Helsinki University of Technology

Department of Automation and Systems Technology

Control Engineering Laboratory

Professor Ph.D. Heikki Koivo

Diploma Thesis

Distributed Control of a Deformable System –
Analysis of a Neocybernetic Framework

prepared by

cand. kyb. Michael Sailer

Professor Dr. Tech. Heikki Hyötyniemi

Professor Dr.-Ing. Dr.h.c. Michael Zeitz

March 2006

Preamble

Solemn Affirmation

Hereby I, Michael Sailer, declare on oath that I made the thesis at hand self-dependent and only with the help of the mentioned references.

Espoo, 17th of March 2006

Used Abbreviations

act:	actuator
b:	break
exp:	expected
ext:	external
fem:	Femlab
fo:	first order
g:	gravity
int:	internal
mat:	Matlab
rel:	relative
PCA:	Principal Component Analysis
PCR:	Principal Component Regression
PSA:	Principal Subspace Analysis
SISO:	Single-input-single-output
so:	second order

Contents

Chapter 1

Introduction

It was Norbert Wiener who brought up the term cybernetics when his book *Cybernetics: Or Control and Communication in the Animal and the Machine* [25] was published in 1948. The general idea of cybernetic systems is that observed complex behavior in systems is caused by the dynamics of present actors. These dynamics are a result of interactions and feedback structures among the actors. Per definition, cybernetics can be described as the study of systems and control on an abstracted level. Therefore it constitutes an excellent framework for the combination of control theory, information theory and communication theory and as an interdisciplinary research area it can be applied to different domains like biology or technology [15][14].

Until now, traditional control purposes are mainly implemented in a centralized manner. Hence, the total potential of cybernetic thinking in control theory is not exploited yet, if one thinks of distributed sensor/actuator networks for control purposes. Distributed sensor/actuator networks are gaining attention, but still the coordination in such systems is not completely decentralized as there usually exist central units or sinks for coordination and monitoring purposes [16]. The key question here is how emergent behavior in complex systems can evolve, when only fundamental, decentralized interactions among low level sensor/actuator pairs are realized without explicit knowledge of the global picture. One answer to this question is given by the neocybernetic approach of modeling and controlling complex real-life systems.

In this framework, self-regulation and self-organization can evolve in systems. For

analysis of these phenomena multivariate statistical tools are needed. A system theoretic approach reveals that Hebbian/anti-Hebbian learning results in emergent system behavior, what can, for example, be used for smart data analysis in distributed sensor networks [11]. Even stronger views can be derived from the new neocybernetic approach to "elastic systems" [15]. These systems are determined by strong interconnections with their regarded environment caused by material or information flows from and into the system. It turns out that the new approach can be used for the decentralized control of deformable systems in a distributed sensor/actuator network and this is shown in what follows.

As the neocybernetic thinking is not widespread yet, the general ideas of neocybernetics and the new approach on elastic systems are presented in the following two chapters. After that the knowledge about elastic systems is used to realize a simulation area for the research on the distributed control of deformable mechanical systems with locally acting sensor/actuator pairs. In addition to that the highly iterative simulation process demands a more sophisticated simulation strategy where the strong interconnection between the used simulation tools Femlab and Matlab is restructured. The outcomes of the simulations are presented in the thesis and interpreted.

Chapter 2

About Neocybernetics

2.1 Overview

If one wants to describe the idea of neocybernetics, one can say that this new or extended theoretical framework tries to provide a new approach to describe real life complex systems. Thereby it compensates shortcomings of other existing theories like complexity theory, system theory, control theory or the traditional cybernetics that are also related with the classification and handling of real-life complex systems. Simplified, one can say that cybernetics itself offers a special view on complex systems, while neocybernetics offers a special view on cybernetic systems [13].

Compared to the complexity theory, where structurally complex nonlinear functions are studied as independent entities, the neocybernetic approach concentrates on structurally simple large scale systems. Here, the complexity is a result of system-wide interactions and high-dimensionality of the systems. Control theoretic approaches have their focus more on centralized, individual feedback loops rather than distributed control structures. Here, neocybernetics describes system dynamics with mutual interactions and feedback structures among decentralized lower level actors. As a result self-organization and self-regulation emerge. These mechanisms of emergence are emphasized and researched with the neocybernetic approach. In traditional cybernetic systems the issue of emergence is not addressed yet [13].

This chapter gives a closer look at the intuition of emergence in complex systems. As
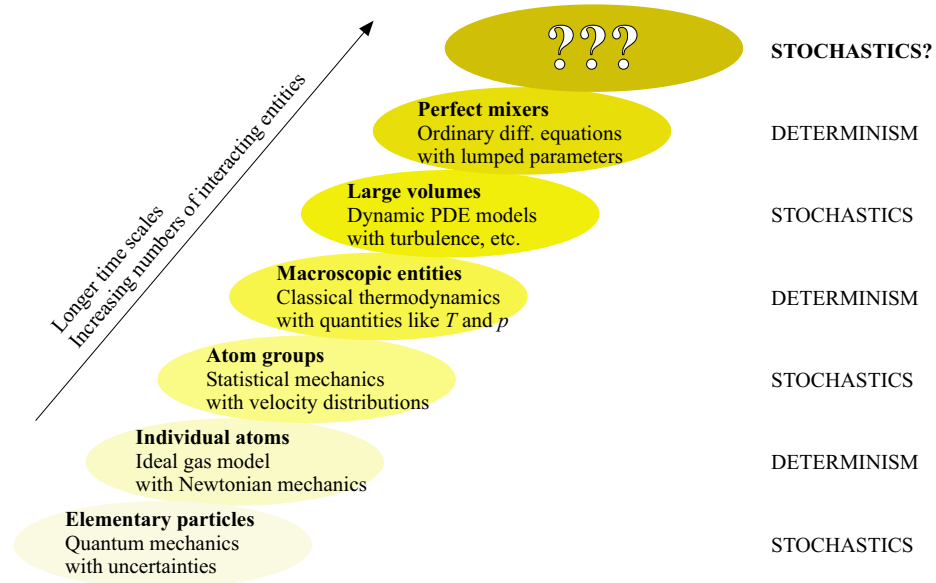
Figure 2.1: Different levels of abstraction while modeling a gaseous system [15].

a result, it is stated out that statistical tools can be applied to capture emergence in neocybernetic systems. Therefore the considered systems have to be stationary and only stable systems are taken into account. Linear structures are applied in order to reach scalability of the results.

2.2 Emergence in Complex Systems

The key concept in complexity theory is emergence, meaning that some higher order phenomena cannot be reduced to the analysis of their individual components. Qualitatively new, unanticipated functionalities may emerge, as a result of the cumulation of simple low-level operations. In these cases the reductionistic modeling approaches collapse, as the "whole" is more than the sum of all parts.

In order to get a better understanding of emergence, explicit examples can be studied first and the common features can be represented in an explicit mathematical framework. An example for emergence are models of gaseous systems in different scales, like presented in Figure 2.1. There it can be seen that emergence solidifies in a way that higher order behavior of a system cannot be described satisfactory as a sum of all lower level interactions [15].

The elementary particles are presenting the system on the lowest level. The orbital models, describing these particles are stochastic. At the atom level, the Newtonian approximative ideal gas model can be taken into account to define atoms as billiard balls. On this level the concepts of velocity and momenta are deterministic. In a larger system, individual collisions of millions of atoms cannot be traced and statistical mechanics becomes the appropriate framework. One step higher, macroscopic and deterministic quantities like pressure, temperature or entropy describe the state of system. In still larger volumes convections and turbulences result out of temperature distributions and statistics are playing an important role again, in order to describe the system. On the highest level, the deterministic modeling of ideal mixers (e.g. cells) with lumped parameters emerges, if complete turbulence is assumed. On that level there are concentrations that need to be taken into account.

In the described hierarchy it is obvious that stochastic and deterministic behavior alternate. The alternation seems to be intuitively right as for example two successive deterministic levels could be merged to one level and no emergence could be realized. It also turns out that the number of interacting entities increases with the level height. On a still higher level of abstraction, statistical tools can describe the system again, if the actual dynamic processes are abstracted. But in order to apply statistical methods the underlying system has to be stationary and stationary signals can only be reached in stable systems. The claim of stability can be explained, as unstable systems would end up in explosion or an exhaustion of resources and therefore extinction. Hence, modeling of neocybernetic systems concentrates on stable systems, where statistical tools can be applied [15], [13].

2.3   Key ideas

As one has a closer look at the basic principles of neocybernetic modeling, there are three areas that have to be discussed, the dynamic balances in systems, the high-dimensionality and the linearity pursuit of modeling.

2.3.1 Dynamic balances

As stated out in the foregoing section, it is the emphasis on the final balance rather than on processes that are finally leading to that balance. Instead of concentrating on nonlinear interactions that are forming the process one is interested in the pattern that emerges.

The researched balance in the system is a balance among tensions that are compensated by internal mechanisms. In practice these dynamic balances are caused by feedbacks. How these feedbacks are realized is not essential, as long as the system can maintain the claim of stability. It also has to be mentioned that the concept of balance can be interpreted in a wider sense, as the balances are defined with respect to the considered variables. One can for example research the derivatives of some other quantities and consider these derivatives to be balanced. Thereby one achieves a constant level of dissipation. Furthermore, the idea of stable systems can be relaxed. Systems can also include oscillating parts, as long as it keeps its integrity and statistical terms can be used to describe the system [15], [10].

2.3.2 High dimensionality

As the real structure of the environment can not be assumed to be known, one has to take measurements of the environmental responses into account in order to build an appropriate model of the considered environment. Compared to conventional complex systems, where the models are structurally complex, the complexity in neocybernetic models is a result of high-dimensional parallel handling of measured information. The considered data is collected in data vectors. With these data vectors the modeling machinery tries to construct appropriate connections among the available data segments. As the data normally turns out to be redundant, one can use multivariate methods [9] and corresponding mathematical tools for a better analysis of neocybernetic models [15].

This bottom-up analysis of data can be carried out by Hebbian neurons, for example, and is presented later on in Section 2.4 in order to get a better understanding of neocybernetic modeling approaches [12].

To achieve a consistent mathematical framework that is easy enough to provide a simple analysis of complex systems and still captures the essence of the considered neocybernetical models, the following section introduces the aspect of linear modeling.

2.3.3 Linear modeling

It turns out that the idea of linear modeling often offers a rather good match with reality, at least if the nonlinearities in the system are linearizable and smooth. The big advantage of linear modeling is that there exist powerful tools for analysis and also intuitive transfers from a domain field to another can be made easily. With the use of linear structures scalability of models can be reached [10].

A deeper view on the issue of linearity and a detailed argumentation can be found in [15] and [10]. The linear framework can also be relaxed later on if necessary. More on that can be found in [12]. In this work the used application tools for neocybernetic modeling are linear tools.

One example that satisfies the introduced ideas of high-dimensionality and linearity is the modeling of linear Hebbian neuron grids. It turns out that with the proper statistical tools, Hebbian learning results in self-organization and self-regulation. Section 2.4 introduces these ideas and results are interpreted from the neocybernetic point of view.

2.4 Example: Hebbian Neuron Grids

As mentioned above, the modeling of a Hebbian/anti-Hebbian neuron grid can be taken as a prototype of modeling neocybernetic systems. When the neurons are connected together properly, self-stabilization and self-organization takes place in a neuronal system. It turns out that the implementation of these grids for data analysis carries out PSA and if further structural constraints are implemented PCA and PCR are realizable.

The resulting emergent pattern can also be formulated in terms of an optimality cri-

terion (see Section 2.4.5) and the neocybernetic strategy constructs a "mirror image" of the environment, being itself a model of the environment. It captures relevant behavioral patterns as they are manifested in the data.

The following sections introduce the mathematical framework of Hebbian neuron grids and interpretations are made later on. The information can be read in [12] at length and additional information can be found in [10].

2.4.1   Dynamics in a Hebbian Neuron Grid

It is assumed that the d inputs to a neuronal system are collected in the vector u and the vector of neuronal activities in a grid of c neurons is denoted as x. Now, there is a new input sample $u(l)$ during each time constant l and one is interested in the internal neuronal state $x(l)$ given by the neuronal activities.

It cannot be assumed that the internal state changes instantaneously. In the following part a dynamic neuronal structure is introduced, describing the adaptation. For that purpose another time constant t is needed. If a new set of input variables $u(l)$ is applied, t starts from zero and the adaptation of the continuous-time state vector $x_{cont}(t,l)$ goes on, until the reaching of the steady state. The challenge in that kind of system is that only the input vector $u(l)$ is known and one should determine the internal state and the internal system structure. Therefore some assumptions have to be made.

If one assumes that the momentary change in the neuronal activity is linearly proportional to the current neuronal state and also to the input activity, one can model the whole grid of neurons simultaneously, when matrix formulation is used. The dynamic in the grid of neurons is described as follows

$$\frac{dx_{cont}}{dt}(t,l) = Ax_{cont}(t,l) + Bu(l) \tag{2.1}$$

Here, the synaptic weights are collected in the matrices A and B. Matrix A collects the synaptic weights between neurons itself, as matrix B determines the weights between neurons and inputs. In order to make the discussions directly compatible with the algorithmic implementations later on, the continuous (2.1) can be presented

in discrete-time formulation. The derivative can be approximated with

$$\frac{dx_{cont}}{dt} \approx \frac{x_{cont}(t+h,l) - x_{cont}(t,l)}{h} \tag{2.2}$$

After defining a discrete-time activity vector as

$$x(\kappa, l) = x_{cont}(\kappa h, l) \tag{2.3}$$

(2.1) can be written in discrete-time as

$$x(\kappa + 1, l) = x(\kappa, l) + hAx(\kappa, l) + hBu(l) \tag{2.4}$$

If the system is asymptotically stable, the state will finally converge to

$$\bar{x}(l) = lim_{\kappa \to \infty}\{x(\kappa, l)\} \tag{2.5}$$

Here it has to be assumed that the adaptation of the neuronal grid is faster than the change of the input data $u(l)$.

2.4.2 Modeling of a Hebbian Neuron

Based on the linear Hebbian neuron model, complex structures could already be developed as can be read in [6] or [18], for example. On the other hand, models have become increasingly complex, what makes it difficult to find efficient methods for the analysis.

Basis for the following mathematical considerations are neurophysiological observations that can be ascribed to the physician Donald O. Hebb half a century ago [7]. The Hebbian law can be formulated as follows:

- **Hebbian law.** Synaptic connection between the neuron and an incoming signal becomes stronger if the signal and the current neuron activity correlate with each other.

That simple rule gives a hint, how low-level neuronal functions are connected to fulfill higher-level functionalities and higher order emergence could take place.

According to the Hebbian law, the synaptic weight r_{ab} between neuron a and input b can be calculated as follows

$$r_{ab}(l+1) = r_{ab}(l) + \rho\bar{x}_a(l)u_b(l) \tag{2.6}$$

The factor $\rho > 0$ determines the synaptic dynamics and $\bar{x}(l)$ is the steady-state neural activity for the input $u(l)$. As the synaptic connections can become stronger and stronger without limit, the occurring instability can be eliminated by adding a nonlinear term, named Oja's rule [19]. The nonlinear term makes an analysis of the overall system difficult; stability can also be reached by using a negative linear feedback term additionally. The term can be added in (2.6) and as a result the synaptic weight can be calculated as

$$r_{ab}(l+1) = r_{ab}(l) + \rho \bar{x}_a(l) u_b(l) - \frac{1}{\tau} r_{ab}(l) \tag{2.7}$$

The parameter $\tau > 0$ is the time constant determining the rate of decay. To represent all input-output connections, matrix expression can be used and it follows

$$R(l+1) = R(l) + \rho \bar{x}(l) u^T(l) - \frac{1}{\tau} R(l) \tag{2.8}$$

If one assumes that the second order statistical properties of the data do not change over time and the factor τ is large the steady-state value for the matrix R of synaptic weights can be calculated out of (2.8) as

$$R = \rho \tau E\{\bar{x} u^T\} \tag{2.9}$$

Matrix R is called the covariance matrix of the vectors $\bar{x}$ and u. Covariance matrices can be defined for signals in vectors x and u in the same way. The covariance matrices are decomposed in what follows as

$$R_{\bar{x}\bar{x}} = E\{\bar{x}\bar{x}^T\} = \begin{pmatrix} E\{\bar{x}_1\bar{x}_1\} & \cdots & E\{\bar{x}_1\bar{x}_c\} \\ \vdots & \ddots & \vdots \\ E\{\bar{x}_c\bar{x}_1\} & \cdots & E\{\bar{x}_c\bar{x}_c\} \end{pmatrix} \tag{2.10}$$

and

$$R_{\bar{x}u} = E\{\bar{x}u^T\} = \begin{pmatrix} E\{\bar{x}_1 u_1\} & \cdots & E\{\bar{x}_1 u_d\} \\ \vdots & \ddots & \vdots \\ E\{\bar{x}_c u_1\} & \cdots & E\{\bar{x}_c u_d\} \end{pmatrix} \tag{2.11}$$

Here it has to be mentioned that the expectation values have to be calculated over the time index l as introduced in Section 2.4.1. The matrix form for the expectation values makes it very easy to represent parallel operation and still interaction and adaptation

operations are local. Weights between neurons are determined by the corresponding output and input neurons alone. If it is assumed now that the synapses follow the Hebbian principle, the matrices A and B in (2.1) can be chosen as

$$A = -\mu E\{\bar{x}\bar{x}^T\} \tag{2.12}$$

and

$$B = \mu E\{\bar{x}u^T\} \tag{2.13}$$

Here, factor $\mu > 0$ is a step size factor that is adjusting the time scale in the adaptation algorithm. As can be seen in (2.12) and (2.13), the covariance structures are the same, but the signs are different. The explanation can be derived from a single synapse: Stability in the grid can be maintained, if negative feedback is applied in form of linear dynamic structures. If the covariance matrix A is positive definite, all eigenvalues of A are non-negative and the model (2.1) is stable. In what follows it is assumed that the adaptation of the covariance matrices is much slower than the change in the input data and the neuronal grid dynamics in (2.1) are still much faster.

The minus sign in the structure seems to invert the basic Hebbian law and one can define the anti-Hebbian law as follows:

- **Anti-Hebbian law.** Synaptic connection between two neurons becomes weaker if the neuronal activities correlate with each other.

An interpretation of the Hebbian law is that the effects from prior level are excitatory, as the lateral connections between the same level neurons are inhibitory. The Hebbian/anti-Hebbian structure is carrying out PSA as can be seen in the following section. While Hebbian learning searches for maximum variation directions in the analyzed data, the anti-Hebbian learning results in an organization among the neurons as it implements some kind of competitive learning.

2.4.3   Principal Subspace Analysis

If it can be assumed that the system reaches the stationary state $\bar{x}(l)$, (2.4) holds

$$\bar{x}(l) = \bar{x}(l) + hA\bar{x}(l) + hBu(l) \tag{2.14}$$

and with (2.12) and (2.13) it is

$$\bar{x}(l) = \bar{x}(l) - \mu h E\{\bar{x}\bar{x}^T\}\bar{x}(l) + \mu h E\{\bar{x}u^T\}u(l) \tag{2.15}$$

If one solves for the steady-state, $\bar{x}(l)$ is

$$\bar{x}(l) = \underbrace{E\{\bar{x}\bar{x}^T\}^{-1}E\{\bar{x}u^T\}}_{\phi^T} u(l) \tag{2.16}$$

regardless of factors μ and h, as long as the iteration is stable. The inverse covariance matrix $E\{\bar{x}\bar{x}^T\}^{-1}$ exists, if the variables in x are not linearly dependent.

One can define a static linear mapping between the input $u(l)$ and the steady-state neuronal activity $\bar{x}(l)$ as

$$\bar{x}(l) = \phi^T u(l) \tag{2.17}$$

where the $d \times c$ matrix ϕ is defined as

$$\phi = E\{\bar{x}u^T\}^T E\{\bar{x}\bar{x}^T\}^{-1} \tag{2.18}$$

Because the final neural activity, meaning $\bar{x}(l)$ corresponding to $u(l)$, is not known beforehand, determination of the covariance matrices is an iterative process (see Section 2.4.4). One can see from (2.16) that the final neural states are determined by the covariance matrices and the covariance matrices itself are determined by the steady state respectively.

It can be proved that the stable fixed point for the mapping matrix ϕ spans the principal subspace of the input data [12]. That means that the columns of ϕ are linearly independent combinations of the c most significant eigenvectors of the input covariance matrix $E\{uu^T\}$, corresponding to the largest eigenvalues. Moreover, the variability in x equals the total variance along the c most significant principal component directions in u. More on PCA can be found in appendix F and [9].

As the number of neurons is smaller than the number of inputs, not all variation in the data u can be explained by x, but the largest principal components explain variations in the best possible way, trying to find the minimum for a quadratic cost criterion (see also Section 2.4.5).

Overall, one can say that Hebbian/anti-Hebbian learning results in self-regulation,

what means the resulting system is balanced, and also self-organisation, as the learning carries out PSA and emphasizes the directions in the data with the highest variation. Hebbian/anti-Hebbian learning is used in order to construct models for data analysis purposes.

2.4.4   Algorithmic Implementation

If one has a closer look at the mapping matrix ϕ from (2.18) it can be realized that only the input data $u(l)$ is known, but not the final steady-state $\bar{x}(l)$ that is needed for adapting the covariance matrices. The whole process is highly iterative. But if such analysis is applied for technical purposes, a streamlining of the dynamic iteration process can be done. The outcomes of the dynamic process can directly be employed, as one implements explicit matrix inversions. The used algorithm can be described in two steps:

1. First the covariance estimates have to be updated. Here λ describes the forgetting factor and there holds $0 \ll \lambda < 1$

$$\begin{cases} E\{\bar{x}\bar{x}^T\}(l) = \lambda E\{\bar{x}\bar{x}^T\}(l-1) + (1-\lambda)\bar{x}(l-1)\bar{x}^T(l-1) \\ E\{\bar{x}u^T\}(l) = \lambda E\{\bar{x}u^T\}(l-1) + (1-\lambda)\bar{x}(l-1)u^T(l-1) \end{cases} \qquad (2.19)$$

2. The signals estimate out of (2.16) can now be found as

$$\begin{cases} \bar{x}(l) = E\{\bar{x}\bar{x}^T\}^{-1}(l)E\{\bar{x}u^T\}(l)u(l) \end{cases} \qquad (2.20)$$

Here the covariances can be for example initialized as $E\{\bar{x}\bar{x}^T\}(0) = \epsilon I_c$, where ϵ is a small constant.

