theta=1,\dots,\Theta$, a set of corresponding target displacements $\boldsymbol{u}_{\mathrm{tar}}^\theta\in\mathbb{R}^{p(\theta)}$, and a set of loads $\boldsymbol{F}_l^\theta$, $l=1,\dots,L(\theta)$, for each stimuli. Loads may be seen as external stimuli, or used to ensure mechanical stability of the resulting structure.

In the optimization, a subset of the element volumes and the neural network weights are allowed to vary. For the element volumes we use a SIMP-like interpolation scheme and write

$$x_i = \begin{cases}\rho_i^q\bar{x}_i & \text{if } i\in\mathscr{X}\\ \bar{x}_i & \text{otherwise,}\end{cases}\tag{3}$$

where $q>1$ and $\bar{x}_i$ are constants and $\mathscr{X}$ is the set of indices corresponding to the volumes subject to optimization. The variables ρ_i should satisfy $0\le\rho_i\le1$. For the neural network weights we use

$$W_{ij} = W_{ji} = \begin{cases}\omega_{ij}\overline{W} & \text{if } j\in\mathscr{N}_i \text{ and } i\in\mathscr{N}_j\\ 0 & \text{otherwise,}\end{cases}$$

where $\overline{W}$ is a constant, and $\mathscr{N}_i$ is the set of indices corresponding to elements in some neighborhood of element i. The variables $\omega_{ij}\in[-1,1]$, which we refer to as active weights, are used as design variables, and are collected in a vector $\boldsymbol{\omega}\in\mathbb{R}^K$. In this paper we work with a nested version of the optimization problem, so in the following the state variables are considered as functions of the design, where a design is characterized by a pair of vectors $(\boldsymbol{\rho},\ \boldsymbol{\omega})$.

As a suitable objective in the optimization, we consider the following measure of the deviation from the desired shapes:

$$\frac{1}{2}\sum_{i=1}^{N}\big(h_i(\boldsymbol{x}(\boldsymbol{\rho}),\boldsymbol{W}(\boldsymbol{\omega}),\boldsymbol{I}^{\theta_i},\boldsymbol{F}_{l_i}^{\theta_i},\boldsymbol{u}_{\mathrm{tar}}^{\theta_i})\big)^2,$$

where

$$N = \sum_{\theta=1}^{\Theta} p(\theta)L(\theta),$$

and

$$h_i(\boldsymbol{x}(\boldsymbol{\rho}), \boldsymbol{W}(\boldsymbol{\omega}), \boldsymbol{I}^{\theta_i}, \boldsymbol{F}^{\theta_i}_{l_i}, \boldsymbol{u}^{\theta_i}_{\text{tar}}) = \boldsymbol{c}^{\theta_i}_{r_i}\boldsymbol{u}(\boldsymbol{x}(\boldsymbol{\rho}), \boldsymbol{W}(\boldsymbol{\omega}), \boldsymbol{I}^{\theta_i}, \boldsymbol{F}^{\theta_i}_{l_i}) - (u_{\text{tar}})^{\theta_i}_{r_i}$$

where $\boldsymbol{c}^{\theta_i}_{r_i}$ are row vectors with ones in the positions corresponding to the degree of freedom of interest and zeros in all other places. The indices θ_i, r_i and l_i depend on i in such a way that we obtain a correct pairing of all quantities.

In addition to upper and lower bounds on $\boldsymbol{\rho}$ and $\boldsymbol{\omega}$, the number of elements are restricted by introducing the constraint function

$$g(\boldsymbol{\rho}) = \sum_{i\in\mathscr{X}} \rho_i - M,$$

where M is a constant. It should be noted that this constraint, together with appropriate loads, is necessary to make the interpolation scheme defined in (3) meaningful. Without this constraint, intermediate values of the ρ_i:s are no less efficient than the extremal values, and the power law approach will not work as intended, i.e, to drive the variables towards 0 or 1.