As mentioned before, the input neurons are carrying out PSA . It is also possible to implement PCA carried out by the input neurons directly. How the constraints have to be set, in order to achieve that can be read in [12] and is not conducted here.

The introduced Hebbian/anti-Hebbian learning can also be explained in a more abstract sense, what is shortly presented in the following section.

2.4.5   Optimality of Hebbian learning

A general quadratic optimality criterion that has to be minimized, can be presented as follows

$$J(x) = \frac{1}{2}(u - \varphi x)^T W(u - \varphi x) \tag{2.21}$$

Matrix W is symmetric, positive definite and compatible with the vector u. Matrix φ is a $d \times c$ dimensional mapping matrix. In order to search for the unique minimum, the gradient can be expressed as

$$\frac{dJ}{dx}(x) = \varphi^T W \varphi x - \varphi^T W u \tag{2.22}$$

The closed form solution for x is

$$\bar{x} = \left(\varphi^T W \varphi\right)^{-1} \varphi^T W u \tag{2.23}$$

If one wants to search the minimum iteratively with the steepest descent method [24] the formulation can be presented as

$$\begin{aligned}
x(\kappa + 1) &= x(\kappa) - \beta \frac{dJ}{dx}(x(\kappa)) \\
&= x(\kappa) - \beta \varphi^T W \varphi x(\kappa) + \beta \varphi^T W u
\end{aligned} \tag{2.24}$$

If one compares this equation to (2.4) one can see a certain similarity, if the different applied sets of data l are neglected. With (2.12), (2.13) and (2.17) the expressions are identical if

$$\begin{cases}
\beta \varphi^T W \varphi &= -hA = \mu h E\{\bar{x}\bar{x}^T\} = \mu h \phi^T E\{uu^T\}\phi \\
\beta \varphi^T W &= hB = \mu h E\{\bar{x}u^T\} = \mu h \phi^T E\{uu^T\}
\end{cases} \tag{2.25}$$

These expressions are valid if

$$\begin{cases}
\beta &= \mu h \\
\varphi &= \phi \\
W &= E\{uu^T\}
\end{cases} \tag{2.26}$$

The optimality criterion from (2.21) results with (2.26) in

$$J(x) = \frac{1}{2}(u - \phi x)^T E\{uu^T\}(u - \phi x) \tag{2.27}$$

As a result one can conclude that Hebbian/anti-Hebbian learning can also be formulated in the framework of optimization. The algorithm in (2.4) can be interpreted as

the descent gradient algorithm, trying to find the minimum for the introduced cost criteria. One can state out that in the neocybernetic point of view not the process itself represented by the iteration is important (process view), but the final emerging pattern (pattern view) that results out of the iteration.

The algorithm tries to minimize the difference between u and ϕx. With the weighting matrix $W = E\{uu^T\}$ it can be said that errors in the directions with high variations are emphasized. That makes the algorithm robust if one searches for the principal components while suppressing noise as well.

The next chapter is introducing a new neocybernetic approach called elastic systems. It turns out that Hebbian/anti-Hebbian learning is a special case of evolutionary adaptation performed by elastic systems in order to embed systems better into their environment.

Chapter 3

About Elastic Systems

The new ideas of modeling neocybernetic systems as elastic systems can be read in [15] at length. As that approach starts from determining the goals of evolution there has to be stated out that [1]

> The alternative employed here is rather radical [15].

3.1   Overview

In this chapter a new approach is introduced in order to describe emergent behavior in real-life systems in form of self-organization and self-regulation. Starting from this evolutionary goal the idea of elastic systems is introduced. Elastic systems can be characterized as systems that are connected strongly to their environment: In real-life systems there exist no unidirectional flows: if a system takes energy from its environment the energy consumption changes the environment respectively. This behavior can be described as an implicit feedback structure from the system into its environment. From this point of view new ideas about self-organization and self-regulation in neocybernetic systems can be derived. Starting from the biological domain, it can be shown that the ideas of elastic systems can be extended and an approach for decentralized control purposes can be derived (see Chapter 4).

---

[1]It has to be emphasized that the ideas presented in [15] are still progressing and the derivations made here should give general intuitions about elastic systems.
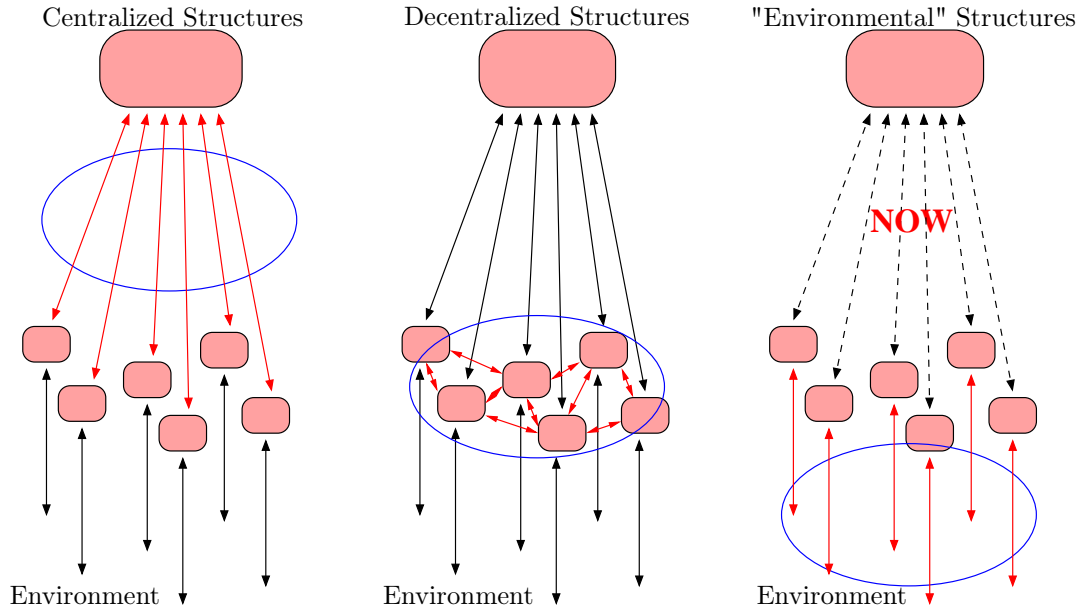
PSfrag replacements

Figure 3.1: New approach of control structures.

Figure 3.1 describes the general structures of control and the role of the new "environmental" structures. In a traditional control approach information of subsystems is collected and evaluated by a central unit in order to coordinate the system in the desired direction. New neocybernetic modeling approaches concentrate on emergent behavior, as a result of communication structures among distributed actors, where finally self-organization and self-regulation emerges. Hebbian/anti-Hebbian learning leads to that behavior, for example, as introduced in Section 2.4. It turns out that new thoughts on elastic systems lead to emergent behavior as well and can be seen – in a wider sense – as an extension of Hebbian/anti-Hebbian learning.

3.2 Theoretical Approach

The following sections form the theoretical framework of elastic systems. After general thoughts of data structuring, the idea of elastic systems is presented and evolutionary fitness is addressed. Afterwards, the emergent behavior of elastic systems in form of self-regulation and self-organization is researched closer.

3.2.1 Static and Dynamic Structuring of Data

Consider a deformed balanced system resulting out of applied external tensions. The static equilibrium state $\bar{x}$ of that linear system can be described as

$$A\bar{x} = Bu \tag{3.1}$$

Vector u has dimension m and describes the environmental conditions, whereas vector $\bar{x}$ of dimension n contains variables, describing the internal system-specific equilibrium states. It has to be mentioned that the internal states are not necessarily observable by an external observer. In order to keep the system well-defined, there exist as many constraints as there are latent variables in the form of $\bar{x}$, with matrix A square ($A \in \mathbf{R}^{n \times n}$). Here, a linear dependency between $\bar{x}$ and u is assumed. If the matrix A is invertible, the steady state $\bar{x}$ of the system can be directly solved from (3.1) as

$$\bar{x} = \underbrace{A^{-1}B}_{\phi^T}u \tag{3.2}$$

and a direct mapping matrix from u to $\bar{x}$ is defined as

$$\phi^T = A^{-1}B \tag{3.3}$$

The main motivation for (3.1) is to extend static modeling towards dynamic structures. If the data structures are selected properly one can define a dynamic model

$$\frac{dx}{\gamma dT} = -Ax + Bu, \qquad \gamma > 0 \tag{3.4}$$

The selection of accurate data structures means that the matrix $-A$ is stable. Factor γ is a scaling factor to adjust the time axis. The steady state for x of (3.4) is $\bar{x} = \lim_{T \to \infty} x$. As the dynamics of the systems are linear, the steady state is unique, independent of the initial state of the system. With the use of that structure, the static pattern has been transformed into a dynamic one, where the emergent static structure reflects the underlying dynamic equilibrium.

Here, the motivation for the dynamical description of a system can be explained intuitively. The static dependencies of (3.1) are basically dynamic equilibria, so that the system will find a new balanced state when external tensions are changing. It is

the interaction among actors in the system which try to drive the system back into a new balanced state. So the mathematical dynamic model describes what a system really does. (3.4) can be interpreted as a negative gradient resulting out of a cost criterion that has to be minimized. So if (3.4) is integrated with respect to variable x the cost criterion can be described as

$$\mathcal{J}(x, u) = \frac{1}{2}x^T A x - x^T B u \tag{3.5}$$

Here it can be emphasized again that neocybernetics concentrates more on the emergent pattern of the system as a result of the actual process than on the process itself that leads to that pattern. The cost criterion itself represents the pattern view and the optimization process represents the process view. This idea was already introduced in Section 2.4.5 for the Hebbian/anti-Hebbian learning. A second criterion from (2.21) that was introduced for the Hebbian learning can also be interpreted within the new structure:

$$\begin{aligned} J(x, u) &= \tfrac{1}{2}(u - \varphi x)^T W (u - \varphi x) \\ &= \tfrac{1}{2}x^T \varphi^T W \varphi x - x^T \varphi^T W u + \tfrac{1}{2}u^T W u \end{aligned} \tag{3.6}$$

so that there holds

$$J(x, u) = \mathcal{J}(x, u) + \frac{1}{2}u^T W u \tag{3.7}$$

These two cost criteria are equivalent with respect to x. The constant factor $\frac{1}{2}u^T W u$ does not change the gradient and therefore neither the dynamics of the system. The two criteria are consistent if one defines the matrices A and B as

$$\begin{cases} A = \varphi^T W \varphi \\ B = \varphi^T W \end{cases} \tag{3.8}$$

These structures are studied closer (see Section 3.2.4.3). If (3.8) holds with W positive definite [15], then the eigenvalues of matrix A are non-negative and a process described with (3.4) always remains stable. Cost criterion (3.6) also gives another view of the gradient based minimization. The steepest descent gradient approach used for minimization can be implemented as a continuous time process of the form

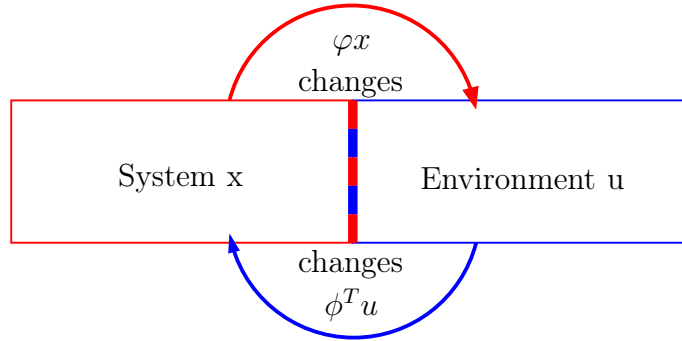$$\frac{dx}{\gamma dT} = \varphi^T W (u - \varphi x) \tag{3.9}$$

Figure 3.2: Interconnection between environment and system.

One can see that this equation equals (3.4), if (3.8) is used. The dynamics of the system can therefore be described as the steepest descent gradient approach of the cost criterion (3.6). Especially the latter part $u - \varphi x$ leads to very sophisticated results, what is discussed later on (see Section 3.2.4.1).

While the matrix ϕ^T implements mapping from the environmental variables into the system variables $\bar{x}$, matrix φ can be interpreted as an inverse mapping from the space of the system variables x into the space of u. It can be seen from (3.3) with (3.8) that there must hold

$$\phi^T = \left(\varphi^T W \varphi\right)^{-1} \varphi^T W \tag{3.10}$$

Later on, more useful results for the interconnection of these mapping matrices can be found.

Now, a new way of thinking has to be emphasized. The changing in any variable of the system causes changes in other variables no matter whether the variable belongs to x or u. The diffusion processes φx and $\phi^T u$ from and into the system might find a balance but it is difficult to find the "original causes" for that. Environmental variables can change the system states, but also a change in the system states can change the environment respectively. A result of these considerations is that a clear separation between the system itself and the environment does not exist any longer. Actually, there exists no intact environment to start with, as a change of state variables leads directly to a change in the environmental variables. The assumption of this two way connection seems to blur the traditional view of distinguishing between a system and its environment. These new ideas of neocybernetic thinking can be
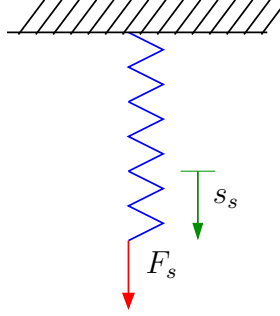
PSfrag replacements

F_s

s_s

Figure 3.3: Prototypical spring being stretched.

visualized in Figure 3.2.

In order to improve the capture of neocybernetic systems, the connection between the mappings φ and ϕ^T has to be researched closer, as they are essential.

3.2.2 Idea of Elastic Systems

The introduced cost criterion (3.5) can also be seen in a wider sense and if the terminology of "elastic systems" is introduced the cost criterion has a very familiar look. For that study first a system of n interconnected springs where m external forces are applied.

A single spring with the spring constant k is stretched by an amount s_s, caused by an external force F_s. Hence, there are internal and external energies stored in the spring (Figure 3.3):

- Due to the potential field: $W_{ext} = \int_0^{s_s} F_s \, ds_s = -F_s s_s$

- Due to the internal tensions: $W_{int} = \int_0^{s_s} k s_s \, ds_s = \frac{1}{2} k s_s^2$

This relationship can be extended to a system with many forces and interconnected springs as mentioned above. Then, for example, the internal tension between two points s_1 and s_2 can be written as follows

$$W_{int}\left(s_{s1}, s_{s2}\right) = \frac{1}{2} k_{12}(s_{s1} - s_{s2})^2 = \frac{1}{2} k_{12} s_{s1}^2 - k_{12} s_{s1} s_{s2} + \frac{1}{2} k_{12} s_{s2}^2 \qquad (3.11)$$

As one defines the vectors $s_s \in \mathcal{R}^n$ and $F_s \in \mathcal{R}^m$ matrix formulation can be used for the above introduced internal and external energies. To achieve that, the interaction

factors are collected in the matrices A and B and it follows

$$W_{int}(s_s) = \frac{1}{2} \begin{pmatrix} s_{s1} \\ \vdots \\ s_{sn} \end{pmatrix}^T A \begin{pmatrix} s_{s1} \\ \vdots \\ s_{sn} \end{pmatrix} \tag{3.12}$$

and

$$W_{ext}(s_s, F_s) = - \begin{pmatrix} s_{s1} \\ \vdots \\ s_{sn} \end{pmatrix}^T B \begin{pmatrix} F_{s1} \\ \vdots \\ F_{sm} \end{pmatrix} \tag{3.13}$$

Now, vector u indicates forces that are acting in a discretized mechanical system and vector x the resulting deformations. Furthermore, matrix A is supposed to be the elasticity matrix and matrix B is a projection matrix, mapping the external forces onto the deformation axes. To gain a stable mechanical system that bears external stresses, matrix A must be symmetric and positive definite. These conditions are fulfilled if (3.8) holds.

With these assumptions, (3.5) becomes the difference between the stored potential energies in the system. Principle of Minimum Potential Energy [27] defines that a pressured structure finds an equilibrium that minimizes this criterion as it tries to exhaust the external forces with minimum change of internal deformations.

Any balanced neocybernetic system follows this criterion. In non-mechanical structures the same intuition can be applied as derived for mechanical systems. The physical interpretation of "force" and "deformation" can be different, but the overall behavior of the system remains intact. If a system is exposed to external tensions deformations take place, but if the external pressure is released the system returns to its original state. In chemistry for example that manner is called Le Chatelier Principle [4] where the dynamic equilibrium of chemical reactions moves in order to counteract external changes.

> Shortly it can be said that every neocybernetic system is identical with elastic systems, characterized by dynamic equilibria.

To quantify the effect of environmental pressures on the system one can once more consider the elastic mechanical system as a prototype. Here, the change in potential

energy results out of the product of force and displacement. That can be extended to any elastic system, regardless of the physical units and variables. The product of the variables $\bar{x}_i u_j$ can therefore be interpreted as energy, being transferred from the environment into the system through the variables u_j and $\bar{x}_i$. Now, this observation is studied in what follows under the following definition:

> **Emergy** (a scalar dimensionless quantity) is the product of the (abstract) force and the corresponding (abstract) deformation.[15]

This definition of "emergy" makes it possible to apply the idea of elastic systems to a wide scope.

As mentioned above, there is not only a transfer of emergy from the environment into the system but also a transfer of emergy from the system itself into the environment. These emergy transfers can be characterized by the matrices ϕ^T and φ. Because of that duality, it is not only $\bar{x}$ that should be seen as a reaction to u but also vice versa.

3.2.3 Evolutionary Fitness

Starting from the discussions above, about the connection between neocybernetic systems with their environment, there should exist a criterion that emphasizes this match with environment. If one includes the ideas about the transfer of emergy from and into systems, a fitness criterion for the system would be

> Maximize the average amount of emergy that is being transferred between the system and the environment

The physical interpretation of the environmental variables is not important, the system will interpret these as resources, trying to exploit them as effectively as possible. Here, it is not predetermined how the available emergy will be used by the system.

As one takes for example a yeast cell, where provided glucose steps can be seen as the external "forces" influencing the system, the energy can be used in different ways. In some cases the mannose-production path outperforms other activities in order to reproduce. The provided energy could also lead to heat production on the other hand.

Altogether one can state out that reproduction and survival are competing goals, but in the long run the yeast cell will exploit the available forces most efficiently [15].

It was already mentioned in Section 3.2.2 that energy transported from the environment to the system can be written as the product of the affected variables. Now, the defined momentary emergy floating from the environmental variable j to the state variable i can be written as $\bar{x}_i u_j$ or if all variables are simultaneously taken into account as $\bar{x}u^T$. In the other direction, the emergy floating from the state variable i into the environmental variable j can be written as $u_j \bar{x}_i$ and for all variables simultaneously as $u\bar{x}^T$. That matrix determines the emergy traverse from the system into the environment. If evolutionary development proceeds with consistency, then the differences between those variable pairs should determine the growth rates of the corresponding links. One can assume for the mapping matrices ϕ^T and φ that a stochastic adaptation process takes place, whereas the observations determine the stochastic gradient direction:

$$
\begin{cases}
\dfrac{d\phi^T}{dt} & \propto \quad \bar{x}(t)u^T(t) \\[2mm]
\dfrac{d\varphi}{dt} & \propto \quad u(t)\bar{x}^T(t)
\end{cases}
\tag{3.14}
$$

The mentioned adaptation process has to be slower than the dynamics of the system itself, described by (3.4). As one has a closer look at (3.3) and (3.14) together, one can realize that these adaptation processes are unstable. High correlations between system and environmental variables could result in still higher correlations between these variables and lead to growth without limit. This adaptation rule is an extension of the Hebbian learning principle introduced in Section 2.4. There, linear feedback for stabilization was proposed from the neocybernetic point of view.

If one assumes for the moment that $\bar{x}$ and u remain bounded for some reason without any explicit additional term, then (3.14) should find a steady state. Moreover, for the solution of this steady state it can be assumed that the matrix elements ϕ_{ij}^T are relative to the correlations between $\bar{x}_i$ and u_j

$$
\phi^T = q \cdot E\{\bar{x}u^T\}
\tag{3.15}
$$

and in the opposite direction there is

$$\varphi = b \cdot E\{u\bar{x}^T\} \tag{3.16}$$

The parameters q and b are constant coupling coefficients and their role will be studied later. Additionally it can be said that the mapping matrices ϕ and φ should be proportional to each other so that there follows

$$\varphi = \frac{b}{q}\phi \tag{3.17}$$

From (3.15) and (3.16) it can be seen that the adaptation in the system is completely local for any element in the matrices ϕ or φ, although the evolutionary goal is presented in collective matrix formulation.

3.2.4 Towards Self-Organization

The two main aspects that are studied closer now are self-regulation and self-organization. The answer for self-regulation is negative feedback. First it has to be researched, how that feedback is implemented, as there are no organized communication structures or signal transfer infrastructures within the considered systems. Section 3.2.4.1 explains the background analysis to understand the question of feedback and after that the aspect of self-organization is researched closer in Section 3.2.4.2. Section 3.2.4.3 gives additional mathematical understanding of the new proposed system adaptation structures and its consequences.

3.2.4.1 Feedback through Environment

In Section 2.4 the problem of stability and therefore self-regulation was solved by adding a linear term for limitation. Here a different approach is considered that was already mentioned in Section 3.1 and can be described as an implicit feedback through the environment.

As mentioned before, there exist no uni-directional flows in real systems: when emergy is consumed by the system, this emergy is taken from the environment. One can speak of an exhaustion of environmental resources in that case. For better understanding

one can take a closer look at the pattern matching process described by (3.9). The first part $\varphi^T W$ matches data against the model and the latter part $u - \varphi x$ can be defined as some virtual environment that is matched. Thereby, the negative feedback structure $-\varphi x$ represents material flow from the system into the environment. So the changed environment $\tilde{u}$ is

$$\tilde{u} = \underbrace{u}_{\text{actual environment}} - \underbrace{\varphi x}_{\text{feedback}} \tag{3.18}$$

That means that the system itself never sees the original environment u, only the distorted $\tilde{u}$, where the momentary material flow φx back in the environment is taken into account. Once more, one is only interested in the final balance of the system after all transients have vanished so that there follows from (3.18)

$$\bar{u} = u - \varphi \bar{x} \tag{3.19}$$

The key issue here is that the negative feedback from the system into the environment keeps the system in balance. From now on it is assumed that the system never sees the original environment u but the "system-influenced" environment $\bar{u}$. With that non-ideality, new functionalities like self-organization are possible and will be shown in the next section. In the beginning no special assumptions are made for the mapping matrix φ. This matrix is any $m \times n$ mapping matrix. Starting from (3.19), multiplying $\bar{x}^T$ from the right, taking expectations and restructuring the resulting equation one has

$$E\{(u - \bar{u})\bar{x}^T\}E\{\bar{x}\bar{x}^T\}^{-1} = \varphi \tag{3.20}$$

If one defines a quantity, describing the difference between the undisturbed open-loop environment u and the disturbed balanced closed-loop environment $\bar{u}$ as

$$\Delta u = u - \bar{u} \tag{3.21}$$

there follows from (3.19)

$$\Delta u = \varphi \bar{x} \tag{3.22}$$

and with (3.20) this leads to

$$\Delta u = E\{\bar{x}\Delta u^T\}^T E\{\bar{x}\bar{x}^T\}^{-1}\bar{x} \tag{3.23}$$

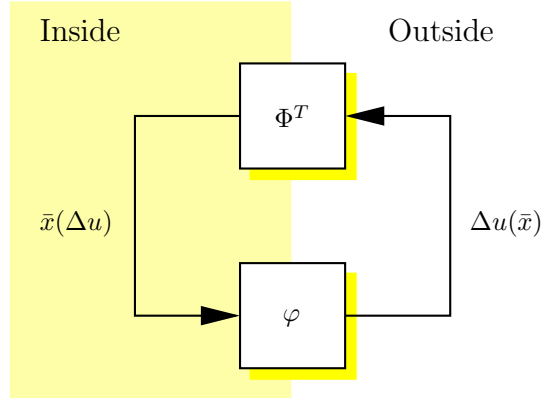PSfrag replacements   Inside   Outside

Figure 3.4: Algebraic loop between system and environment [15].

As Δu is assumedly linearly dependent of the undisturbed environment u this variable can be seen as the actual system input that directs the closed loop behavior and a mapping from Δu to $\bar{x}$ can be defined as

$$\bar{x} = \Phi^T \Delta u \tag{3.24}$$

Hence, here exists an algebraic loop in the system between $\bar{x}$ and Δu as it can be seen in Figure 3.4. The input Δu leads to a change of the system variables $\bar{x}$ what leads instantaneous to a change of Δu respectively. Assuming that the feedback $-\varphi$ implements stabilization, the system in Figure 3.4 will search a balanced state so that there holds from (3.24) with (3.22)

$$\bar{x} = \Phi^T \varphi \bar{x} \tag{3.25}$$

In order that this is not only true for trivial $\bar{x} \equiv 0$ there must hold

$$\Phi^T \varphi = I_n \tag{3.26}$$

meaning that the feedforward and feedback mappings are mutually orthogonal. In order to determine Φ, symmetry to (3.20) is assumed and the matrix Φ^T is evaluated as

$$\Phi^T = E\{\bar{x}\bar{x}^T\}^{-1} E\{\bar{x}\Delta u^T\} \tag{3.27}$$

In the following part this heuristic attempt for Φ^T is elaborated closer and the results of this assumption are presented. It can already be seen that this mapping from Δu

to $\bar{x}$ is similar to the Hebbian/anti-Hebbian mapping rule from (2.16). This time, not the undisturbed environment u is taken into account, but Δu as the difference between the open-loop and closed-loop environment.
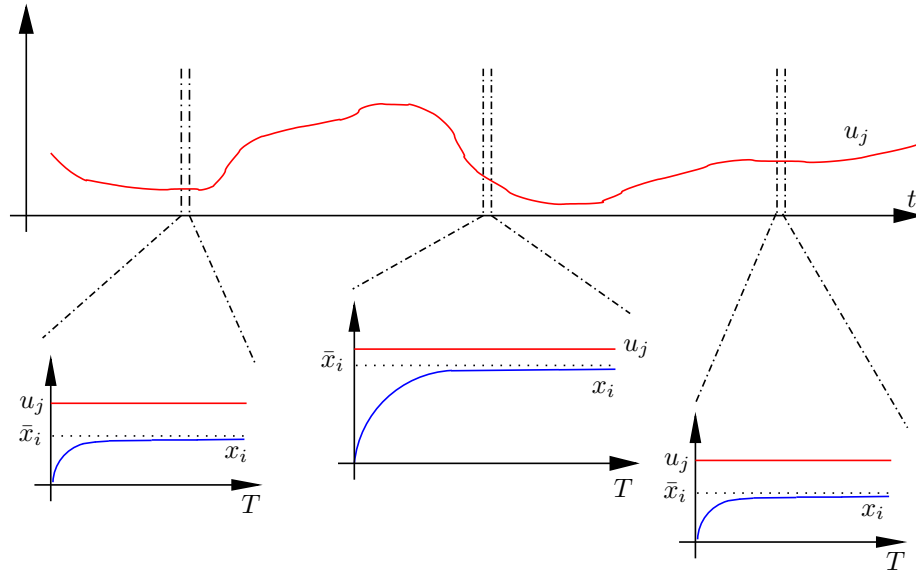
3.2.4.2 Principal Subspace

Figure 3.5: Illustration of two time scales. It is assumed that the dynamics of u (on the t scale) are much slower than that of x (on the T scale) [15]

In the sections above one was interested in the balances $\bar{x}$ of x. For that analysis the environment was assumed fixed. But to reach more interesting results, the neocybernetic principles are applied again. Now it is assumed that there exist various levels of balances within the system and these balances are exploited on each level. As shown in Figure 3.5, the balances of u can be relaxed, as one assumes that the dynamics of the environment have much slower changes than the dynamics of x. If the environment u has fixed statistical properties, one can find a balanced model of balances. A neocybernetic system is a "second-order balance model" over the variations in the system. As one considers these thoughts one reaches stronger views to see systems including self-organization as shown below.

If there holds that the dynamics of u are much slower than the dynamics of x one can study the statistical properties of $\bar{x}$. For that, the covariance matrix $E\{\bar{x}\bar{x}^T\}$ of $\bar{x}$

is constructed. Taking (3.24), multiplying $\bar{x}^T$ from the right and taking expectation values one gains with (3.27)

$$\bar{x}\bar{x}^T = E\{\bar{x}\bar{x}^T\}^{-1}E\{\bar{x}\Delta u^T\}\Delta u\Delta u^T E\{\bar{x}\Delta u^T\}^T E\{\bar{x}\bar{x}^T\}^{-1} \tag{3.28}$$

As covariance matrices are symmetric [26] there holds for $E\{\bar{x}\bar{x}^T\}$:

$$E\{\bar{x}\bar{x}^T\} = E\{\bar{x}\bar{x}^T\}^T \tag{3.29}$$

Applying the expectation operator $E\{\cdot\}$ on both sides, (3.28) leads to

$$E\{\bar{x}\bar{x}^T\} = E\{\bar{x}\bar{x}^T\}^{-1}E\{\bar{x}\Delta u^T\}E\{\Delta u\Delta u^T\}E\{\bar{x}\Delta u^T\}^T E\{\bar{x}\bar{x}^T\}^{-1} \tag{3.30}$$

Multiplied from the left and from the right side with $E\{\bar{x}\bar{x}^T\}$ there follows

$$E\{\bar{x}\bar{x}^T\}^3 = E\{\bar{x}\Delta u^T\}E\{\Delta u\Delta u^T\}E\{\bar{x}\Delta u^T\}^T \tag{3.31}$$

and with the mapping rule $\bar{x} = \Phi^T\Delta u$ there holds

$$\left(\Phi^T(E\{\Delta u\Delta u^T\}\Phi\right)^3 = \Phi^T E\{\Delta u\Delta u^T\}^3\Phi \tag{3.32}$$

In the case of $n = m$ any orthogonal matrix $\Phi^T = \Phi^{-1}$ will do. If the number of system variables n is smaller than the number of environmental variables m, the solution for (3.32) is non-trivial. In [12] it is proved that any subset of input data Δu principal component axes can be selected to constitute Φ, meaning that the columns Φ_i are any set of linear combination of n eigenvectors out of m existing eigenvectors θ_j of the data covariance matrix $E\{\Delta u\Delta u^T\}$, so that there holds $\Phi = \theta D$. D is any orthogonal $n \times n$ matrix so that $D^T = D^{-1}$. In this case, there holds

$$\Phi^T\Phi = I_n \tag{3.33}$$

As one goes back to the assumption from (3.27) one can prove that orthogonality from (3.26) is fulfilled, selecting Φ as introduced above and with φ from (3.20):

$$
\begin{aligned}
\Phi^T\varphi \quad &= \quad E\{\bar{x}\bar{x}^T\}^{-1}E\{\bar{x}\Delta u^T\}E\{\bar{x}\Delta u^T\}^T E\{\bar{x}\bar{x}^T\}^{-1} \\
&\overset{(3.24)}{=} \quad E\{\bar{x}\bar{x}^T\}^{-1}\Phi^T E\{\Delta u\Delta u^T\}E\{\Delta u\Delta u^T\}^T\Phi E\{\bar{x}\bar{x}^T\}^{-1} \\
&= \quad E\{\bar{x}\bar{x}^T\}^{-1}\Phi^T E\{\Delta u\Delta u^T\}^2\Phi E\{\bar{x}\bar{x}^T\}^{-1} \\
&= \quad E\{\bar{x}\bar{x}^T\}^{-1}E\{\bar{x}\bar{x}^T\}^2 E\{\bar{x}\bar{x}^T\}^{-1} \\
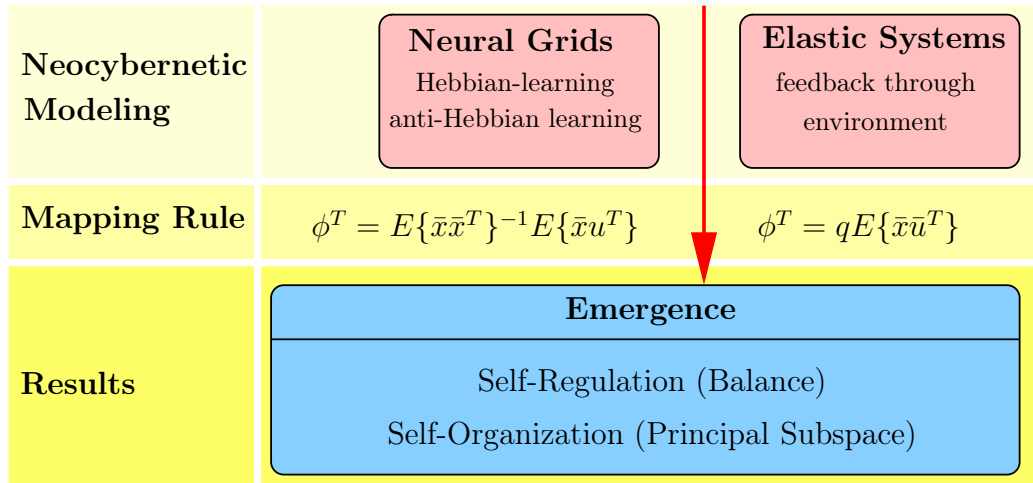&= \quad I_n
\end{aligned}
\tag{3.34}
$$

Neocybernetic Modeling	**Neural Grids** Hebbian-learning anti-Hebbian learning	**Elastic Systems** feedback through environment
Mapping Rule	$\phi^T = E\{\bar{x}\bar{x}^T\}^{-1}E\{\bar{x}u^T\}$	$\phi^T = qE\{\bar{x}\bar{u}^T\}$
Results	**Emergence** Self-Regulation (Balance) Self-Organization (Principal Subspace)	

Figure 3.6: Neural Grids compared to Elastic Systems. The result of these two approaches is self-regulation (stability) and self-organization (principal subspace), but the applied tools are different.

The above made derivations show that for the different combinations of eigenvectors θ_j, the n most significant eigenvectors are selected and therefore principal subspace analysis for the input data Δu is implemented by the system [12]. As the mixing matrix D varies, the result is not unique, but the spanned subspace is the same and captures always the same variation in the input data. Besides that it turns out that any reconstruction $\hat{u}$ of the environment would be equally accurate, no matter if the principal components or the principal subspace are used.

Finally, one can conclude the same observations that where already made in case of Hebbian/anti-Hebbian learning of input data. In the case of Hebbian/anti-Hebbian learning, this learning carried out self-regulation, as the system remained stable and self-organization in terms of principal subspace analysis. The same can now be said for systems, where the feedback is included implicitly through the environment. Here, self-regulation and self-organization emerge in the same way (see Figure 3.6).

3.2.4.3   Closer Look at Cost Criteria

If one compares (3.3) with (3.27) the matrices A and B can be chosen for the mapping between $\bar{x}$ and Δu as [2]

$$
\begin{aligned}
A &= E\{\bar{x}\bar{x}^T\} \\
B &= E\{\bar{x}\Delta u^T\}
\end{aligned}
\tag{3.35}
$$

to connect the data structures appropriately. Compared to (2.16), this mapping matrix ϕ^T was also found when Hebbian learning was applied together with anti-Hebbian structures, realizing active feedback structures (see Section 2.4). The new approach concerning the distorted environment leads to the same results as the Hebbian/anti-Hebbian learning structures but it offers a simpler learning. Anyhow there is a difference observable between the Hebbian/anti-Hebbian structure and the implicit feedback through the environment: while the explicit feedback structure implemented in the Hebbian/anti-Hebbian case analyzes the undisturbed environment, the implicit feedback structure in the case of elastic systems takes the environmental disturbances Δu into account. These disturbances are not known beforehand, as the difference between the open-loop and the closed-loop environment is only measurable after the system is adapted. Then, the analysis made for the original u can be made for the disturbances Δu. In the case of Hebbian/anti-Hebbian learning there is assumed that the environment does not change, meaning that the feedback is implemented without affecting the environment itself.

In order to avoid contradictions there follows for W from (3.8) with (3.22)

$$
W = E\{\Delta u \Delta u^T\}
\tag{3.36}
$$

In the case of explicit feedback the weighting matrix is $E\{uu^T\}$ respectively (see (2.26) in Section 2.4.5). It turns out that the ideas of the previous sections describe an appropriate way of describing behaviors in locally controlled but idealized systems. The observations for emergent behavior in neocybernetic system carry out the modeling process of the environment Δu applying principal subspace analysis (slow adaptation of ϕ) and pattern matching (fast process of determinating $\bar{x}$).

---

[2]With linear dependency between Δu and u the derivations concerning dynamic balances can also be made for Δu

Due to the assumed linearity between Δu and u the cost criterion can also be applied for Δu. If one applies the new knowledge about the matrices A and B in (3.5) under the assumption that (3.27) holds , there follows for the cost criterion of the balanced system, where $\bar{x}$ is already found

$$
\begin{aligned}
\mathcal{J}(u) &= \tfrac{1}{2}\bar{x}^T A\bar{x} - \bar{x}^T B\Delta u \\
&\overset{(3.35)}{=} \tfrac{1}{2}\bar{x}^T E\{\bar{x}\bar{x}^T\}\bar{x} - \bar{x}^T E\{\bar{x}\Delta u^T\}\Delta u \\
&\overset{(3.24)}{=} \tfrac{1}{2}\bar{x}^T E\{\bar{x}\bar{x}^T\}\bar{x} - \Delta u^T E\{\bar{x}\Delta u^T\}^T E\{\bar{x}\bar{x}^T\}^{-1} E\{\bar{x}\Delta u^T\}\Delta u \\
&= \tfrac{1}{2}\bar{x}^T E\{\bar{x}\bar{x}^T\}\bar{x} - \underbrace{\Delta u^T E\{\bar{x}\Delta u^T\}^T E\{\bar{x}\bar{x}^T\}^{-1}}_{\bar{x}^T} E\{\bar{x}\bar{x}^T\} \underbrace{E\{\bar{x}\bar{x}^T\}^{-1} E\{\bar{x}\Delta u^T\}\Delta u}_{\bar{x}} \\
&= -\tfrac{1}{2}\bar{x}^T E\{\bar{x}\bar{x}^T\}\bar{x}
\end{aligned}
$$

The average of that criterion can be written as

$$
\begin{aligned}
E\{\text{trace}\{\mathcal{J}(u)\}\} &= -\tfrac{1}{2}E\left\{\text{trace}\left\{\bar{x}^T E\{\bar{x}\bar{x}^T\}\bar{x}\right\}\right\} \\
&= -\tfrac{1}{2}E\left\{\text{trace}\left\{\bar{x}\bar{x}^T E\{\bar{x}\bar{x}^T\}\right\}\right\} \\
&= -\tfrac{1}{2}\text{trace}\left\{E\left\{\bar{x}\bar{x}^T E\{\bar{x}\bar{x}^T\}\right\}\right\} \qquad (3.37) \\
&= -\tfrac{1}{2}\text{trace}\left\{E\{\bar{x}\bar{x}^T\}^2\right\} \\
&= -\tfrac{1}{2}\sum_{i=1}^n \lambda_i^2
\end{aligned}
$$

The above results can be explained with the properties of the trace of a matrix and the linearity of the operators. The matrix trace is the sum of the diagonal elements and simultaneously it is the sum of the matrix eigenvalues. Additionally, matrices within trace can be rotated if they are appropriately compatible [29]. This result means that the completely adapted system maximizes the sum of the n most significant eigenvalue squares as seen from within the system. If the criterion from (3.5) is used, the optimum reaches

$$
\begin{aligned}
E\{\text{trace}\{J(u)\}\} &\overset{(3.6)}{=} E\left\{\text{trace}\{\mathcal{J}(u) + \tfrac{1}{2}u^T W u\}\right\} \\
&\overset{(3.36),(3.37)}{=} -\tfrac{1}{2}\sum_{i=1}^n \lambda_i^2 + \tfrac{1}{2}\text{trace}\left\{E\{\Delta u^T E\{\Delta u\Delta u^T\}\Delta u\}\right\} \\
&= \sum_{i=1}^n \lambda_i^2 + \tfrac{1}{2}\text{trace}\left\{E\{\Delta u\Delta u^T\}^2\right\} \\
&= \tfrac{1}{2}\sum_{j=n+1}^m \lambda_j^2
\end{aligned}
$$

(3.38)

It has to be mentioned that the eigenvalues λ_i are eigenvalues of the covariance matrix $E\{\Delta u\Delta u^T\}$ and not the open-loop environment $E\{uu^T\}$. Only if intelligent active feedback agents are realizing the feedback the eigenvalues are those of $E\{uu^T\}$.

3.3   Practical Approach

The following part gives intuition how elastic systems can be applied to real-life complex systems and the resulting behavior of these systems is researched. Finally, it is presented, how the free choice of system variables offers the opportunity of the implementation of active control signals in elastic systems.

3.3.1   Applicability of Derivations

The above considerations gave a qualitative understanding about the properties of an existent implicit feedback loop, but there is a lack of applicability, as the system is not able to see the original environment u without disturbing it directly. Δu is only known after the adaptation of the system. From now on, it is assumed that the system only sees the balanced environment $\bar{u}$ disturbed by the feedbacks and according to (3.15) a mapping can be defined as [3]

$$\bar{x} = \phi^T \bar{u} \tag{3.41}$$

with

$$\phi^T = qE\{\bar{x}\bar{u}^T\} \tag{3.42}$$

Now it is only the real measurable environment being involved in the local interactions. In what follows the properties of this definition are researched closer.

By multiplying $\bar{x}^T$ from the right and taking expectation values (3.41) leads with (3.42) to

$$E\{\bar{x}\bar{x}^T\} = qE\{\bar{x}\bar{u}^T\}E\{\bar{x}\bar{u}^T\}^T \tag{3.43}$$

---

[3]To get an intuition how (3.41) and (3.27) are related together and represent the same system there can be derived

$$
\begin{aligned}
\phi^T &\stackrel{(3.42)}{=} qE\{\bar{x}\bar{u}^T\} \\
&\stackrel{(3.17)}{=} qE\{\bar{x}(u - b/q\phi\bar{x})^T\} \\
&= qE\{\bar{x}u^T\} - bE\{\bar{x}\bar{x}^T\}\phi^T
\end{aligned} \tag{3.39}
$$

and solving for ϕ^T and letting b grow one has

$$\phi^T = \left(E\{\bar{x}\bar{x}^T\} + \frac{1}{b}I_n\right)^{-1}\frac{q}{b}E\{\bar{x}\bar{u}^T\} \tag{3.40}$$

If none of the latent variables $\bar{x}_i$ fades away and therefore matrix $E\{\bar{x}\bar{x}^T\}^{-1}$ exists, this holds.

By multiplying $\bar{u}^T$ from the right and taking expectation values (3.41) can be written as

$$E\{\bar{x}\bar{u}^T\} = qE\{\bar{x}\bar{u}^T\}E\{\bar{u}\bar{u}^T\} \tag{3.44}$$

Now, (3.43) becomes with (3.44)

$$E\{\bar{x}\bar{x}^T\} = q^2 E\{\bar{x}\bar{u}^T\}E\{\bar{u}\bar{u}^T\}E\{\bar{x}\bar{u}^T\}^T \tag{3.45}$$

or

$$\begin{aligned}
\tfrac{1}{q}I_n &= \sqrt{q}E\{\bar{x}\bar{x}^T\}^{-1/2}E\{\bar{x}\bar{u}^T\}E\{\bar{u}\bar{u}^T\}E\{\bar{x}\bar{u}^T\}^T E\{\bar{x}\bar{x}^T\}^{-1/2}\sqrt{q} \\
&= \bar{\theta}^T E\{\bar{u}\bar{u}^T\}\bar{\theta}
\end{aligned} \tag{3.46}$$

with

$$\bar{\theta}^T = \sqrt{q}E\{\bar{x}\bar{x}^T\}^{-1/2}E\{\bar{x}\bar{u}^T\} \tag{3.47}$$

For the matrix $\bar{\theta}^T$ there follows from (3.43)

$$\bar{\theta}^T\bar{\theta} = I_n \tag{3.48}$$

With that result there can be said that θ spans the subspace, determined by n of the principal components of the expectation matrix $E\{\bar{u}\bar{u}^T\}$ and it can be proved with simulations that the principal subspace is spanned by the n most significant principal components. Moreover, all eigenvalues $\bar{\lambda}_j$ in the closed loop system are equalized with $1/q$ (see (3.46) and Chapter 6).