For algorithmic purposes, in addition to the design variables we introduce the auxiliary variables y_i, $i = 1, \ldots, 2N+1$. The first $2N$ y_i:s allow us to state the optimization problem in a form suitable for least squares problems [26], while y_{2N+1} is used to penalize violation of the volume constraint. The optimization problem for configuration of the NMSMD now reads as follows:

$$\begin{cases} \min\limits_{(\boldsymbol{\rho},\,\boldsymbol{\omega},\,\boldsymbol{y})} \ \dfrac{1}{2}\displaystyle\sum_{i=1}^{2N} y_i^2 + cy_{2N+1} + \gamma\psi(\boldsymbol{\omega}) \\ \text{s. t.} \begin{cases} h_i(\boldsymbol{x}(\boldsymbol{\rho}), \boldsymbol{W}(\boldsymbol{\omega}), \boldsymbol{I}^{\theta_i}, \boldsymbol{F}^{\theta_i}_{l_i}, \boldsymbol{u}^{\theta_i}_{\text{tar}}) - y_i \leq 0 & i = 1, \ldots, N \\ -h_i(\boldsymbol{x}(\boldsymbol{\rho}), \boldsymbol{W}(\boldsymbol{\omega}), \boldsymbol{I}^{\theta_i}, \boldsymbol{F}^{\theta_i}_{l_i}, \boldsymbol{u}^{\theta_i}_{\text{tar}}) - y_{i+N} \leq 0 & i = 1, \ldots, N \\ g(\boldsymbol{\rho}) - y_{2N+1} \leq 0 \\ \boldsymbol{0} \leq \boldsymbol{\rho} \leq \boldsymbol{1},\ -\boldsymbol{1} \leq \boldsymbol{\omega} \leq \boldsymbol{1},\ \boldsymbol{y} \geq \boldsymbol{0}, \end{cases} \end{cases}$$

where $\boldsymbol{1}$ denotes a vector of ones of appropriate size, and c is a constant. In this problem we have added a neural network complexity penalty term $\gamma\psi(\boldsymbol{\omega})$, where

$$\psi(\boldsymbol{\omega}) = \frac{1}{K}\sum_{k=1}^{K} \omega_k^2,$$

and γ is a constant. The motivation for this term is that weights that do not contribute to the improvement of the design should become small, resulting in a less complex design and possibly improve the convergence properties of the optimization solver (In the field of neural networks, complexity regularization is used to counter problems with overfitting which may result in poor generalization to new data [11]).

In each iteration of the optimization procedure, we need to solve $\sum_{\theta=1}^{\Theta} L(\theta)$ state problems of the form

$$\begin{aligned} \boldsymbol{F}^\theta_l &= \boldsymbol{A}(\boldsymbol{x}(\boldsymbol{\rho}))\dot{\boldsymbol{u}}^\theta_l + \widetilde{\boldsymbol{K}}(\boldsymbol{x}(\boldsymbol{\rho}))\boldsymbol{u}^\theta_l + \boldsymbol{B}^T\boldsymbol{f}^a(\boldsymbol{x}(\boldsymbol{\rho}), \boldsymbol{v}^\theta_l) \\ \boldsymbol{C}\dot{\boldsymbol{v}_l}^\theta &= \boldsymbol{\phi}(\boldsymbol{B}\boldsymbol{u}^\theta_l) + \boldsymbol{W}(\boldsymbol{w})\boldsymbol{s}(\boldsymbol{x}(\boldsymbol{\rho}), \boldsymbol{v}^\theta_l) - \boldsymbol{R}\boldsymbol{v}^\theta_l + \boldsymbol{I}^\theta, \end{aligned}$$

where, in contrast to the previously stated equations, we here use a perturbed stiffness matrix $\widetilde{\boldsymbol{K}}(\boldsymbol{x}(\boldsymbol{\rho})) = \boldsymbol{B}^T(\boldsymbol{D}(\boldsymbol{x}(\boldsymbol{\rho})) + \boldsymbol{D}(\delta\boldsymbol{1}))\boldsymbol{B}$, where δ is a small number. This is done in order to avoid situations where the stiffness matrix become singular during the optimization. The state problems are solved using an explicit Runge-Kutta method as implemented in the Matlab routine `ode45`. An alternative that was also tried is to solve for the equilibrium points directly using Newtons method with line search. In practice, however, this was found to be a less effective approach.

The solution scheme is outlined in Fig. 2. Given an initial design, the first step is to solve the state problems in the way described above. We then compute objective and constraint function values together with respective sensitivities, the latter calculated using an adjoint analytical method. Next, the MMA-subproblem is generated and solved using a primal-dual interior point method to obtain a new

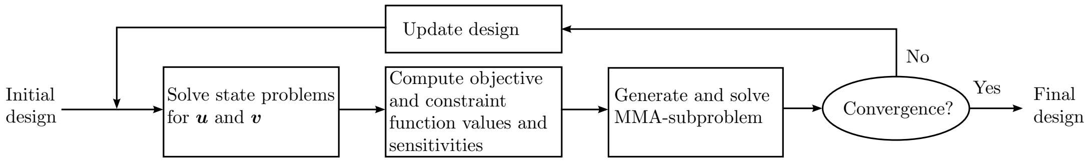
Update design
Initial design
Solve state problems for $\boldsymbol{u}$ and $\boldsymbol{v}$
Compute objective and constraint function values and sensitivities
Generate and solve MMA-subproblem
Convergence?
No
Yes
Final design

Figure 2: Outline of the solution scheme.

design. The procedure is terminated if the change in the design between two consecutive iterations gets below a certain value and the design is feasible with respect to all constraints.

4. Numerical Examples

4.1. A small example

In this example we start with the smallest possible ground structure as shown in Fig. 3. There are six elements and, taking account of the symmetry of the weight matrix, $\boldsymbol{W}(\boldsymbol{\omega})$, 18 active weights. The black circles indicate the range of the direct signal connections. The objective is that the y-coordinate of node four should be 0.55 when the device is subject to an input stimuli applied to element number one and a constant force applied at node number four. The volume of element one is fixed (it was noted that if this was not so, the solver often ended up in a local minima where element number one disappeared and all available material was used on elements three and five in order to obtain a small displacement for node four). An example solution is shown in the top left plot in Fig. 4. The active force of element three produce an elongation of that element which pushes node number four upwards. The wires show the relative importance of the signals in the neural network, i.e. $W_{ij}s_j$, with arrows indicating directions. The top right plot shows the time evolution of the mechanical state. The structure begins by making a downwards dip, but turn upwards again after a short period of time and eventually reaches the state shown in the top left plot. Finally, time histories for displacements and control voltages are shown in the bottom plots of Fig. 4.

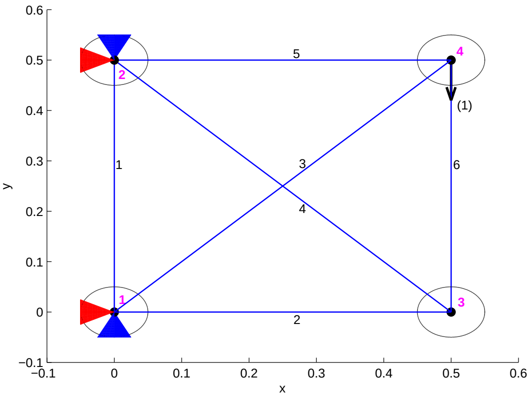

Figure 3: Ground structure. Black circles indicate range of direct signal connections. In the first example a constant force is applied to node number four and an input stimuli is applied to element number one.