Now, the realizable mapping matrix ϕ^T is taken into account and the effects of the virtual mapping on $\bar{x}$ and u are researched closer. Starting from (3.41) with (3.19) and solving for $\bar{x}$ there holds

$$\bar{x} = (I_n + \phi^T\varphi)^{-1}\phi^T u \tag{3.49}$$

and applying the evolutionary learning from (3.15) and (3.16) there is

$$\begin{aligned}
\bar{x} &= (I_n + bqE\{\bar{x}\bar{u}^T\}E\{\bar{x}\bar{u}^T\}^T)^{-1}qE\{\bar{x}\bar{u}^T\}u \\
&\overset{(3.43)}{=} (I_n + bE\{\bar{x}\bar{x}^T\}^T)^{-1}qE\{\bar{x}\bar{u}^T\}u
\end{aligned} \tag{3.50}$$

Taking (3.50) to derive $\bar{x}^T$ and forming the covariance matrix $E\{\bar{x}\bar{x}^T\}$ there holds

$$\left(I_n + bE\{\bar{x}\bar{x}^T\}\right) E\{\bar{x}\bar{x}^T\} \left(I_n + bE\{\bar{x}\bar{x}^T\}\right) = q^2 E\{\bar{x}\bar{u}^T\}E\{uu^T\}E\{\bar{x}\bar{u}^T\}^T \quad (3.51)$$

and rewritten

$$\begin{aligned}
\left(I_n + bE\{\bar{x}\bar{x}^T\}\right)^2 \\
= q\sqrt{q}E\{\bar{x}\bar{x}^T\}^{-1/2}E\{\bar{x}\bar{u}^T\}E\{uu^T\}E\{\bar{x}\bar{u}^T\}^T E\{\bar{x}\bar{x}^T\}^{-1/2}\sqrt{q} \quad (3.52) \\
= q\bar{\theta}^T E\{uu^T\}\bar{\theta}
\end{aligned}$$

This can be done, as there holds for a square matrix M that $M\,f(M) = f(M)\,M$, if a function f is defined in terms of matrix power series. Finally matrix $E\{\bar{x}\bar{x}^T\}$ is

$$E\{\bar{x}\bar{x}^T\} = \frac{\sqrt{q}}{b}\bar{\theta}^T E\{uu^T\}^{1/2}\bar{\theta} - \frac{1}{b}I_n \quad (3.53)$$

The eigenvalues of $E\{\bar{x}\bar{x}^T\}$ can be expressed in terms of the n most significant eigenvalues λ_j of the undisturbed environment $E\{uu^T\}$:

$$eig\{E\{\bar{x}\bar{x}^T\}\}_j = \frac{\sqrt{q\lambda_j} - 1}{b} \quad (3.54)$$

Compared to the discussions in Section 3.2.4.2 one can see now that there is a loss in variation within the system. Whereas in the nominal principal component model the maximum variation is defined as [9]

$$\sum_{i=1}^{n} E\{\bar{x}_{ii}^2\} = \sum_{j=1}^{n} \lambda_j \quad (3.55)$$

the new analysis reveals a reduced variation in the data.

3.3.2   Balance between System and Environment

As (3.46) revealed, an equalization of the environmental variation takes place and a loss of excitation respectively. This equalization is not only a result of the adaptation of the system but also affected by changes in the environment itself. These results of "constant elasticity" are very important in the case of the distributed control of the steel plate, as Chapter 6 reveals. The coupling coefficient q determines directly the shape of the steel plate. In order to gain a neocybernetic system, where all n latent variables are occupied, there must hold

$$q\lambda_n > 1 \quad (3.56)$$

as can be seen from (3.54).

It has to be mentioned that there are also different values for q_i and b_i possible in order to generate individual feedback loops, represented by the different latent variables x_i so that there follows

$$\phi^T = Q \cdot E\{\bar{x}\bar{u}^T\}$$
$$\varphi^T = \tilde{B} \cdot E\{\bar{x}\bar{u}^T\}$$

(3.57)

for the mappings ϕ^T and φ^T. The matrices Q and $\tilde{B}$ are defined as

$$Q = \begin{pmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_n \end{pmatrix}, \qquad \tilde{B} = \begin{pmatrix} b_1 & & 0 \\ & \ddots & \\ 0 & & b_n \end{pmatrix}$$

(3.58)

In this case, the matrix $E\{\bar{x}\bar{x}^T\}$ becomes diagonal as the different eigenvalues become localized. The remaining covariance matrix corresponding to the neocybernetic modes in the observed environment becomes

$$\begin{pmatrix} \frac{1}{q_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{q_n} \end{pmatrix}$$

(3.59)

The reduced covariance matrix of the neocybernetic modes in the system is resprectively

$$\begin{pmatrix} \frac{\sqrt{q_1\lambda_{j(1)}-1}}{b_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\sqrt{q_n\lambda_{j(n)}-1}}{b_n} \end{pmatrix}$$

(3.60)

The notation $j(i)$ means that any permutation of the n most significant eigenvalues of $E\{uu^T\}$ is possible. With that selection of the variables q_i and b_i all cross-correlations of the system variables are eliminated and the covariance matrix $E\{\bar{x}\bar{x}^T\}$ is diagonal. However, the covariance-matrix $E\{\bar{u}\bar{u}^T\}$ is not diagonal.

In Chapter 6 it can be seen that the neocybernetic properties of the system still remain available if the number of latent variables equals the number of system variables $n = m$. In order to have all modes neocybernetic, there must hold
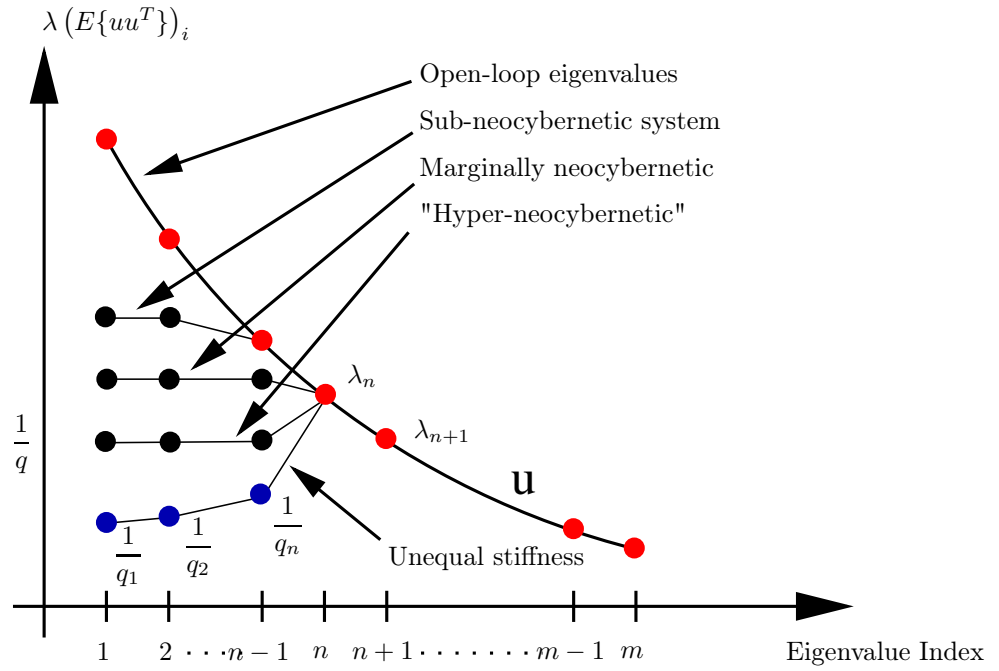
$$q_i\lambda_n > 1$$

(3.61)

Figure 3.7: Schematic illustration of the q factor influence on the resulting system behavior [15].

Figure 3.7 visualizes the different possible equalizations of the eigenvalues depending on the choice of the coupling coefficient q_i. A system, where the constraint (3.61) holds, is called marginally neocybernetic. Systems, where the coupling is too weak is called sub-neocybernetic. The behaviour of sub-neocybernetic systems can be explained in what follows.

As the mapping matrix $\bar{\theta}^T$ demands an invertible matrix $E\{\bar{x}\bar{x}^T\}$, the latent variables x_i do not fade away in fully neocybernetic systems. But there exists a trivial solution, letting the variables fading away

$$\bar{x} \equiv 0 \qquad \text{and} \qquad E\{\bar{x}\bar{u}^T\} \equiv 0 \tag{3.62}$$

Another behavior can be seen in Figure 3.7. Systems can become hyper-neocybernetic. That is the case, if the variation structure is outweighed by less dominant variation directions. As the system still sees the originial variation in u rather than the compensated $\bar{u}$ there are no convergence problems, no matter how high the values of q_i are selected.

The coupling coefficients q_i and b_i remain free design parameters that can have any

values, as long as (3.61) is fulfilled. These coefficients can be explained intuitively:

- **Stiffness ratio** q_i determines how tightly the system is connected to its environment and how strong the system affects its environment.

- **Dissipation rate** b_i determines how efficiently variation on the lower environmental level is transferred onto the higher system level itself.

To assure neocybernetic operation on the system where all variables $\bar{x}_i$ remain occupied there could also be taken a variable q_i factor, making the parameters adaptive

$$q_i = \frac{\nu}{E\{\bar{x}_i^2\}} \tag{3.63}$$

with the parameter $\nu > 1$.

3.3.3 Implementing Control Structures as System Variables

The discussions above were idealized and an evolutionary optimality in the sense of emergy transfer was assumed. However, (3.15) holds generally for a neocybernetic adaptation process, leading to an equalization of the visible environmental signals. On the other hand it can be shown that the whole cybernetic structure still holds if the inverse adaptation from the system to the environment described with (3.16) does not hold and is , for example, fixed for some physical reason with the constant mapping matrix F. If one assumes that the feedback still stabilizes the system from Figure 3.4 the balance will be searched so that there holds

$$\Phi^T F = I_n \tag{3.64}$$

so that the feedforward and feedback mappings are still mutually orthogonal. The feedback structure $\Delta u = F\bar{x}$ can be written as in (3.23) and to make (3.64) hold Φ is given again by (3.27), spanning the principal subspace of $E\{\Delta u \Delta u^T\}$, because (3.34) still holds. In the case of a fixed mapping F the system properties remain essentially the same, as the system optimizes the observed variances Δu in the fixed subspace spanned by F.

It turns out that the latent variables can also be selected freely. For that, it is assumed

that there exists a mapping of the form

$$x' = Dx \tag{3.65}$$

with some invertible matrix D. Then the original formulation of the mapping from the environmental variables $\bar{u}$ onto the system variables $\bar{x}$ from (3.41) can be rewritten by multiplying the matrix D from the left and there follows

$$\bar{x}' = \underbrace{q \cdot E\{\bar{x}'\bar{u}^T\}}_{\phi^T} \bar{u} \tag{3.66}$$

These observations can be applied for analysis of non-ideal real-life systems.

In the case of the research of the deformable mechanical systems as introduced in the following chapters these latent variables x'_i can be selected as actual control signals acting on the system. If one concentrates on the steady-state values of the adaptation process once more, this neocybernetic scheme implements a multivariate control system.

Chapter 4

Control of Elastic Systems

This chapter introduces the application of active control in elastic systems with a distributed set of sensor/actuator couples. In the first section local adaptation is proposed. The adaptation process in a mechanical system is introduced afterwards and the mathematical implementation of these features in a simulation environment is presented.

4.1 Decentralized Control with Local Information

As shown in Section 3.3.3 is it possible to define active control signals $\bar{x}'$ as latent variables in the case of elastic systems. This feature can be extended to set up a distributed control of deformable systems with local sensor/actuator couples. In a distributed network of sensor/actuator couples, the number m of environmental variables u equals the number n of latent control variables x'.

In practical approaches it is usual that not all sensor/actuator information is available for everybody. If it is assumed that the distributed sensor/actuator couples have no active communication structures implemented and adapt only due to its local information, the connection structure (3.66) is diagonal with matrix ϕ^T and respect

to individual control parameters q_i

$$\phi^T = Q \begin{pmatrix} E\{\bar{x}'_1\bar{u}_1\} & & 0 \\ & \ddots & \\ 0 & & E\{\bar{x}'_m\bar{u}_m\} \end{pmatrix} \tag{4.1}$$

Now every sensor/actuator couple is acting completely locally. The sensors measure only local deformations $\bar{u}$ and out of that local counter forces $\bar{x}'$ are implied by the actuators.

However, this very basic consideration of local acting of the participating sensor/actuator couples leads to global emergent behavior in the system. The global goal of self-regulation and self-organization can be seen as the deformation of the structure is getting restructured, according to (3.46). More on the results of the proposed adaptation process can be read in Chapter 6.

In what follows, the realization of the adaptation process is applied to a mechanical deformable system in form of a structural steel plate. The control purpose is performed in a simulation environment by combination of the tools Femlab and Matlab.

4.2   Realization of the Adaptation Process

As one applies external forces on the system surface of a steel plate, a continuous deformation s takes place. These deformations are measured by the applied sensors on the plate and the actuators are evolving forces to counteract to the deformations. With the discretization of the system by using a restricted number m of sensor/actuator couples, not all applied external forces are visible and not all deformations in the system can be compensated, but the "visible" part, projected through m deformation measurements, can be mastered in the neocybernetic framework. For that part, a restructuring of the deformation variations takes place as (3.46) reveals.

The evolutionary control law for the distributed system can directly be taken from (3.66). Here, the environment u is the measured deformation s and the latent control variables x' are the actuator forces F_{act} of the sensor/actuator couples. If a set of external forces is applied, the balanced state for the actuator forces $\bar{F}_{act}$ and the bal-
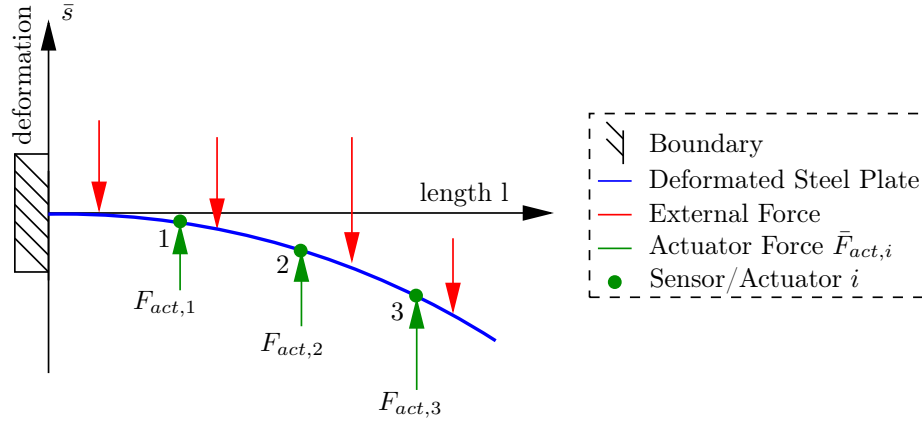
Figure 4.1: Exemplary first order adapted system with distributed sensor/actuator couples. The steel plate is fixed on the left side. The system is in steady-state where the control law follows (4.2). The actuator forces are $\bar{F}_{act,i}$ and the measured deformation is $\bar{s}_i$ for the single actuator/sensor couples i.

anced measured deformations $\bar{s}$ can be described by the control law analog to (3.66) as

$$\bar{F}_{act} = qE\{\bar{F}_{act}\bar{s}^T\}\bar{s}^T \tag{4.2}$$

In this special distributed case, the individual observations and feedbacks are paired in form of sensor/actuator couples. So the general control law (4.2) can be evaluated for every single sensor/actuator pair i, taking only local sensor information into account

$$\bar{F}_{act,i} = q_i E\{\bar{F}_{act,i}\bar{s}_i\}\bar{s}_i \quad i = 1, ..., m \tag{4.3}$$

Generally, there are three possibilities for the control variable q_i analog to the previous chapter

$$q_i = \begin{cases} q = const \\ q_i \qquad \forall i = 1, ..., m \\ \dfrac{\nu}{E\{\bar{F}_{act,i}^2\}} \end{cases} \tag{4.4}$$

The previously introduced feedback through environment can now be explained intuitively as the latent variables are chosen as active control signals: as a set of external forces is applied, the system deforms respectively. This deformation is noticed by the sensors and forces are applied by every single actuator in order to counteract. These actuator forces lead instantly to a change in deformation of the steel plate, so that the counter forces are adapted again until the system finds a balance for the deformations

$\bar{s}$ and the actuator forces $\bar{F}_{act}$ described by (4.3). The resulting balanced system can be described as a "first-order neocybernetic system". Figure 4.1 illustrates this first order balance.

As the applied external forces are not considered to be constant but changing over time, the system tries to find a second order balance through adequate updating of the mapping matrix ϕ^T

$$\phi^T = qE\{\bar{F}_{act}\bar{s}^T\} \tag{4.5}$$

Once again, the locality of the feedback structures can be taken into account, meaning that the mapping matrix is diagonal and every entry ϕ_{ii}^T can be updated locally for every sensor/actuator couple i

$$\phi_{ii}^T = q_i E\{\bar{F}_{act,i}\bar{s}_i\} \quad i = 1, ..., m \tag{4.6}$$

This adaptation of the internal structures tries to match the observed environmental deformations towards maximum experienced stiffness. In Section 3.2.4.2 there was already mentioned that this second adaptation process is considered to be slower than the first order adaptation of the system described by the control law (4.3). The resulting second-order balanced system can be described as a "second-order neocybernetic system".

In the simulation this second order adaptation process is realized by an update of the covariance matrix elements $E\{\bar{F}_{act,i}\bar{s}_i\}$ for every actuator $i = 1, ..., m$ and respectively the update of the elements $E\{\bar{F}_{act,i}^2\}$ if a variable q_i factor is used (see (4.4)). Figure 4.2 visualizes the ongoing adaptation processes exemplary. How the first order adaptation and the update of the expectation values are implemented in the simulation can be read in the following section 4.3.

In summary, there can be said that two adaptation processes are taking place

- **First-order adaptation.** The control law (4.2) finds a balance between the measured steel plate deformation $\bar{s}$ and the actuator force $\bar{F}_{act}$ that compensates deformation, resulting of a set of applied external forces. This first-order adaptation minimizes the actual observed deformation emergy.

- **Second-order adaptation.** The second adaptation process optimizes the mapping matrix ϕ^T, in order to adjust the system to different sets of external forces. The system learns from the different conditions it is exposed to. The second order adaptation minimizes the average observed deformation emergy.
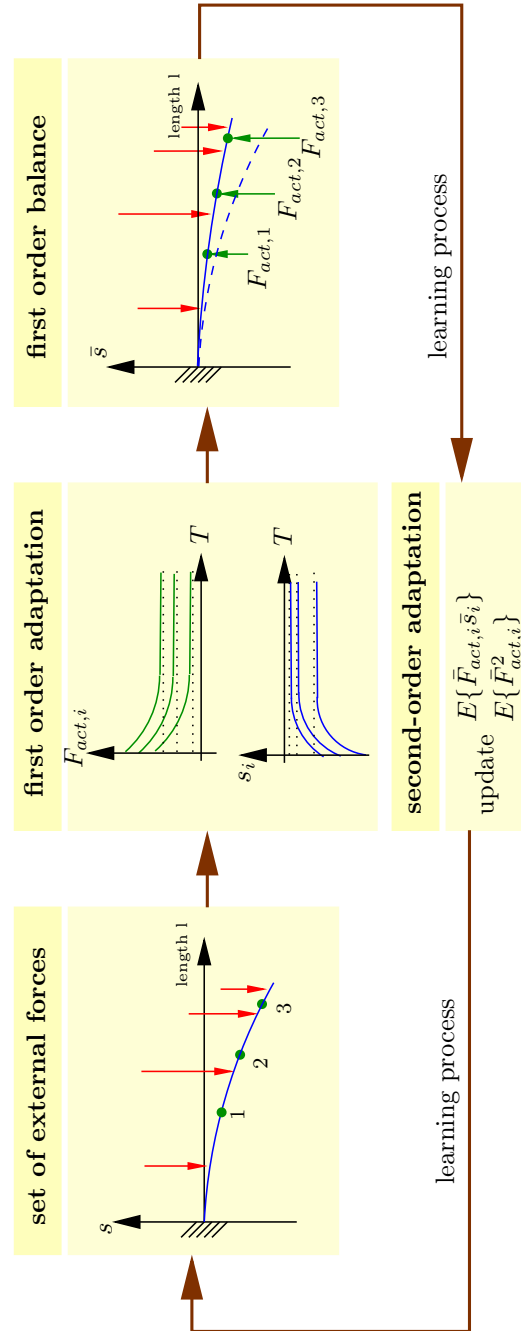
Figure 4.2: Exemplary learning process of the distributed control of a deformable mechanical system.

4.3   Simulative Mathematical Implementation

For the ongoing sections the expectation values $E\{\bar{F}_{act,i}\bar{s}_i\}(l)$ and $E\{\bar{F}_{act,i}^2\}(l)$ are shortly written as

$$
\begin{aligned}
e_i(l) &= E\{\bar{F}_{act,i}\bar{s}_i\}(l) \\
e_{q,i}(l) &= E\{\bar{F}_{act,i}^2\}(l)
\end{aligned}
\qquad i = 1, ..., m \qquad (4.7)
$$

Here, the introduced indicator l describes the use of different sets of external forces that are considered to be constant while the first order balances are calculated. As the previous section revealed, there are two adaptation processes taking place. The first-order adaptation, described by (4.2), expresses the steady state for the actuator forces $\bar{F}_{act}(l)$ and the measured deformation $\bar{s}(l)$ for one particular set l of external forces. But this steady state is not known beforehand and has to be found iteratively. Figure 4.2 shows a "smooth" first order adaptation that could take place in the real system. In order to realize an implementable process the adaptation has to be discretized for simulation purposes and is realized as an iterative process that can be described analog to (4.3) for every single sensor/actuator pair as

$$
\begin{aligned}
F_{act,i}^o(l) &= q_i(l)E\{\bar{F}_{act,i}\bar{s}_i\}(l)s_i^{o-1}(l) \\
&\stackrel{(4.7)}{=} \underbrace{q_i(l)e_i(l)}_{\phi_{ii}^T(l)=\text{const.}} s_i^{o-1}(l) \quad i = 1, ..., m \quad o = 1, 2, ...
\end{aligned}
\qquad (4.8)
$$

$q_i(l)$ can be constant for all actuator forces or a variable factor for every single actuator analog to (4.4)

$$
q_i(l) = \begin{cases} q = const \\ q_i \\ \dfrac{\nu}{E\{\bar{F}_{act,i}^2\}(l)} \stackrel{(4.7)}{=} \dfrac{\nu}{e_{q,i}(l)} \end{cases} \qquad i = 1, ..., m \qquad (4.9)
$$

The factor o in (4.8) denotes the ongoing iteration. The first order adaptation (4.8) for one set l of external forces can be described as follows:

First, the set l of external forces is applied and the steel plate is simulated. The resulting deformations are measured by the sensors and the initial values $s_i^0(l)$ for the algorithm are determined. Out of these deformations, the actuator forces $F_{act,i}^1(l)$ are calculated with (4.8) and applied to the steel plate. Then, the system is simulated again, now with external applied forces and the resulting actuator forces. As a result of these actuator forces, the measured deformations $s_i^1(l)$ will change. The new deformations lead with (4.8) to new actuator forces $F_{act,i}^2(l)$ that are applied again.

The iteration is stopped when the maximum measured deformation change of one sensor i is smaller than a predefined value Δs

$$\max_i \{|s_i^{o_b}(l) - s_i^{o_b-1}(l)|\} < \Delta s \qquad i = 1, ..., m \qquad (4.10)$$

o_b determines the iteration step that leads to a stop of the iteration and (4.10) is fulfilled. If the simulation is stopped the first order steady states for the actuator forces $\bar{F}_{act,i}(l)$ and the deformations $\bar{s}_i(l)$ can be approximated with

$$\begin{aligned} \bar{F}_{act,i}(l) &= F_{act,i}^{o_b}(l) \\ \bar{s}_i(l) &= s_i^{o_b}(l) \qquad i = 1, ..., m \end{aligned} \qquad (4.11)$$

If the first order balance is found, the mapping matrix elements $\phi_{ii}^T(l)$

$$\phi_{ii}^T(l) = q_i(l)e_i(l) \qquad i = 1, ..., m \qquad (4.12)$$

are updated respectively, before the next set $l+1$ of external forces is applied afterwards. This update can be described as the second order adaptation that is taking place in order to optimize the system for different sets of external forces. One can see from (4.12) with (4.9) that one, or in the case of variable q_i factors two different kinds of expectation values ($e_i(l)$ and $e_{q,i}(l)$) have to be considered for the matrix element update. These expectation values depend on the final steady states $\bar{F}_{act,i}(l)$ and $\bar{s}_i(l)$ of the actuator forces and deformations. These values are not known beforehand. That is why the update of the expectation values follows a similar algorithmic implementation as proposed for the Hebbian/anti-Hebbian learning in section 2.4.4 in order to determine the mapping matrix elements $\phi_{ii}^T(l)$.

For that, the expectation values $e_i(1)$ and $e_{q,i}(1)$ for the first set of applied external forces have to be initialized with proper values

$$
\begin{aligned}
e_i(1) &= \epsilon_i < 0 \\
e_{q,i}(1) &= \nu_i > 0
\end{aligned}
\qquad i = 1, ..., m
\tag{4.13}
$$

A proper selection of the initial values is essential for the success of the adaptation process. Therefore, it is necessary that the expectation values $e_i(1)$ are initialized with values smaller than zero and remain so for the whole adaptation process and the expectation values $e_{q,i}(1)$ are larger than zero and remain so. The choice of negative initial values for $e_i(1)$ can be motivated with the causality that measured negative deformations s_i initialize positive actuator forces $F_{act,i}$ to counteract this deformation. Furthermore it has to be emphasized that the values, especially for ϵ_i, should start from some small values, so that the adaptation process can be carried out by the system itself, as dominant expectation values gain weight and less important stay small.

With the definition of initial expectation values, the second order update can be calculated analog to (2.19) as

$$
e_i(l + 1) = \lambda e_i(l) + (1 - \lambda)\bar{F}_{act,i}(l) \cdot \bar{s}_i(l) \qquad i = 1, ..., m
\tag{4.14}
$$

and

$$
e_{q,i}(l + 1) = \lambda_q e_{q,i}(l) + (1 - \lambda_q)\bar{F}_{act,i}^2(l) \qquad i = 1, ..., m
\tag{4.15}
$$

Here, λ and λ_q are forgetting factors and determine, how much of the past information of the system should be taken into account for the learning process. To achieve reasonable results, the values should be higher than

$$
\lambda, \lambda_q > 0.95
\tag{4.16}
$$

The whole adaptation process can be stopped, when the maximum change of the matrix elements $\phi_{ii}^T(l)$ is smaller than a certain value

$$
\max_i\{|\phi_{ii}^T(l_b + 1) - \phi_{ii}^T(l_b)|\} < \Delta\phi \qquad i = 1, ..., m
\tag{4.17}
$$

Analog to the first order breaking indicator o_b, defines l_b the set number of applied external forces that stops the second order adaptation.

This rule can be used if there is no change in the set l of external applied forces. It has to be slightly changed, if varying external forces are taken into account. In that case, mean values of adjacent mapping matrix elements are considered for the break of the second order adaptation (see Section 6.2.3).

Figure 4.3 visualizes the flowchart of the implemented algorithm, used for adaptation of the simulated deformable systems. The colorization indicates already, how the two used environments MatLab and FemLab are involved in the simulation. More on this interconnection can be read in Section 5.1.1.

In what follows, the measured deformation $s_i^o(l)$ of the steel plate sensors is the total displacement of the sensor i for sensor $i = 1, ..., m$. Anyhow, the displacement is not necessarily the evaluated information. In different setups the sensors could also evaluate different quantities, like stresses or strains in order to use these variables for adaptation purposes.
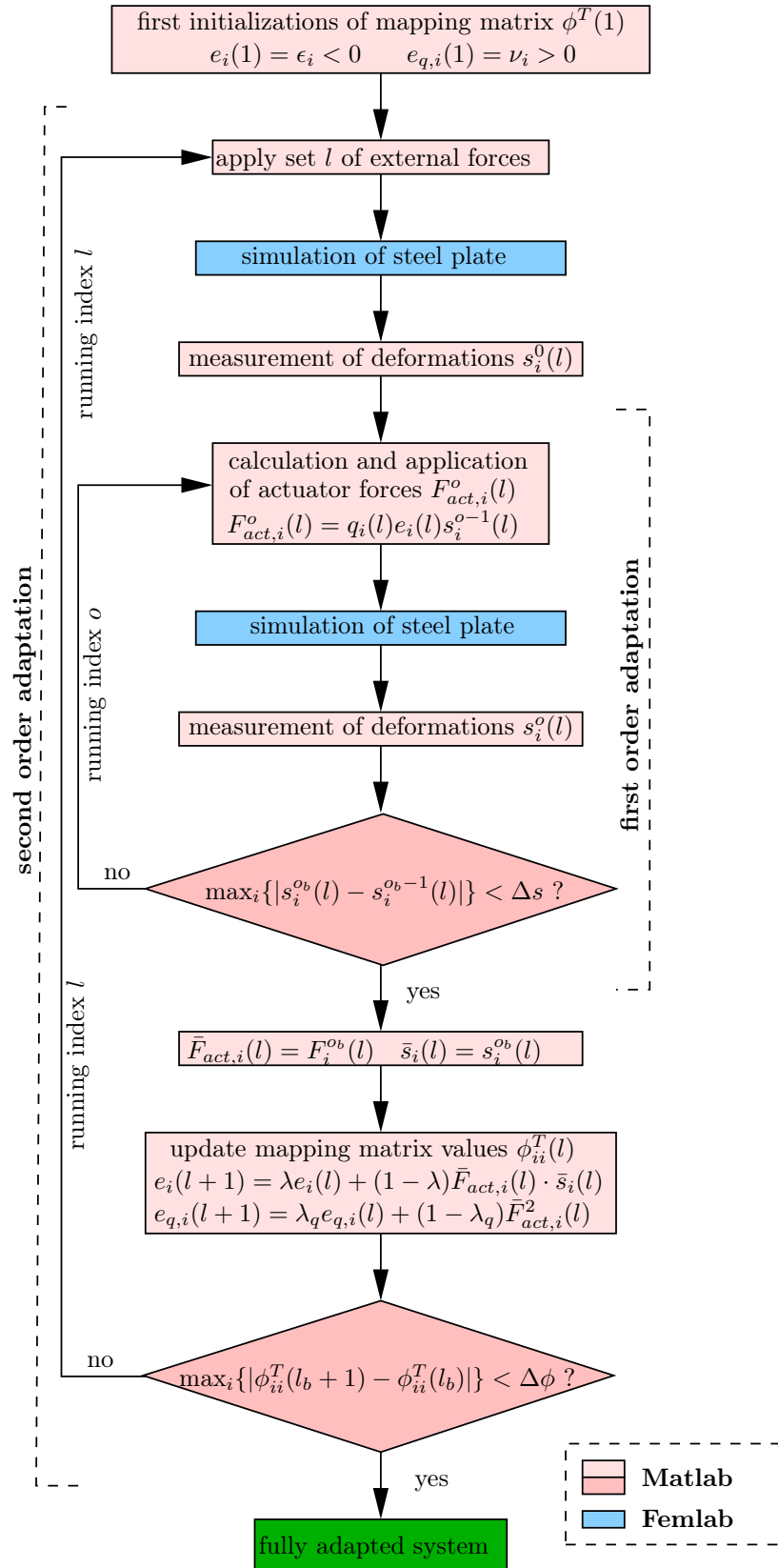
Figure 4.3: Flowchart of the algorithmic implementation of the adaptation of a deformable system.

Chapter 5

Derivation of a New Simulation Algorithm

The previous chapter introduced a completely new framework for the control of elastic systems. This chapter deals with the realization of the simulation of deformable systems and concentrates on the derivation and validation of a faster simulation algorithm.

5.1 Simulation Environment

The following part introduces shortly the used simulation environment consisting of Matlab and Femlab. After that the shortcomings of the conventional simulative control algorithm are addressed.

5.1.1 Interconnection of Matlab and Femlab

In order to simulate mechanical deformable systems, the tool Femlab 3.1 [1], [2] is used for the numerical simulation of deformed systems, where different forces (external and actuator) are applied. Femlab is a multiphysics modeling tool, where physical processes are described with partial differential equations and solved with the finite element method. For the proposed control purposes of deformable mechanical systems the structural mechanics toolbox of Femlab is used. Nowadays, the successor of Femlab is COMSOL Multiphysics$^{\text{TM}}$[5].
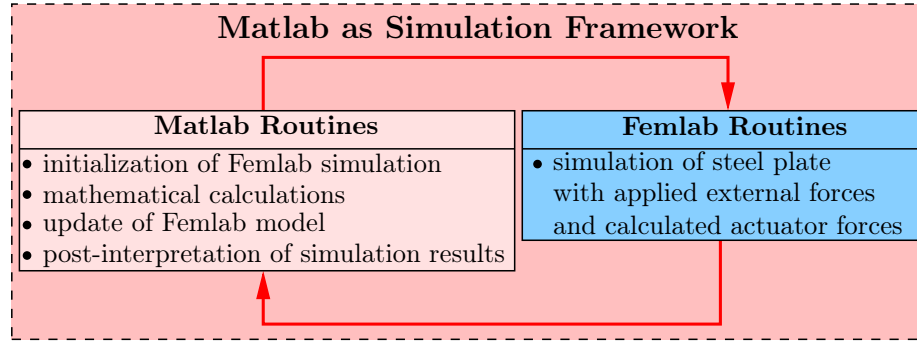
Figure 5.1: Interconnection between Matlab and Femlab for the streamlining of the simulation algorithm.

The process for the adaptation of a simulated deformable system in the neocybernetic framework is a highly iterative process and a lot of simulations have to be taken into account until proper results can be achieved, as Section 4.3 reveals. The first order adaptation is an iterative process that has to be repeated until steady states for the actuator forces $\bar{F}_{act,i}(l)$ and the resulting displacements $\bar{s}_i(l)$ of the sensor/actuator pairs i are found. The iteration routine has to be repeated for different sets of external forces l in order to gain information for the second order balance of the system.

That is why the use of Femlab alone for the simulations carried out, is very clumsy and difficult manageable. For the iterative purposes Femlab does not offer simulation techniques, where different applied forces are updated automatically and the new deformed system is calculated afterwards. For an automation of the whole simulation process Matlab is taken as a second tool and Femlab routines are embedded in a Matlab-automated simulation area. Femlab offers the possibility to save all designed models as Matlab m-files, so it can directly be used for automation. The colored flowchart 4.3 presents in general, how these two tools are interconnected with the introduced simulation algorithm. A more general view of how Matlab and Femlab are linked can be seen in Figure 5.1 [1].

---

[1]It has to be emphasized that the appropriate combination of Matlab and Femlab was implemented by the author of this work and not present yet. However, as the thesis concentrates more on the neocybernetic framework for the control of deformable systems, the realization is presented only briefly.

5.1.2   Shortcomings of Conventional Algorithm

During simulations of the steel plate it turns out that the propagated algorithm from Section 4.3 and its execution shown in the flowchart 4.3 have two major problems that make it difficult to achieve good simulation results in order to analyze the new neocybernetic controlling approach. These two challenges are introduced in the following two sections. First, the simulation times are researched closer and then the possible instability of the first order iteration is presented.

5.1.2.1   Simulation Times

As already mentioned, the algorithm for the second order adaptation of a deformable system is very computing concerning the complexity of the simulation: plenty of simulations of the deformed steel plate have to be calculated with Femlab. For that reason, the total simulation time is immense, as a single Femlab simulation can take approximately $20s$ or even longer. Figure 5.2 illustrates exemplary how long one overall simulation of the steel plate could take until the system is fully adapted. As one tries to research different steel plate setups like different boundary conditions, shapes of the steel plate or layouts of sensor/actuator couples, it is very time intensive to gain interpretable results. Also the closer research of the influence of the control parameters $q_i(l)$ on the final control results is limited. These concerns put the question if some better algorithm can be found in order to shorten the total simulation time for an adaptation process.

5.1.2.2   Instability and Alternation

Despite the long simulation times one even more crucial problem can be observed. Different simulations reveal that too high control parameters $q_i(l)$ for the actuators lead to an instable iteration of the first order adaptation, so that convergence fails. Figure 5.3 shows exemplary the development of the measured sensor displacements $s_i^o(l)$ for an adaptation process, turning instable after some stable second order updates of the mapping matrix elements $\phi_{ii}^T(l)$. It can also be seen in the figure that the first order iterations are oscillating what holds in the simulated cases, where the
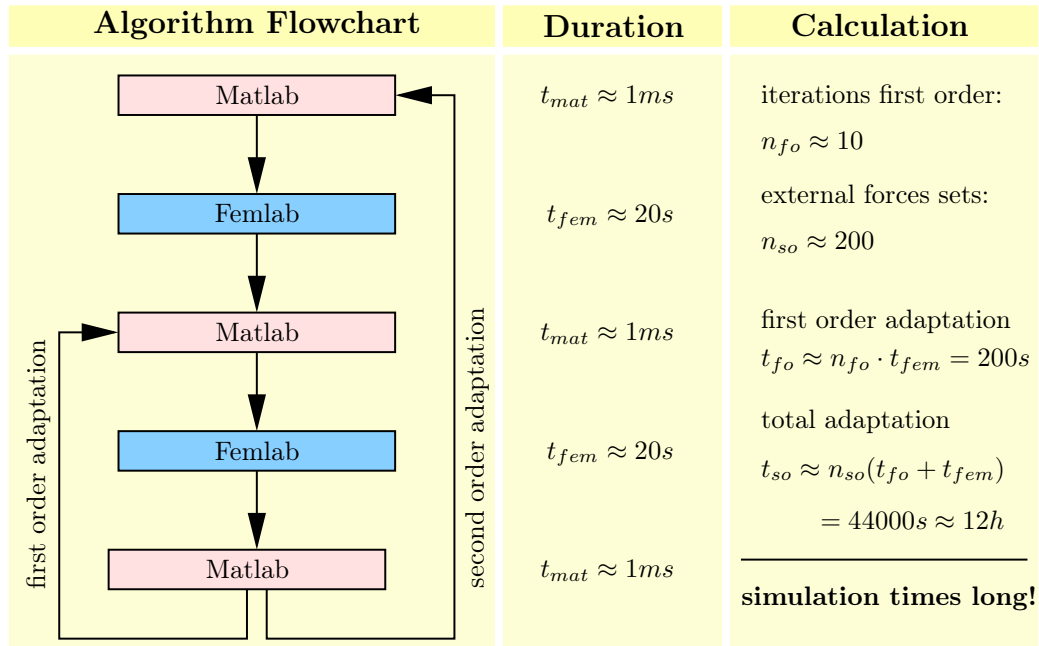
Figure 5.2: Approximate durations of the single simulation steps. The left column shows the flowchart of the simulation algorithm related to Figure 4.3. The middle column considers approximate times for single flowchart steps of the simulation and the right column gives an exemplary simulation time calculation until the steel plate system is fully adapted.

external forces sets l have always same sign. This observation can be used later on, to extend the new applied simulation algorithm to get balanced steel plate systems even if instable first order iterations appear (see Section 5.2.2.3).

The alternation can be explained with the used iteration algorithm (4.8). Big measured displacements lead to big counteracting actuator forces, as control law (4.8) reveals proportionality between the measured displacement and the resulting actuator force. The application of these big actuator forces leads to a reduced deformation of the steel plate in the next iteration step, as most of the external forces are compensated by the applied actuator forces. In the next step, the control law calculates smaller actuator forces, as the resulting steel plate deformations and therefore the sensor displacement measurements are smaller. Now, the application of these smaller actuator forces leads to bigger displacements likewise as the applied external forces are compensated worse than the step before. Table 5.1 shows these causalities once more.
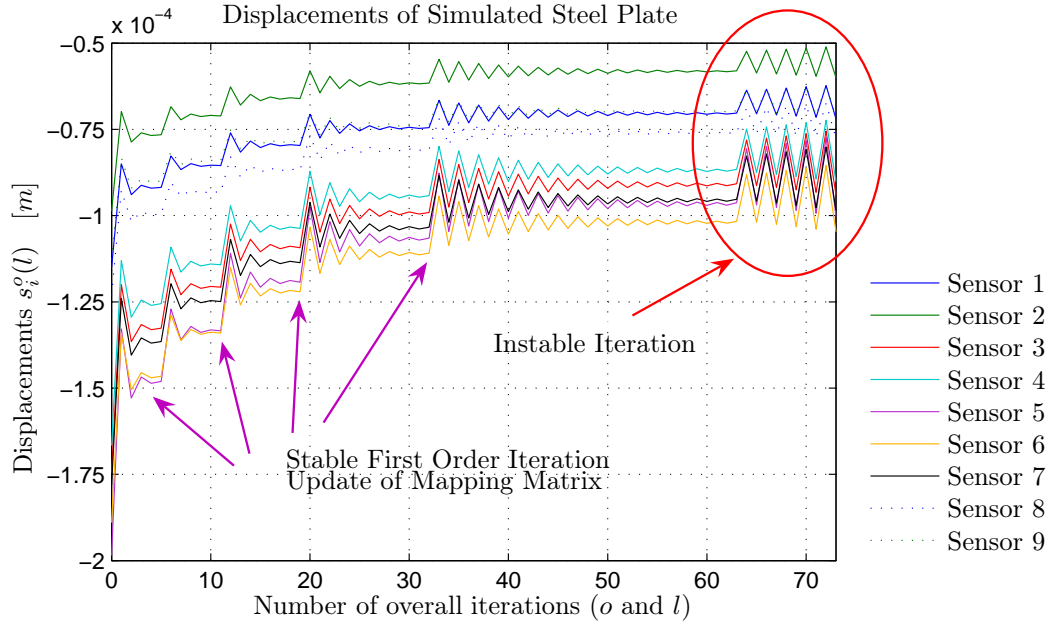
Figure 5.3: Exemplary instable development of the displacements $s_i^o(l)$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.1), control parameter "Set 1" from Table C.3

However, this alternation is not always for every sensor/actuator couple the case, as the single couples can not be considered to be isolated from each other. The measured sensor displacements are a result of all calculated actuator forces, so that there exists the possibility that neighboring pairs work against each other, in the sense of alternation. Is this the case, new control parameters have to be considered.

There has to be emphasized that the occurring instability due to high coupling parameters $q_i(l)$ is a result of the implemented simulation algorithm for the first order adaptation and should therefore not appear in real-life systems. In Section 3.3.2 it was already mentioned that convergence takes place, no matter how high the control parameters $q_i(l)$ are selected. However, this claim can not be researched deeper in simulations.

The following section puts the possible instability into a theoretical framework, so that predictions can be made before the instability actually occurs during the adaptation process. In addition to that, the research on instability leads to a faster algorithm

that can be used for the adaptation process of the steel plate.

leads to		
$s_i^{o-1}(l)$ big	$\overset{(4.8)}{\rightarrow}$	$F_{act,i}^o(l)$ big
$F_{act,i}^o(l)$ big	$\overset{simulation}{\rightarrow}$	$s_i^o(l)$ small
$s_i^o(l)$ small	$\overset{(4.8)}{\rightarrow}$	$F_{act,i}^{o+1}(l)$ small
$F_{act,i}^{o+1}(l)$ small	$\overset{simulation}{\rightarrow}$	$s_i^{o+1}(l)$ big
$s_i^{o+1}(l)$ big	$\overset{(4.8)}{\rightarrow}$	$F_{act,i}^{o+2}(l)$ big

Table 5.1: Quantitative progress of the first order iteration for one sensor/actuator couple i. The successive displacements and actuator forces alternate.

5.2   Introduction of a New Algorithm

The following sections deals with the manageability of instability and proposes a new simulation algorithm to avoid the presented shortcomings of instability and long simulation times. After that, it is presented, how the system parameters for the new algorithm are identified. In order to prove the accuracy of the new algorithm a validation is shown in the end.

5.2.1   Research on Instability of First Order Adaptation

A possible instability of the first order iteration causes the whole adaptation process to fail. This matter of instability makes it crucial to estimate proper starting values for the control variables $q_i(l)$ before the simulation is started. However, one cannot know beforehand if the control parameters were selected correctly or not. That can not be said until the adaptation finds a second order balanced system or fails before that.

The stability problem provokes the question, if the instability can be predicted before it actually occurs. This section introduces a closer mathematical analysis of the first order iteration process from (4.8) and it can be shown that the instability can be put in a mathematical framework that makes prediction possible. For that reason

theoretical thoughts about the deformations of elastic systems are made beforehand and after that the mathematical analysis of the algorithm is derived for prediction purposes.

5.2.1.1 Derivation of a Recursive Sequence for the Sensor Displacements

If one has a closer look at the deformation of an elastic system like the researched steel plate setups, one can assume that a single measured displacement $s_i(l)$ of a sensor i can be described as a sum of displacements caused by all applied external and actuator forces $F_{ext,j}(l)$ and $F_{act,i}(l)$ and the gravity operating on the steel plate. It is assumed that a set l of p external forces is applied and there are m sensor/actuator couples distributed on the steel plate. Now, the measured deformation $s_i(l)$ of a sensor i can be approximated with

$$s_i(l) \cong \underbrace{s_{g,i}}_{\text{gravity}} + \underbrace{\sum_{k=1}^{p} \eta_{ik} \cdot n_k \cdot F_{ext,k}(l)}_{\text{influence external forces}} + \underbrace{\sum_{j=1}^{m} \xi_{ij} \cdot m_j \cdot F_{act,j}(l)}_{\text{influence actuator forces}} \qquad (5.1)$$

The variables are explained as follows:

- $s_{g,i}$: measured displacement of sensor i, caused by the operating gravity, if no forces are applied.