4.2. Inverting the mechanical response

In this subsection we use the same ground structure as in the previous example. The y-coordinate of node number four should remain fixed at y = 0.5, for all stimuli and loads, while the x-coordinate should be 0.55 when no stimuli is applied, and 0.45 if a stimuli is applied to element number one, i.e., the stimuli is used to invert the mechanical response. In both cases a constant force is applied to node number four. A third load case with a constant horizontal force applied at node four was used in order ensure that element number five remained large enough to have a mechanically stable structure. The resulting designs are shown in Fig. 5. As can be seen, there are now five elements present, and the signaling patterns are fairly complex.

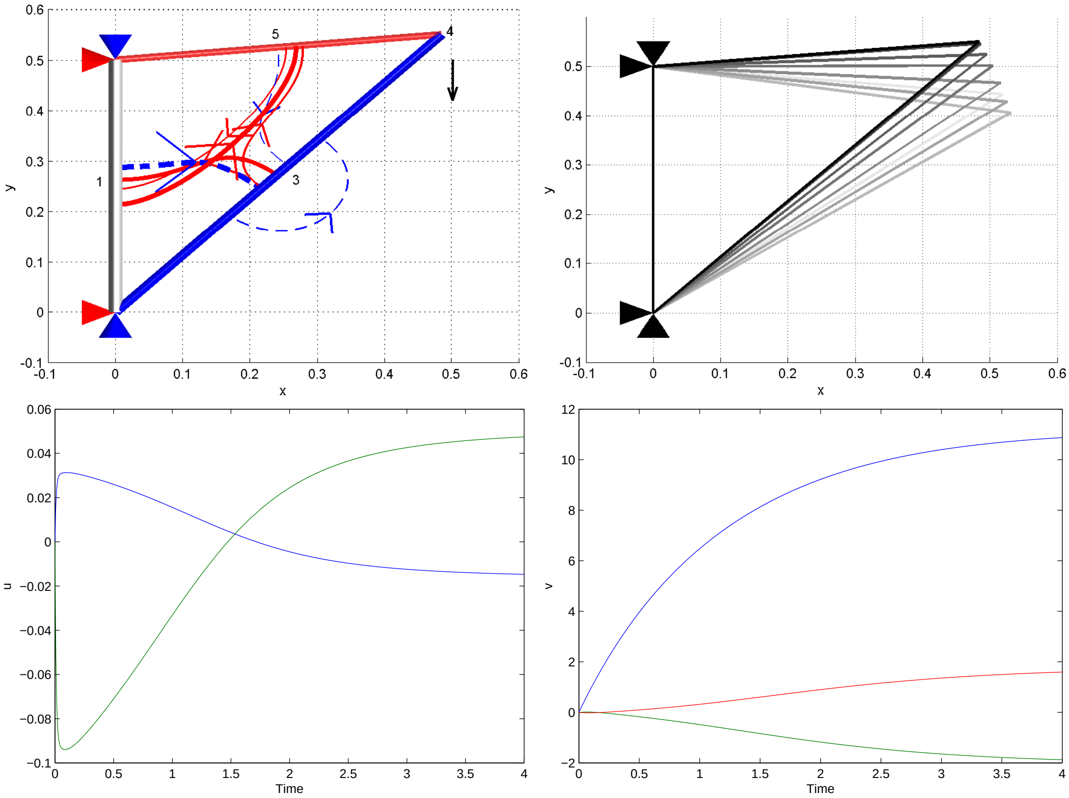

Figure 4: Top left: Final configuration. Red color signifies positive signals and active forces, respectively, while blue signifies the opposite. Note that a positive active force strives contract the corresponding element. Top right: Time evolution of the mechanical state. Bottom left: Time histories for displacements. Bottom right: Time histories for control voltages.