- η_{ik}: influence of external applied force k on the measured displacement of sensor i. η_{ik} can be seen as a constant damping factor, indicating how much of the applied external force k is actually measurable as displacement for sensor i. The factor depends on the boundary conditions of the simulated steel plate and how distant the considered external force and the sensor are.

- n_k: determines the stiffness of the steel plate at the point of the applied external force k. This constant factor can be seen as an "inverse spring constant" indicating how the applied external force is mapped on a displacement at the attacking point.

- ξ_{ij}: influence of actuator force j on the measured displacement of sensor i. Analog to η_{ik}, ξ_{ij} can be described as a constant damping factor, indicating how

much of the applied actuator force j is actually measurable as displacement for sensor i. The factor depends on the boundary conditions of the simulated steel plate and how distant the applied actuator force and the sensor are.

- m_j: determines the stiffness of the steel plate at the point of the applied actuator force j. Analog to n_k, this constant factor can be described as an "inverse spring constant" indicating how the actuator force can be mapped on a displacement at the attacking point.

As the set l of applied external forces is constant during the first order adaptation, (5.1) can be written as

$$s_i(l) = \tilde{s}_i(l) + \sum_{j=1}^{m} \xi_{ij} \cdot m_j \cdot F_{act,j}(l) \tag{5.2}$$

with

$$\tilde{s}_i(l) = s_{g,i} + \sum_{k=1}^{p} \eta_{ik} \cdot n_k \cdot F_{ext,k}(l) \tag{5.3}$$

The term $\tilde{s}_i(l)$ is constant during the calculation of the first order adaptation of the system. (5.2) can be formulated as

$$s_i^o(l) = \tilde{s}_i(l) + \sum_{j=1}^{m} \xi_{ij} \cdot m_j \cdot F_{act,j}^o(l) \tag{5.4}$$

if the first order iteration is taken into account. The iteration step is once again denoted with o. The displacement $s_i^o(l)$ of sensor i depends on the calculated actuator forces $F_{act,j}^o(l)$ in this step of the iteration and with (4.8) there follows

$$s_i^o(l) = \tilde{s}_i(l) + \sum_{j=1}^{m} \xi_{ij} \cdot m_j \cdot \underbrace{q_j(l)e_j(l)}_{\phi_{ii}^T(l)} s_j^{o-1}(l) \tag{5.5}$$

The initial displacements $s_i^0(l)$ for the start of the iteration are the sensor displacements without the application of any actuator forces (see Figure 4.3). (5.3) fulfills this demand so that the initial displacements can be taken as

$$s_i^0(l) = \tilde{s}_i(l) \tag{5.6}$$

If all sensors i are taken into account at once, one can use matrix notation so that there follows analog to (5.5) for the displacement of all sensors

$$s^o(l) = \tilde{s}(l) + \underbrace{\Xi M \phi^T(l)}_{Z(l)} s^{o-1}(l) \tag{5.7}$$

with the sequence matrix $Z(l)$

$$Z(l) = \Xi M \phi^T(l) \tag{5.8}$$

The displacement vectors are

$$s^o(l) = \begin{pmatrix} s_1^o(l) \\ \vdots \\ s_m^o(l) \end{pmatrix}, \quad s^o(l) \in \mathcal{R}^m, \qquad \tilde{s}(l) = \begin{pmatrix} \tilde{s}_1(l) \\ \vdots \\ \tilde{s}_m(l) \end{pmatrix}, \quad \tilde{s}(l) \in \mathcal{R}^m \tag{5.9}$$

and the matrices Ξ, M and $Q(l)$ can be written as

$$
\begin{aligned}
\Xi &= \begin{pmatrix} \xi_{11} & \xi_{12} & \cdots & \xi_{1m} \\ \xi_{21} & \ddots & & \xi_{2m} \\ \vdots & & \ddots & \vdots \\ \xi_{m1} & \xi_{m2} & \cdots & \xi_{mm} \end{pmatrix}, \quad \Xi \in \mathcal{R}^{m \times m} \\
M &= \begin{pmatrix} m_1 & & 0 \\ & \ddots & \\ 0 & & m_m \end{pmatrix}, \quad M \in \mathcal{R}^{m \times m} \\
\phi^T(l) &= \begin{pmatrix} q_1(l)e_1(l) & & 0 \\ & \ddots & \\ 0 & & q_m(l)e_m(l) \end{pmatrix}, \quad \phi^T(l) \in \mathcal{R}^{m \times m}
\end{aligned}
\tag{5.10}
$$

The result of (5.7) is a relationship between consecutive sensor displacements $s^o(l)$ and $s^{o-1}(l)$ of the first order iteration. This recursive sequence can now be analyzed, in order to predict nonexistent convergence.

5.2.1.2 Convergence Research

The researched recursive sequence for the sensor displacements follows from (5.7) with the initial displacements $s^0(l)$ analog to (5.6)

$$
\begin{aligned}
s^o(l) &= \tilde{s}(l) + Z(l)s^{o-1}(l) \\
s^0(l) &= \tilde{s}(l), \quad o = 1, 2, \dots
\end{aligned}
\tag{5.11}
$$

Now, the convergence of this recursive sequence for the displacement vector $s^o(l)$ can be derived easily, if matrix $Z(l)$ is diagonalizable so that there exists a transformation

on principal axes with the transformation

$$s(l) = Ds_*(l) \tag{5.12}$$

Matrix $Z(l)$ is diagonalizable, if all algebraic and geometric multiplicities of the eigen-values of the matrix $Z(l)$ are equal [28]. The transformation matrix D from (5.12) is the matrix of eigenvectors of $Z(l)$ and the diagonalized matrix $\Lambda(l)$ contains the eigenvalues of $Z(l)$ respectively. (5.11) delivers with (5.12) the transformed recursive sequence for the displacements $s_*^o(l)$

$$
\begin{aligned}
Ds_*^o(l) &= D\tilde{s}_*(l) + Z(l)Ds_*^{o-1}(l) \\
\rightarrow \quad s_*^o(l) &= \underbrace{D^{-1}D}_{I_m}\tilde{s}_*(l) + \underbrace{D^{-1}Z(l)D}_{\Lambda(l)}s_*^{o-1}(l) \\
&= \tilde{s}_*(l) + \Lambda(l)s_*^{o-1}(l) \\
s_*^0(l) &= \tilde{s}_*(l) \quad o = 1, 2, ...
\end{aligned}
\tag{5.13}
$$

with

$$
\Lambda(l) = \begin{pmatrix} \lambda_1(l) & & 0 \\ & \ddots & \\ 0 & & \lambda_m(l) \end{pmatrix}, \quad \Lambda(l) \in \mathbf{R^{m \times m}} \tag{5.14}
$$

The essential observation here is that the single transformed sensor displacements $s_{i,*}^o(l)$ are only dependent on their own predecessor as matrix $\Lambda(l)$ is diagonal. The overall process converges now, if all single sequences $s_{i,*}^o(l)$ converge [3]. For a single sequence i it follows from (5.13) with (5.14)

$$s_{i,*}^o(l) = \tilde{s}_{i,*}(l) + \lambda_i(l)s_{i,*}^{o-1}(l) \tag{5.15}$$

and with respect to the initial value, the sequence can be written as a geometric progression

$$s_{i,*}^o(l) = \tilde{s}_{i,*}(l) \sum_{j=0}^{o} \lambda_i(l)^j \tag{5.16}$$

Now, the complete system converges towards some steady state, if every geometric progression converges. In case of the geometric progression (5.16), this holds when

$$|\lambda_i(l)| < 1 \qquad \forall \, i = 1, ..., m \tag{5.17}$$

is fulfilled [17]. In other terms, one can state out that the first order iteration is stable as long, as the magnitudes of the eigenvalues of the matrix $Z(l)$ are smaller than one

$$|eig\{Z(l)\}|_i < 1 \qquad \forall\, i = 1, ..., m \tag{5.18}$$

Hence, instability can be tested before the next first order iteration is started with a new set of external applied forces. For that, matrix $Z(l)$ is updated and the eigenvalues are calculated. As the matrix $Z(l)$ has to be updated after every stable first order iteration, the eigenvalues of $Z(l)$ are researched "online" during the adaptation process and can therefore only predict the following iteration.

5.2.2   Derivation of the Algorithm

This section introduces a faster control algorithm that can be used for the simulation instead of the proposed conventional algorithm from Section 4.3. In addition to that, extensions are made in order to improve the speed of the algorithm (see Section 5.2.2.2) or to extend the application areas (see Section 5.2.2.3). In the end, an overview for the used algorithm for the simulation of the adaptation process is presented. As annotation a smart solution for the balances of the actuator forces $\bar{F}_{act}(l)$ and the displacements $\bar{s}(l)$ is shortly presented. However, it turns out that this approach is restricted due to numerical problems of matrix inversions.

5.2.2.1   New Approach

Equation (5.1) can also be presented in compact matrix notation for all sensor displacements $s(l)$. If the first order iteration o is also taken into account, the matrix notation for the displacements $s^o(l)$ dependent on the actuator forces $F_{act}^o(l)$ can be written as

$$s^o(l) = \underbrace{s_g + H \cdot N \cdot F_{ext}(l)}_{\tilde{s}(l)} + \Xi \cdot M \cdot F_{act}^o(l) \tag{5.19}$$

With p external forces and m actuator forces there follows

$$F_{ext}(l) = \begin{pmatrix} F_{ext,1}(l) \\ \vdots \\ F_{ext,p}(l) \end{pmatrix}, F_{ext}(l) \in \mathcal{R}^p$$

$$F_{act}^o(l) = \begin{pmatrix} F_{act,1}^o(l) \\ \vdots \\ F_{act,m}^o(l) \end{pmatrix}, F_{act}^o(l) \in \mathcal{R}^m$$

(5.20)

Analog to (5.10), H and N are defined as follows

$$H = \begin{pmatrix} \eta_{11} & \cdots & \eta_{1p} \\ \vdots & \ddots & \vdots \\ \eta_{m1} & \cdots & \eta_{mp} \end{pmatrix}, H \in \mathcal{R}^{m \times p}$$

$$N = \begin{pmatrix} n_1 & & 0 \\ & \ddots & \\ 0 & & n_p \end{pmatrix}, N \in \mathcal{R}^{p \times p}$$

(5.21)

Equation (4.8) can be written in matrix notation using the mapping matrix $\phi^T(l)$ as defined in (5.10), so that the calculation of the next actuator forces F_{act}^o with the measured displacements $s^{o-1}(l)$ can be presented with the control law as

$$F_{act}^o(l) = \phi^T(l) s^{o-1}(l) \tag{5.22}$$

The initial displacement $s^0(l)$ for the first order adaptation, after a new set l of external forces is applied can be calculated as the result of the gravity and the applied external forces

$$s^0(l) = \tilde{s}(l) \overset{(5.19)}{=} s_g + H \cdot N \cdot F_{ext}(l) \tag{5.23}$$

Now it turns out that the whole process can be simulated, without the explicit use of Femlab routines during the ongoing adaptation process. (5.19) replaces the Femlab simulation of the steel plate deformation with the applied calculated actuator forces. The resulting displacements $s^o(l)$ can directly be evaluated with (5.19) after the actuator forces $F_{act}^o(l)$ are calculated with the control law (5.22). Figure 5.4 visualizes, how the new algorithm can be used without the explicit simulation of the steel plate deformation with Femlab routines.
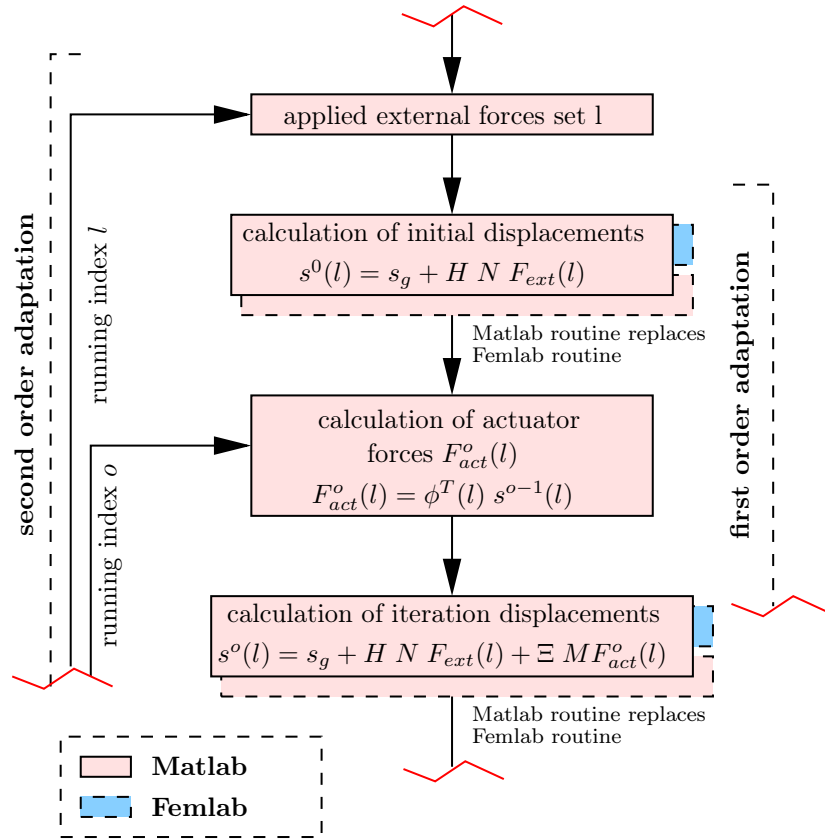
Figure 5.4: New flowchart of the faster algorithmic implementation of the adaptation process. The Femlab simulation of the steel plate deformation is replaced with (5.19).

Before the adaptation process for the steel plate can be started, it is necessary to determine the matrices H, N, Ξ, M and the displacements s_g. These matrices and vectors determine the steel plate deformation behavior. Therefore, it is essential to identify these parameters properly. How these parameters can be found is presented in Section 5.2.3.

5.2.2.2 Initial Values of First Order Iteration

During the simulations of different steel plate setups it turns out that the algorithm can also be advanced, by setting smarter initial values for the first order displacements $s^0(l)$. As some entries of the mapping matrix $\phi^T(l)$ grow with more and more sets l of external forces the calculated actuator forces are growing respectively with (5.22). That leads to bigger changes in the deformation of the steel plate for the ongoing first order iteration, so that first order convergence takes increasing iteration steps

with increasing sets l of external forces.

In order to achieve faster convergence results, the steady state values $\bar{s}(l-1)$ of the previous first order iteration are taken as the initial values $s^0(l)$ for the new applied set l of external forces. So the new law for the initial values $s^0(l)$ is

$$s^0(l) = \begin{cases} s_g + H \cdot N \cdot F_{ext}(l) & l = 1 \\ \bar{s}(l-1) & l = 2, ... \end{cases} \qquad (5.24)$$

Idea of that approach is that the steady states $\bar{s}(l-1)$ of the first order iteration are closer to the final, smaller steel plate deformations after the plate is fully adapted. So the initial displacement $s^0(l)$ is only in case of the first set $l = 1$ of applied forces calculated with (5.23) and in the other cases, the previous final steady states $\bar{s}(l-1)$ are used for the start of the next first order iterations. An exemplary adaptation process of a steel plate and the advantage of the new proposal for the initial displacements is presented in Section 5.3.3.

5.2.2.3   Extension of the Algorithm for Instable First Order Iterations

In Section 5.1.2.2 it was already mentioned that the measured sensor displacements $s_i^o(l)$ and the applied actuator forces $F_{act,j}^o(l)$ for the search of the first order balance are generally oscillating, as it is shown in Figure 5.3. As one assumes that the instability is only a result of the discretization and therefore a numerical problem that does not occur in real systems, one can extend the algorithm in order to gain a second order balanced system, even if the inner adaptation is instable. The extension makes it possible to research higher control factors $q_i(l)$ as the eigenvalue research of the $Z(l)$ matrix would normally allow.

So far the simulation is stopped, if the magnitude of one of the eigenvalues $\lambda_i(l)$ of the sequence matrix $Z(l)$ is bigger than one, and the iteration is turning instable (see Section 5.2.1). If one assumes that the growth of the sensor displacements $s_i^o(l)$ is exponential and alternating for all sensors i (see Figure 5.3), the thoughts from Appendix D can be taken into account and steady states $\bar{s}_i(l)$ of the "inverse exponential growing sequences" can be calculated meaning the value, where the alternating sequence for the displacement is coming from and not where it is going to. This steady
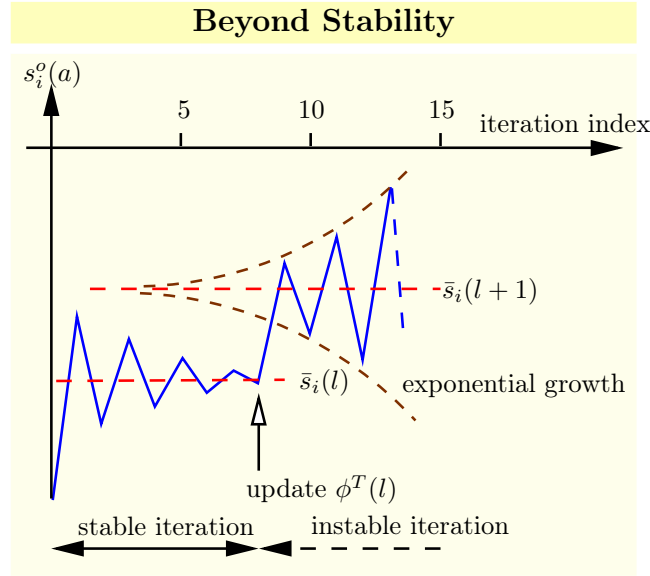
Figure 5.5: Approximation of the instable first order iteration of a single sensor displacement $s_i^o(l)$ with an exponential growing sequence where the steady state can be calculated.

state is taken as the searched steady state $\bar{s}_i(l)$ of the first order adaptation. The necessary balanced actuator force $\bar{F}_{act,i}(l)$ that causes this displacement can be evaluated afterwards with the control law (5.22). The mathematical calculation is presented in what follows. Figure 5.5 visualizes, how the instable iteration is alternating around the searched steady state $\bar{s}_i(l+1)$ for the sensor displacement $s_i^o(l+1)$.

If the check of the eigenvalues of the matrix $Z(l)$ reveals that the magnitude of one eigenvalue is bigger than one and the iteration will turn instable, the initial displacement $s_i^0(l)$ and the two following ones $s_i^1(l)$ and $s_i^2(l)$ are calculated for every sensor i analog to the used algorithm, presented in flowchart 5.4. Finally, the steady states $\bar{s}_i(l)$ for the inverse sequence can be calculated for every sensor with (D.10) and it follows

$$\bar{s}_i(l) = s_i^0(l) + \frac{s_i^1(l) - s_i^0(l)}{\frac{|s_i^2(l)-s_i^1(l)|}{|s_i^1(l)-s_i^0(l)|} + 1} \tag{5.25}$$

With the control law (5.22) the balanced actuator forces can be calculated afterwards

$$\bar{F}_{act}(l) = \phi^T(l)\bar{s}(l) \tag{5.26}$$

As a balanced state is now found for the actuator forces and the measured displacements, the mapping matrix $\phi^T(l)$ from (5.10) can be updated respectively with (4.14)

and (4.15) and a new set of external forces can be applied. As a result, the problem of instability can be relaxed and higher control parameters $q_i(l)$ can be researched.

The results of these calculated second order balanced systems have to be validated with Femlab simulation results as one cannot assure that the proposed "by-pass" of the first order adaptation in case of instability leads to reasonable deformation results that occur when the steady state actuator forces are applied on a steel plate in a Femlab model. Section 5.3.5 shows how a second order balanced system, calculated with the by-pass for the first order adaptation is validated with Femlab simulations of the steel plate deformation.

The adaptation process for the steel plate can not be continued, if the oscillatory alternation is not predictable for every single sensor/actuator pair i, because in this case the steady state displacement $\bar{s}_i(l)$ can not be calculated with (5.25). Then a new set of control parameters has to be taken into account and the whole adaptation process of the steel plate has to be restarted.

Furthermore, the proposed exponential growth makes it appealing to apply that calculation in the other direction as well, so that the first order iteration is always done after three steps. However, the forward implementation of the algorithm is still realized iterative, as the exponential growth is just an approximation of the actual behavior of the sensor displacements $s_i^o(l)$ during the iteration process. Problems can occur if external forces lead to actuator forces that are acting in opposite directions with remarkable different strength. Then, the assumed exponential displacement alternation could be invalid and a forward calculation with assumed exponential growth leads to wrong results. The simulative adaptation of the steel plate fails. Therefore, the use of the conventional forward iteration makes the algorithm more robust.