5. Concluding remarks and future work

In this paper we have demonstrated the configuration of what we call a Neuro-Mechanical Shape Memory Device. The configuration process was cast as an optimization problem which was solved using the gradient based MMA-algorithm, and two numerical examples were shown. In order to obtain simple designs, regularization was imposed in the form of a power law and a constraint for the element volumes, and a penalty term for the neural network weights. In the first example, this yielded a simple design with only three elements, whereas in the second example, the resulting design was comparatively more complex. It is possible that the complexity of the neural network could be reduced further by using the 1-norm, which is known to produce sparse models, for penalization. This comes, however, at the cost of introducing a non-differentiable term in the optimization problem.

In the first example, the time evolution of the mechanical state was shown and the trajectory found to agree with intuition, but for larger and more complex structures this might not be the case. In fact, since we only consider equilibrium states in the optimization process we do not have any control over what happens between the initial and final (equilibrium) states. To some extent, however, the use of regularization to obtain simple designs will probably ensure that trajectories are simple. Another potential problem that can be encountered in the design of mechanisms is buckling, which may occur in intermediate states even if elastic stability in the final state is guaranteed, see for instance [27]. Clearly, these issues are of interest for further investigations.

A natural extension of the presented work would be to take into account the effect of large deformations, possibly following along the lines of [27]. Furthermore, interesting results for recurrent neural networks have been presented in for instance [28] and [22]. In the former, a condition on the weight matrix is derived which ensures that for a given external input, the network will evolve to a unique minima regardless of its initial state. In the latter reference, the authors show that there are limits on the set of memories, or equilibrium points, that can be stored in a Hopfield-network with continuous activation. It is possible that these results could be extended to apply to the neuro-mechanical model presented in this paper.

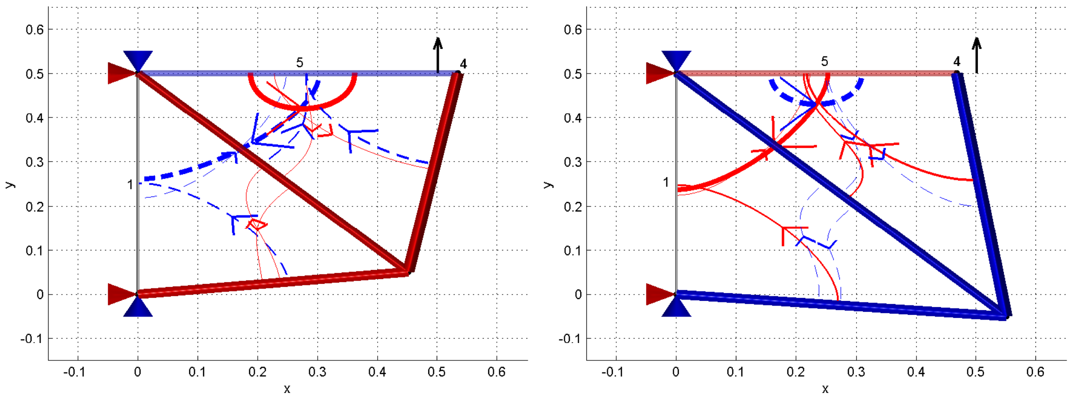

Figure 5: Left: No stimuli is applied but the structure senses the applied force and acts so as to move node four towards (0.55, 0.5). Right: A stimuli is applied to element one, causing the device to displace node four to the left while maintaining its position in the y-direction.

6. Acknowledgments

The authors would like to acknowledge K. Svanberg for providing his implementation of the MMA-algorithm. Thanks also to J. Magnusson for sharing some plot routines.

7. References

[1] M. Sethson and M. Karlsson and P. Krus, Neuro-Mechanical Networks, Self-organizing Multifunctional Systems, *Proceedings of 17th European Simulation Multi-Conference.*, 2002

[2] M. Sethson and M. Karlsson and P. Krus, Neuro-Mechanical Networks as an Architecture for System Design, *Proceedings of AAAI-Synthesis symposium at Stanford University, USA AAAI*, 2003

[3] J. Magnusson and A. Klarbring and M. Sethson, Design and Configuration of Neuro Mechanical Networks, *Structural and Multidisciplinary Optimization*, 37, 335-350, 2009.