5.2.2.4   Implementation

As one takes all previous considerations into account, the complete algorithm that is used for the adaptation process of different steel plate setups can be presented in flowchart 5.6. The new algorithm compensates the two introduced shortcomings from
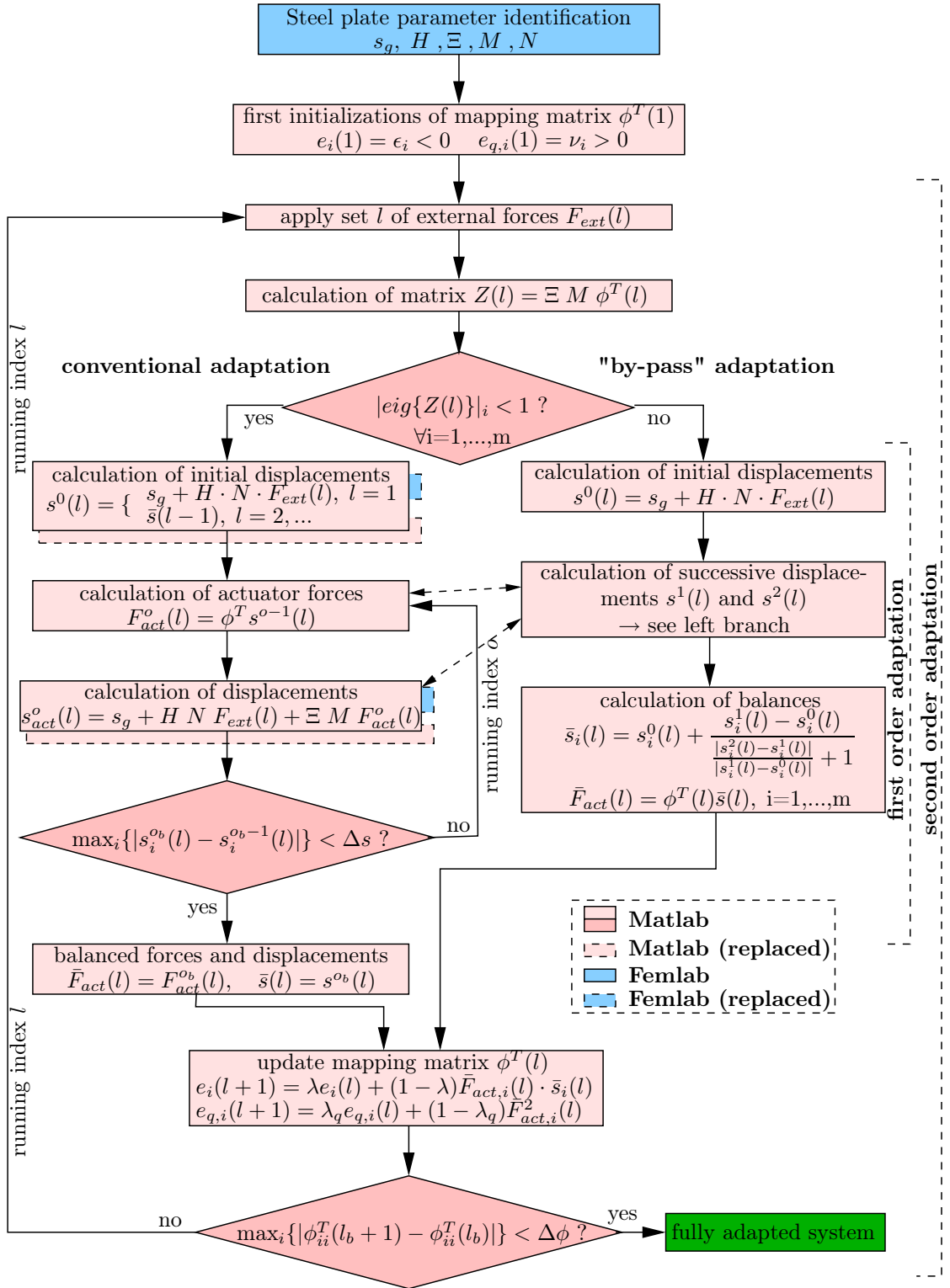
Figure 5.6: Flowchart of the complete used algorithm for the simulation of deformable steel plates.

Section 5.1.2:

- **Simulation Time.** The Femlab based simulation routines that calculate the steel plate deformations are replaced with (5.19) to calculate the measured sensor displacements directly. This replacement fastens the whole algorithm, as the displacements can be calculated with simple matrix and vector operations. The duration of the Femlab based deformation calculations is reduced from a couple of seconds to some milliseconds. As a consequence, the algorithm reduces the total simulation time from hours to a few minutes (improvement: more than 99%!) as all calculations for the system adaptation can be realized with Matlab routines; Femlab is only needed in the beginning of the process. This offers a much better opportunity to research different parameters, like control parameters, boundary conditions and so on. The only new Femlab based part concerns the identification of the needed parameters for (5.19). How the parameter identification is realized, is presented in the following section. This is the only time intensive part. However, it has to be done only once before the adaptation process of the researched steel plate setup is started. As the found parameters are constant for one specific setup, they can easily be saved within Matlab and recalled at any time.

- **Instability.** A deeper analysis of the new equation for the calculation of the sensor displacements makes it possible to predict future instability of the inner iteration algorithm. It turns out that the check for instability can be reduced to an eigenvalue problem of a constructed matrix $Z(l)$ according to (5.8). This matrix is updated and the eigenvalues are checked every time before the inner adaptation process is started. In addition to that, a new approach is introduced that by-passes the instable iteration, so that the second order adaptation process can still proceed, even if the inner adaptation would turn instable. The extension makes it possible to research bigger control parameters, as these parameters are responsible for instability of the inner adaptation.

5.2.2.5 Annotation: Closed Form First Order Iteration

As one goes back to (5.16), presenting a geometric progression for the transformed displacements $s_{i,*}(l)$ one can state out that this progression has the final value

$$\bar{s}_{i,*}(l) = \lim_{o \to \infty} \left(\tilde{s}_{i,*}(l) \cdot \sum_{j=0}^{o} \lambda_i(l)^j \right) = \tilde{s}_{i,*}(l) \cdot \frac{1}{1 - \lambda_i(l)} \qquad |\lambda_i(l) < 1| \qquad (5.27)$$

as long as the magnitude of the eigenvalue $\lambda_i(l)$ is smaller than one [17]. In matrix notation, the steady states for all transformed sensor displacements can be written accordingly

$$\bar{s}_*(l) = \begin{pmatrix} \tilde{s}_{1,*}(l) \cdot \frac{1}{1 - \lambda_1(l)} \\ \vdots \\ \tilde{s}_{m,*}(l) \cdot \frac{1}{1 - \lambda_m(l)} \end{pmatrix}, \quad \bar{s}_*(l) \in \mathcal{R}^m \qquad (5.28)$$

This observation makes the first order iteration after a new set of applied external forces unnecessary. The final sensor displacements $\bar{s}(l)$ can be derived directly. For that, the initial displacements $\tilde{s}(l)$ are calculated with (5.23) for every set l of external applied forces and then transformed in the principal axes coordinate system, where the steady states are given with (5.28) and transformed back into the original system. The transformation of the initial values $\tilde{s}(l)$ in the principal axes system is with (5.12)

$$\tilde{s}_*(l) = D^{-1}\tilde{s}(l) \qquad (5.29)$$

Matrix D is once again the matrix of eigenvectors of the calculated sequence matrix $Z(l)$. The steady states $\bar{s}_*(l)$ can be calculated with (5.28) and afterwards the final steady states of the displacements $\bar{s}(l)$ can be denoted as

$$\bar{s}(l) = D\bar{s}_*(l) \qquad (5.30)$$

For the necessary second order update of the mapping matrix $\phi^T(l)$ the actuator forces $F_{act}(l)$ can now be calculated afterwards with the control law (5.22) so that

$$\bar{F}_{act}(l) = \phi^T(l) \cdot \bar{s}(l) \qquad (5.31)$$

follows. This approach simplifies the control algorithm, as the steady states $\bar{F}_{act}(l)$ and $\bar{s}(l)$ can be calculated explicitly without the need of the first order iteration with the conventional breaking criterion following (4.10).

This theoretic approach turns out to be very sensitive, as the transformation matrix D has to be invertible. The numerical inversion of the transformation matrix D can be problematic, causing errors for the calculated sensor displacements (see Section 5.3.4). That is why the closed form analysis of the steady states is not realized within the actual used simulation algorithm but calculated with the algorithm introduced above.

5.2.3   Identification of System Parameters

It is assumed that m sensor/actuator couples are fixed on the researched steel plate and up to p external forces are applied at predefined positions. Additionally, gravity is acting on the steel plate.

5.2.3.1   Identification of Vector s_g

s_g can be determined if the researched steel plate deformation is simulated with Femlab routines and no forces are applied. The deformation results only of gravity and the sensor displacements can be post-evaluated with Femlab

$$s_g = \begin{pmatrix} s_{g,1} \\ \vdots \\ s_{g,m} \end{pmatrix}, s_g \in \mathcal{R}^m \tag{5.32}$$

The gravitation caused displacements s_g^{ext} of the points where external forces F_{ext} occur are defined as

$$s_g^{ext} = \begin{pmatrix} s_{g,1}^{ext} \\ \vdots \\ s_{g,p}^{ext} \end{pmatrix}, s_g^{ext} \in \mathcal{R}^p \tag{5.33}$$

5.2.3.2   Identification of Matrix M

If linearity between a single sensor displacement s_i and the applied actuator force $F_{act,i}$ at that point is assumed, the relationship for that single sensor/actuator couple i can be approximated with a straight line as

$$s_i = m_i F_{act,i} + s_{g,i}, \quad i = 1, ..., m \tag{5.34}$$

The factor $s_{g,i}$ is already known as the measured displacement caused by gravity (see (5.32)). The slope m_i of this straight line can be determined, if a single test force $F_{act,i} = F_{test}$ is applied on the steel plate.

$$m_i = \frac{s_i - s_{g,i}}{F_{test}}, \quad i = 1, ..., m \tag{5.35}$$

This research can be accomplished for every sensor/actuator couple and the resulting slopes m_i form the matrix entries of matrix M

$$M = \begin{pmatrix} m_1 & & 0 \\ & \ddots & \\ 0 & & m_m \end{pmatrix}, \quad M \in \mathcal{R}^{m \times m} \tag{5.36}$$

5.2.3.3 Identification of Matrix Ξ

The application of one test force F_{test} for a specific actuator force $F_{act,i}$ can also be used to determine the damping factors of the column ξ_i of matrix Ξ. The ith column ξ_i can be defined as follows

$$\xi_i = \begin{pmatrix} \xi_{1i} \\ \vdots \\ \xi_{mi} \end{pmatrix}, \quad i = 1, ..., m \tag{5.37}$$

One matrix column determines, how the single applied actuator force F_{test} at position i affects the displacements of all sensor/actuator couples in relation to the displacement of sensor i where the force is applied. The factors can be found by scaling all measured sensor displacements s with respect to the displacement s_i of the active sensor/actuator couple, where the actuator test force $F_{act,i} = F_{test}$ is applied. Additionally, the influence of gravity has to be considered and the column ξ_i is determined with

$$\xi_i = \frac{1}{s_i - s_{g,i}} \begin{pmatrix} s_1 - s_{g,1} \\ \vdots \\ s_m - s_{g,m} \end{pmatrix} = \frac{1}{s_i - s_{g,i}}(s - s_g), \quad i = 1, ..., m \tag{5.38}$$

Because of this standardization there follows

$$\xi_{ij} = \begin{cases} 1, & i = j \\ \xi, & i \neq j, \end{cases} \quad i, j = 1, ..., m \tag{5.39}$$
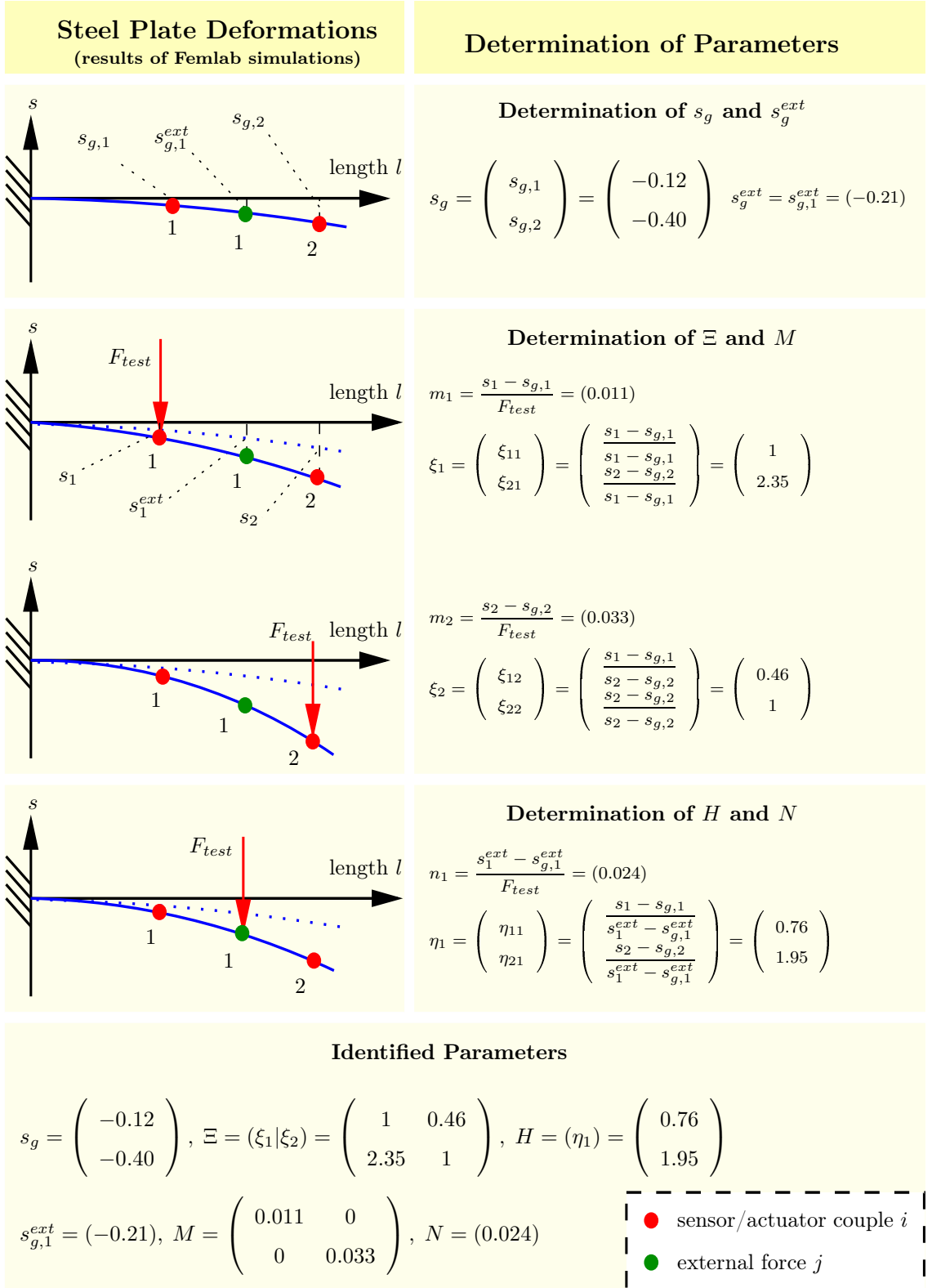
Figure 5.7: Exemplary identification of the steel plate parameters s_g, Ξ, M, H and N.

for the matrix entries ξ_{ij}. Matrix Ξ follows with the determined columns ξ_i as

$$\Xi = (\xi_1 | \cdots | \xi_m), \ \Xi \in \mathcal{R}^{m \times m} \tag{5.40}$$

5.2.3.4 Identification of Matrix N

The assumption of linearity between displacement and applied force leads to an analog calculation of the elements n_j of matrix N. For this, the test force F_{test} is applied at the position of the external force $F_{ext,j}$ and the deformation is simulated with Femlab. Out of the measured displacement s_j^{ext} there follows for the matrix entry n_j

$$n_j = \frac{s_j^{ext} - s_{g,j}^{ext}}{F_{test}}, \quad j = 1, ..., p \tag{5.41}$$

with

$$N = \begin{pmatrix} n_1 & & 0 \\ & \ddots & \\ 0 & & n_p \end{pmatrix}, \ N \in \mathcal{R}^{p \times p} \tag{5.42}$$

5.2.3.5 Identification of Matrix H

Analog to the determination of the columns of matrix Ξ are the columns

$$\eta_j = \begin{pmatrix} \eta_{1j} \\ \vdots \\ \eta_{mj} \end{pmatrix}, \quad j = 1, ..., p \tag{5.43}$$

of matrix N determined by standardization of the measured sensor displacements s of the sensor/actuator couples with respect to the displacement s_j^{ext} at the position of the applied force F_{test}. The column j defines, how the external force j influences the displacements of the sensor/actuator couples i. Additionally, the influence of gravity s_g^{ext} is taken into account and the column η_j is determined as

$$\eta_j = \frac{1}{s_j^{ext} - s_{g,j}^{ext}} \begin{pmatrix} s_1 - s_{g,1} \\ \vdots \\ s_m - s_{g,m} \end{pmatrix} = \frac{1}{s_j^{ext} - s_{g,j}^{ext}} (s - s_g), \quad j = 1, ..., p \tag{5.44}$$

Matrix N follows with the determined columns η_j as

$$H = (\eta_1 | \cdots | \eta_p) \ H \in \mathcal{R}^{m \times p} \tag{5.45}$$

Figure 5.7 illustrates for a simple deformable system, how the parameter identification is working. For that a one dimensional beam was considered with 2 sensor/actuator couples and only 1 possible position for external forces is taken into account.

5.3 Validation of the Algorithm

The key point of the new simulation algorithm is (5.19). This relationship replaces on the one hand the Femlab deformation calculation of the steel. On the other hand, the instability check of the constructed sequence matrix $Z(l)$ is based on that equation. Therefore, it is essential that this approximate displacement calculation of the deformed steel plate holds with the proposed parameter identification.

This section presents the functionality of the introduced algorithm and shows also its shortcomings.

5.3.1 First Order Instability

The predictability of the first order stability based on the magnitude of the eigenvalues of the constructed sequence matrix $Z(l)$ can be shown exemplary for one steel plate adaptation process that is performed with the conventional simulation algorithm according to flowchart 4.3. Figure A.1(a) shows the development of the sensor displacements during the adaptation process and A.1(b) visualizes the corresponding magnitudes of the eigenvalues of the sequence matrix $Z(l)$. It can be seen that first order instability can be predicted as proposed in Section 5.2.1.2 by checking the eigenvalues of the sequence matrix $Z(l)$. The first adaptation steps are stable, until the sixth set of external forces is applied. For this set the eigenvalue check reveals that now an instable iteration process will follow, as the magnitude of one eigenvalue is bigger than one. And indeed, the sensor displacements in Figure A.1(a) show that the iteration turns instable.

5.3.2 Accuracy

In order to achieve reasonable results with the new algorithm (see flowchart 5.4), the development of the sensor displacements should be the same as gained with the conventional algorithm (see flowchart 4.3). To show the accuracy of the formula with the proposed parameter identification from Section 5.2.3, a steel plate is simulated exemplary with both algorithms, the conventional and the newly proposed, and the relative errors for the development of the sensor displacements are compared. In order to neglect the extended approach of by-passing instable first order iterations the simulation takes part for a stable adaptation process, meaning that the magnitudes of the eigenvalues are smaller than one during the whole adaptation process as shown in Figure A.2(c). Figure A.2(b) reveals that the relative errors for the sensor displacements for this simulation never exceed 1%. This good accuracy can be achieved for all researched stable simulations. Therefore, the proposed new algorithm can replace the longlasting conventional simulation algorithm easily, as the results are satisfactory.

5.3.3 Initial Values

Figures A.3(a) and A.3(b) show the developments of the total sensor displacements for two ongoing adaptation processes. In both cases, the same steel plate setup is chosen, but the initial values are differing in these cases.

Figure A.3(a) presents the displacements of the sensor, if the initial values are determined as $s^0(l) = s_g + HNF_{ext}(l)$. The ongoing adaptation process turns out to oscillate very strong and convergence takes very long as can be seen in the total number of iterations, which is more than 3000 to gain a second order balanced system.

Figure A.3(b) shows the sensor displacement development for the new inital value approach according with the initial values determined as $s^0(l) = \bar{s}(l-1)$. The adaptation of the process goes remarkably faster, as the overall iterations for a balanced system take circa 200 iterations and the adaptation is way more smooth compared to the highly oscillatory adaptation above.

Figure A.3(c) compares the convergences of the first order iterations in the two cases of initial values. It can be seen that the conventional initial displacements

$s^0(l) = s_g + HNF_{ext}(l)$ find a balance at circa 50 iterations for the first order balance, as the new approach never exceeds 5 iterations to achieve convergence.

5.3.4 Closed Form First Order Iteration

The problem of the closed form first order iteration according to Section 5.2.2.5 is the invertibility of the transformation matrix D. Figure A.4(a) shows an exemplary steel plate adaptation, where the transformation matrix D gets badly conditioned within the ongoing process. Therefore, the numerical matrix inversion of the transformation matrix D could lead to defective displacement calculations (in the example: sensor two). Figure A.4(b) shows the development of the condition number of the matrix D. It can be seen that the number is growing very strong with values bigger than $1 \cdot 10^4$. A well-conditioned matrix is indicated with a condition number around one [22].

Figure A.4(c) reveals that the adaptation is simulated for a stable process, as the closed form algorithm holds only in cases of stable first order adaptations (see Section 5.2.2.5).

5.3.5 By-Passing for Instable Iterations

Figure A.5(a) shows the development of the sensor displacements of an exemplary fully adapted system that is calculated with the algorithm presented in flowchart 5.6. Figure A.5(b) visualizes the magnitude of the eigenvalues of matrix $Z(l)$. Here, one eigenvalue is beyond the stability border, caused by too big control parameters $q_i(l)$. Hence, this calculated system has to be validated with a Femlab simulation of the steel plate deformation. For that, the calculated actuator forces $\bar{F}_{act}(l_b)$ causing the final displacements $\bar{s}(l_b)$ shown in Figure A.5(a) are applied to the steel plate model in Femlab as a test setup and the deformation of the steel plate is simulated. Afterwards, the calculated deformations $\bar{s}(l_b)$ are compared with the sensor displacements $\bar{s}_{fem}$ evaluated within Femlab. In the case of the simulated steel plate, the final calculated displacements $\bar{s}(l_b)$ and the Femlab evaluated displacements $\bar{s}_{fem}$ have the values shown in Table 5.2.

Sensor	$\bar{s}_i(l_b)$	$\bar{s}_{fem,i}$	$\bar{s}_{rel,i} = \left| \frac{\bar{s}_{fem,i} - \bar{s}_i(l_b)}{\bar{s}_{fem,i}} \right|$
1	$-0.6167 \cdot 10^{-4}$	$-0.6157 \cdot 10^{-4}$	0.0016
2	$-0.5166 \cdot 10^{-4}$	$-0.5220 \cdot 10^{-4}$	0.0104
3	$-0.7069 \cdot 10^{-4}$	$-0.7073 \cdot 10^{-4}$	0.0005
4	$-0.6995 \cdot 10^{-4}$	$-0.7073 \cdot 10^{-4}$	0.0112
5	$-0.7043 \cdot 10^{-4}$	$-0.7052 \cdot 10^{-4}$	0.0013
6	$-0.7136 \cdot 10^{-4}$	$-0.7072 \cdot 10^{-4}$	0.0090
7	$-0.7045 \cdot 10^{-4}$	$-0.7072 \cdot 10^{-4}$	0.0038
8	$-0.6104 \cdot 10^{-4}$	$-0.6068 \cdot 10^{-4}$	0.0109
9	$-0.5875 \cdot 10^{-4}$	$-0.5905 \cdot 10^{-4}$	0.0051

Table 5.2: Proper validation of instability by-pass.