[4] B. Culshaw, *Smart Structures and Materials*, Artech house, inc, 1996

[5] H. Janocha, editor, *Adaptronics and Smart Structures*, Springer-Verlag, 1999

[6] H. R. Bishop, editor, *Mechatronic Systems, Sensors and Actuators*, CRC Press, 2008

[7] S. Padula and R. K. Kincaid, Optimization Strategies for Sensor and Actuator Placement, *NASA, Langley Research*, 1999

[8] M. I. Frecker, Recent Advances in Optimization of Smart Structures and Actuators, *Journal of Intelligent Material Systems and Structures*, 14, 207-216, 2003

[9] M. P. Nash and A. V. Panilov, Electromechanical Model of Excitable Tissue to study Reentrant Cardiac Arrhythmias, *Progress in Biophysics & Molecular Biology*, 85, 501-522, 2004

[10] Ö. Ekeberg, A Combined Neuronal and Mechanical Model of Fish Swimming, *Biological Cybernetics*, 69, 363-374, 1993

[11] S. Haykin, *Neural Networks and Learning Machines*, Prentice Hall, 2009

[12] W. S. McCulloch and W. Pitts, A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, 5, 115-133, 1943

[13] H. Kobayashi and E. Shinemura and K. Suzuki, A Distributed Control for Hyper Redundant Manipulator, *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1958, 1992

[14] V. Kumar and N. Leonard and A. S. Morse, editors, *Cooperative Control*, Springer-Verlag, 2005

[15] G. E. Dullerud and R. D'Andrea, Distributed Control Design for Spatially Interconnected Systems, *IEEE Transactions on Automatic Control*, 48, 1478-1495, 2003

[16] S. C. Goldstein and T. C. Mowry, Claytronics: A Scalable Basis for Future Robots, *RoboSphere 2004*, Moffett Field, CA, 2004

[17] H. Irschik, "A Review on Static and Dynamic Shape Control of Structures by Piezoelectric Actuation", *Engineering Structures*, 24, 5-11, 2002

[18] F. Austin and M. J. Rossi and W. Nostrand, Static Shape Control for Adaptive Wings, *AIAA Journal*, 32, 1895-1901, 1994

[19] Z. Kang and L. Tong, "Topology Optimization-Based Distribution of Actuation Voltage in Static Shape Control of Plates", *Computers and Structures*, 86, 1885-1893, 2008

[20] J. J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proceedings of National Academic Science, USA*, 79, 2554-2558, 1982

[21] J.J Hopfield, Neurons with Graded Response have Collective Computational Properties like those of Two-State Neurons, *Proceedings of National Academic Science, USA*, 81, 3088-3092, 1984.

[22] A. Atiya and Y. S. Abu-Mostafa, An Analog Feedback Associative Memory, *IEEE Transactions on Neural Networks*, 4, 117-126, 1993

[23] R. W. Newcomb and J. D. Lohn, Analog VLSI for Neural Networks, *Handbook of Brain Theory and Neural Networks*, M. Arbib (Ed.), Bradford Books, MIT Press, 1995

[24] C.-J. Thore and A. Klarbring, Neuro-Mechanical Mechanisms, *In preparation*, 2009

[25] A. Preumont, *Mechatronics: Dynamics of Electromechanical and Piezoelectric Systems*, Springer, 2006

[26] K. Svanberg, *MMA and GCMMA, versions September 2007*,
http://www.math.kth.se/˜krille/gcmma07.pdf, 2007

[27] M. Stolpe and A. Kawamoto, Design of Planar Articulated Mechanisms using Branch and Bound, *Mathematical Programming*, 103, 357-397, 2005

[28] A. F. Atiya, Learning on a General Network, *Neural Information Processing Systems*, American Institute of Physics, New York, 1988