The relative error $\bar{s}_{rel,i}$ for the sensors is in this case around one percent, meaning that the calculated actuator forces and displacements resulting from the used algorithm are appropriate. Anyhow, this validation has to be made always if instable eigenvalues are appearing during the adaptation process. It turns out that eigenvalue magnitudes far beyond one lead to results for actuator forces and steel plate deformations that cannot be proved with Femlab simulations. In these cases the responsible control parameters $q_i(l)$ have to be reduced again, as the calculations for the balanced system cannot be validated with Femlab simulations.

Figures A.6(a) and A.6(b) show the simulation results for the displacements of the sensors and the calculated magnitudes of the eigenvalues of the $Z(l)$ matrix for the same steel plate setup but this time higher control values $q_i(l)$ are chosen. Now there are three calculated eigenvalues bigger than the threshold one. The validation of the achieved second order balance is now executed again and there follow for the different calculated and Femlab post-evaluated displacements of the sensors the values from Table 5.3.

Here the relative errors for the sensors are varying strongly between circa one percent and 22 percent with four sensor displacements above ten percent. In that case, the calculated second order balanced system does not hold with Femlab simulated deformations of the steel plate as the differences are too high for some sensors. A

Sensor	$\bar{s}_i(l_b)$	$\bar{s}_{fem,i}$	$\bar{s}_{rel,i} = \left| \frac{\bar{s}_{fem,i} - \bar{s}_i(l_b)}{\bar{s}_{fem,i}} \right|$
1	$-0.2237 \cdot 10^{-4}$	$-0.1828 \cdot 10^{-4}$	0.2236
2	$-0.2229 \cdot 10^{-4}$	$-0.2103 \cdot 10^{-4}$	0.0597
3	$-0.2233 \cdot 10^{-4}$	$-0.2216 \cdot 10^{-4}$	0.0078
4	$-0.2239 \cdot 10^{-4}$	$-0.2328 \cdot 10^{-4}$	0.0383
5	$-0.1871 \cdot 10^{-4}$	$-0.2163 \cdot 10^{-4}$	0.1348
6	$-0.2237 \cdot 10^{-4}$	$-0.2528 \cdot 10^{-4}$	0.1152
7	$-0.2236 \cdot 10^{-4}$	$-0.2390 \cdot 10^{-4}$	0.0644
8	$-0.2236 \cdot 10^{-4}$	$-0.2160 \cdot 10^{-4}$	0.0352
9	$-0.2236 \cdot 10^{-4}$	$-0.1954 \cdot 10^{-4}$	0.1442

Table 5.3: Inaccurate results for instability by-pass.

possible reason for that is that the linearity assumption of (5.19) for too high control parameters $q_i(l)$ is inaccurate, as the resulting deformations of the steel plate are getting too big and elasticity of the steel plate cannot be assumed any longer. In that case, there is no other possibility then neglecting these simulation results and continue researches for smaller control values $q_i(l)$.

Chapter 6

Simulation: Results and Interpretations

This section presents the simulation results and combines them with the approach about elastic systems, derived in Chapter 3. Generally, there are three parts following.

The first section introduces shortly how the adapted system can be described mathematically, according to the derivations made earlier.

Section 6.2.2 shows the results achieved with local learning information. This approach is interesting for control purposes of totally decentralized sensor/actuator networks, where every sensor/actuator couple is just "aware of itself".

The next part is more related to the theoretical approaches of elastic systems and deals therefore with distributed sensor/actuator networks, where the single sensor/actuator couples have not only their information but also sensor information from all other sensor/actuator couples. However, the practical relevancy of these control setups with a full mapping matrix is limited. The section should give an outlook of the simulative validation of the theoretical approaches from Chapter 3 in a more complex simulation framework and also emphasize the evolutionary fitness of the outcomes.

The last section interprets the simulation results in different contexts like the system theoretic point of view, the control aspect and in a wider sense the evolutionary interpretation of steel plate adaptations.

The theoretical considerations about the used simulation algorithm in chapter 5 hold in the case of a full mapping matrix ϕ^T and can be applied for these simulation purposes.

6.1 Criteria for Validation of the Theoretical Approaches and Emergent Behavior

With the assumed mapping from (3.65) all derived equations for latent variables $\bar{x}$ are also valid for control variables $\bar{x}' = \bar{F}_{act}$, so that these variables can be replaced with $\bar{F}_{act}$ in the derived formulas of Chapter 3. The undisturbed environment u is constituted by the initial displacements $\tilde{s}$ of the sensors, when no adaptation is taking place. In what follows the replacements are defined as

$$\begin{cases} \bar{x} = \bar{F}_{act} \\ \bar{u} = \bar{s} \\ u = \tilde{s} \end{cases} \tag{6.1}$$

Furthermore, the variability of the control parameter $q_i(l)$ is taken into account and is therefore written analog to (3.58) as matrix Q. The possible values of the coupling parameters q_i are defined in (4.4).

(3.46) can now be written as

$$Q^{-1} = \bar{\theta}^T E\{\bar{s}\bar{s}^T\}\bar{\theta} \tag{6.2}$$

This equation reveals that the eigenvalues of the covariance matrix of the balanced environment are equalized and determined by the matrix Q. For control purposes in the case of local sensor/actuator information the coupling matrix Q can be taken directly in order to predict the local variances of the sensor displacements (see Section 6.2.1). Therefore the emergent shape of the steel plate can be predicted with the control parameters q_i of Q before the adaptation process is started.

In the case of decentralized learning for the sensor/actuator couples, (6.2) can be taken in order to validate the theoretical approaches. In the balanced system, the eigenvalues of the balanced environmental covariance matrix $E\{\bar{s}\bar{s}^T\}$ are determined

directly by the coupling parameters $q_i(l)$ of matrix Q.

As the number of control signals n equals the number of measured sensor displacements m, the claim for complete neocybernetic modes is determined by the smallest eigenvalue λ_m of the undisturbed environmental covariance matrix $E\{\tilde{s}\tilde{s}^T\}$. To make an equalization of all eigenvalues possible there must hold

$$q_i \lambda_m > 1 \tag{6.3}$$

analog to (3.61). This statement can be taken as a second criteria to compare the theoretical results with simulations.[1]

6.2 Localized Learning

In what follows, the results for local learning of the sensor/actuator couples are shown for different coupling parameters $q_i(l)$. Before that, mathematical considerations are made in order to interpret the results for the steel plate adaptations in the theoretical framework.

6.2.1 Mathematical Considerations

As one takes into account that the covariance matrix $E\{\bar{F}_{act}\bar{s}^T\}$ is diagonal in the researched case, this diagonalization leads to some more conclusions. The covariance matrix $E\{\bar{F}_{act}\bar{F}_{act}^T\}$ from (3.43) is now

$$E\{\bar{F}_{act}\bar{F}_{act}^T\} = QE\{\bar{F}_{act}\bar{s}^T\}E\{\bar{F}_{act}\bar{s}^T\}^T \tag{6.4}$$

This equation reveals that the covariance matrix $E\{\bar{F}_{act}\bar{F}_{act}^T\}$ for the actuator forces becomes diagonalized as well. So (6.4) can be written as

$$E\{\bar{F}_{act}\bar{F}_{act}^T\} = QE\{\bar{F}_{act}\bar{s}^T\}^2 \tag{6.5}$$

---

[1]For the ongoing considerations of local information adaptation, all considered signals $\bar{s}$, $\tilde{s}$ and $\bar{F}_{act}$ have zero mean.

This equation can be used in (3.47) to calculate the mapping matrix $\bar{\theta}^T$ and there follows for the mapping

$$
\begin{aligned}
\bar{\theta}^T &= Q^{1/2} E\{\bar{F}_{act}\bar{F}_{act}\}^{-1/2} E\{\bar{F}_{act}\bar{s}^T\} \\
&= Q^{1/2} \left(Q \underbrace{E\{\bar{F}_{act}\bar{s}^T\}^2}_{>0} \right)^{-1/2} \underbrace{E\{\bar{F}_{act}\bar{s}^T\}}_{<0} = -I_n
\end{aligned} \tag{6.6}
$$

The positive sign of the first covariance term is a result of the quadrature of the mapping covariance matrix $E\{\bar{F}_{act}\bar{s}^T\}$ in (6.5) and the negative sign in the second term is a matter of causality between measured displacement and applied actuator force (see Section 4.3).

Finally it can be said that the covariance matrix of the observed environment $E\{\bar{s}\bar{s}^T\}$ becomes a variance matrix, where the values are determined by the coupling matrix Q. This can be seen from (6.2) with (6.6) and the result is

$$
Q^{-1} = E\{\bar{s}\bar{s}^T\} = E\{\bar{u}\bar{u}^T\} = \begin{pmatrix} E\{\bar{s}_1^2\} & & 0 \\ & \ddots & \\ 0 & & E\{\bar{s}_m^2\} \end{pmatrix} \tag{6.7}
$$

The undisturbed environment matrix $E\{\tilde{s}\tilde{s}^T\}$ becomes a variance matrix as well. This localization in (6.7) makes it very easy to predict the behavior of the steel plate during the adaptation process, as the coupling parameters $q_i(l)$ of the matrix Q determine directly the variances $E\{\bar{s}_i^2\}$ of the sensors after the adaptation of the system. These observations are shown in the following sections exemplary for different setups of the steel plate.

It has to be stated out that the neocybernetic scaling demands an invertible scaling matrix $E\{\bar{F}_{act}\bar{F}_{act}^T\}$ (see (6.6)). Simulations show that the trivial solution of (3.62) can occur in the balanced system, meaning that expectation values $E\{\bar{F}_{act,i}\bar{s}_i\}$ are getting zero, zeroing the induced actuator forces $\bar{F}_{act,i}$ likewise. In that case only the entries corresponding to the non-zero covariance matrix entries $E\{\bar{F}_{act,i}\bar{s}_i\}$ are following (6.7). It turns out that the sensor variances, belonging to the zeroed actuator forces, are smaller in the case of inactive actuators, meaning that (6.7) indicates in these cases the maximum expected variance $\max E\{\bar{s}_i^2\} = 1/q_i$ of the corresponding sensor displacement.

6.2.2   Constant External Forces

In order to get a better understanding about the ongoing steel plate adaptation and the interpretations of the fully adapted system, constant forces can be used for the external forces sets l so that there holds

$$F_{ext}(l) = F_{ext} = const. \qquad \forall l = 1, ..., l^b \tag{6.8}$$

From this starting point rather intuitive observations are gained from the final balanced systems and the results can be adapted easily to the more practical case of varying external forces (see Section 6.2.3). For this approach different coupling coefficients $q_i(l)$ are taken into account and the adapted steel plate is researched closer. Simulations show that in the case of the steel plate adaptation three general observations can be made and categorized from the neocybernetic point of view. These different categories can be determined by the strength of the coupling parameters $q_i(l)$:

- **No emergence in the system.** If the coupling parameter $q_i(l)$ is to small, no global emergence is instantiated and the system falls back to its original deformation. Every sensor/actuator couple turns inactive again, after the adaptation process is started. This can be explained with the fact that the q factor is determining the maximum variance of the balanced sensor displacements and too low coupling coefficients q allow variances that are higher than the original sensor displacement variances. This is the case, if

$$\frac{1}{q_i(l)} > \lambda_1 \tag{6.9}$$

  Here, λ_1 indicates the biggest eigenvalue of the variance matrix $E\{\tilde{s}\tilde{s}^T\}$ of the undisturbed environment.

- **Sub-neocybernetic system.** The coupling parameter $q_i(l)$ is high enough to lead to an adaptation of the system, but not all actuators are staying active in the adapted system. Some actuator forces are zeroed during the adaptation process, as the respective mapping matrix elements ϕ_{ii}^T are zeroed during the iteration process.

- **Neocybernetic system.** All actuator forces are finally on a non zero level, meaning that all actuators are staying active.

6.2.2.1 Constant Coupling Parameters

In this section constant coupling parameters are researched, meaning that $q_i(l)$ is according to (4.4)

$$q_i(l) = q = const \qquad \forall i = 1, ..., m \tag{6.10}$$

Analog to Figure 3.7, Figure 6.1 shows the results of the equalization process for the environmental variables $\bar{s}_i$ of a fully adapted steel plate. The balanced environmental variances $E\{\bar{s}_i^2\}$ are compared to the open loop variances $E\{\tilde{s}_i^2\}$ (blue markers) depending on the choice of the coupling parameter q of the mapping matrix ϕ^T. The variances are sorted in descending order.

It can directly be seen from the picture that the coupling parameter q determines the maximum variation in the data. The higher the value of the coupling coefficient, the bigger is the loss of variation in the data as it is described in Section 3.3.2. It can also be seen that with higher q factors, more and more balanced environmental signals $\bar{s}_i$ are equalized:

For q_1, only one sensor signal is following equalization (6.7). With higher q factors there are two (q_2, q_3), three (q_4) and finally all sensor signals equalized (q_5), as it is predicted with formula (6.7). Only for $q = q_5$ the system behaves neocybernetic. In the other cases the system is sub-neocybernetic and the equalization is not a result for all measured sensor signals. The respective actuator forces are fading away and not active part of the control structure after adaptation.

The zeroing of different modes $\bar{F}_{act,i}(l)$ for low q factors can be explained with the fact that in this case only the actuator/sensor couples with high displacements are gaining weight. The sensors measuring lower displacements (for example the couples closer to the boundary) are turning out to get inactive. The active couples are enough to reach the goal of maximum variation given by the coupling factor q and the lower variances of the inactive sensor/actuator couples are a side effect of the active couples. For an equalization the q factor has to be high enough to emphasize the less dominant sensor/actuator couples as well.
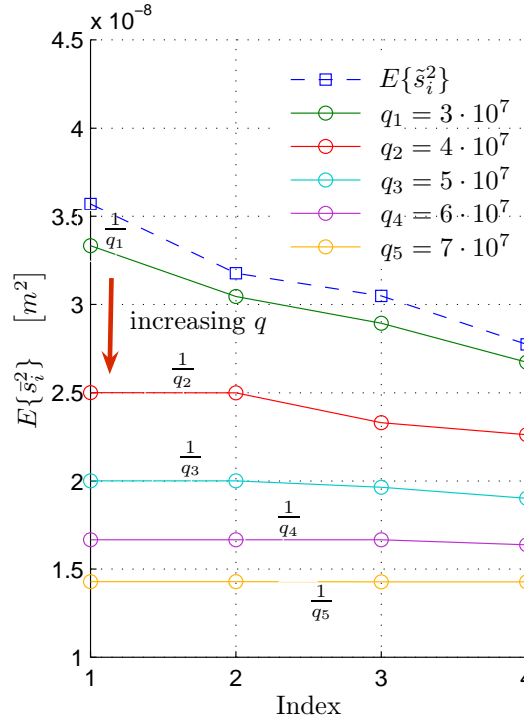
Figure 6.1: Influence of constant q factors on the equalization of the displacement variances of $E\{\bar{s}_i^2\}$.

As the use of a diagonal mapping matrix ϕ^T localizes the eigenvalues of the balanced environmental signals and constant forces are applied this leads to a smart estimation of the maximum sensor displacement, connected closely to (6.7). In this case the maximum measured balanced sensor displacement $\bar{s}_i$ can be determined by the coupling factor q directly and there follows

$$\max|\bar{s}_i| \leq \sqrt{\frac{1}{q}} \qquad \forall i = 1, ..., m \qquad (6.11)$$

These observations can be validated in simulations of the steel plate adaptation process and are shown in appendix B. In the case of $q = q_5$ this means that the steel plate adaptation leads to a state, where all measured sensor displacements $\bar{s}_i$ are equal. In the other cases, some balanced displacements are already smaller than the maximum determined.

The development of the displacements $\bar{s}_i(l)$ of the sensors, the resulting actuator forces $\bar{F}_{act,i}(l)$ and the mapping matrix $\phi^T(l)$ during the adaptation process are shown for the cases $q = q_2$ and $q = q_5$ in Figures B.1(a) to B.2(c) and the results can be

connected to the observations in Figure 6.1. Figure B.1(a) shows for the case of $q = q_2$ that the steady state displacements of sensor 3 and 4 are equalized with the ongoing adaptation process, as only these mapping matrix ϕ^T elements stay active (B.1(a)). In the case $q = q_5$ all sensor displacements are equalized (see Figure B.2(a)). This equalization is a result of all sensor/actuator couples staying active.

6.2.2.2   Individual Coupling Parameters

Now individual coupling parameters are researched, meaning that $q_i(l)$ is according to (4.4)

$$q_i(l) = q_i = const \qquad \forall i = 1, ..., m \tag{6.12}$$

In this case, every coupling parameter q_i is set separately for the sensor/actuator couple i, determining directly the variance of the sensor displacement $E\{\bar{s}_i^2\}$ analog to (6.7). Figure 6.2 shows the variances of the adapted steel plate for three different setups $q_{set,j}$ of individual coupling parameters compared to the open loop variances $E\{\tilde{s}_i^2\}$ (blue marker). The solid lines show the real variance values $E\{\bar{s}_i^2\}$ of the sensor displacements and the dashed lines indicate the expected variances $E_{exp}\{\bar{s}_i^2\}$ determined by the coupling parameter sets $q_{set,j}$.

With increasing coupling parameters, the system becomes fully neocybernetic and every actuator is staying active. This is the case for parameter set $q_{set,3}$ (cyan line), where every sensor variance $E\{\bar{s}_i^2\}$ equals the predetermined variance $E_{exp}\{\bar{s}_i^2\}$. The parameter sets $q_{set,1}$ (green dashed line) and $q_{set,2}$ (red dashed line) lead to system states where not every mode is activated. Parameter set $q_{set,1}$ leads to a system where only actuator one and three are active. The other actuator forces are fading away. In the case of parameter set $q_{set,2}$ actuator four stays active as well and only actuator two becomes inactive. But for both setups it can be said that the variances of the inactive actuators remain smaller than the coupling factor would let it expect. Once again these lower variances can be explained as a side effect of the active actuators on the steel plate (see also section 6.2.2.1). For parameter set $q_{set,3}$ every actuator is activated, the system is fully neo-cybernetic and the expected and real variances are equal.

The development of the displacements $\bar{s}_i(l)$ of the sensors, the resulting actuator

PSfrag replacements
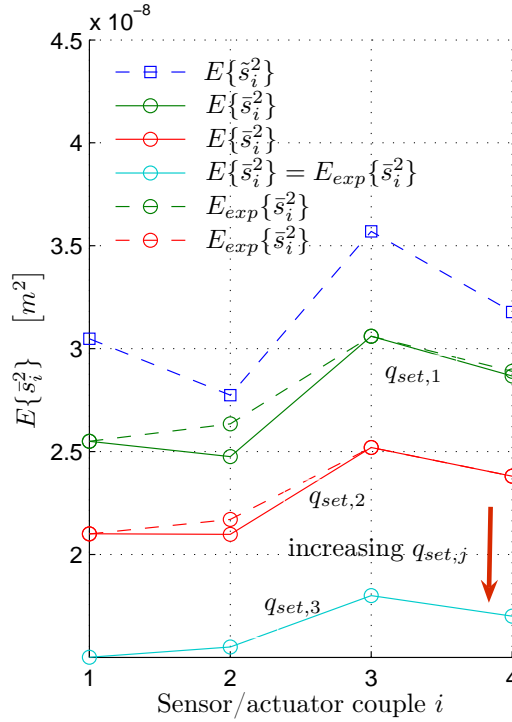
Figure 6.2: Influence of variable q_i factors on the displacement variances $E\{\bar{s}_i^2\}$.

forces $\bar{F}_{act,i}(l)$ and the mapping matrix $\phi^T(l)$ during the adaptation process are shown for the cases $q = q_{set,2}$ and $q = q_{set,3}$ in the figures B.3(a) to B.4(c). Here once again it can be seen that in the case of $q_{set,2}$ actuator force two is fading away as the respective mapping matrix entry is zeroed in the adapted system (figures B.3(b) and B.3(c)). In the case of $q_{set,3}$ every actuator is activated (see Figure B.4(b)) and the steady state variances $E\{\bar{s}_i^2\}$ are directly determined by the values of the individual coupling parameters $q_{set,3}$.

Analog to the thoughts for the maximum sensor displacements in the adapted system from section 6.2.2.1 there follows with (6.11)

$$\max |\bar{s}_i| \leq \sqrt{\frac{1}{q_i}} \qquad \forall i = 1, ..., m \tag{6.13}$$

determining the maximum expected displacement of the single sensors i in the case of differing coupling parameters.

6.2.2.3   Adaptive Coupling Parameters

In this section, adaptive learning coupling parameters are researched, meaning that $q_i(l)$ is according to (4.4)

$$q_i(l) = \frac{\nu}{E\{\bar{F}^2_{act,i}\}} \qquad \forall i = 1, ..., m \tag{6.14}$$

It turns out that this strategy keeps all actuators active, as the coupling coefficients are growing, when the actuator forces are decreasing due to the factor $E\{\bar{F}^2_{act,i}\}$ in the denominator. The fading actuators regain life again. Simulations show that this applied control strategy results in a balanced neocybernetic system for every adaptation process and the variances of the sensor displacements are determined by the steady state values of the adapted coupling coefficient $\bar{q}_i = q_i(l_b)$. Figure 6.3 shows the variances $E\{\bar{s}_i^2\}$ of the adapted system sensor/actuator couples and the influence of the free design parameter ν. It can be seen that an increasing ν factor leads to a steel plate adaptation with less variance. Furthermore, it can be remarked that the original variance structure is generally kept, no matter what value ν has. One could say that the adaptive coupling parameters damp the variances in the system proportional to the original variances.

Figures B.6 and B.7 show the development of the mapping matrix entries $\phi_{ii}^T(l)$ for two different control parameters ν. Remarkable is the fact that the adaptation of the steel plate leads to equal steady states for all matrix entries.

Figures B.5(a) to B.5(c) show the sensor displacements $\bar{s}_i(l)$, the development of the actuator forces $\bar{F}_{act,i}(l)$ and the development of the adaptive coupling parameter $q_i(l)$ during the adaptation process of the steel plate for $\nu = 5 \cdot 10^{12}$. The adaptation of the steel plate turns out to be very smooth, as there is no real "competition" between the single sensor/actuator couples. This can be seen for the individual and constant coupling coefficients (see for example B.4(c)).

Compared to the other two sorts of coupling parameters $q_i(l)$, this system has the advantage that every sensor/actuator couple of the distributed network is actively taking part in the control of the steel plate. On the other hand, as the parameters $q_i(l)$ are adaptive it is somehow difficult to determine, how the balanced system is
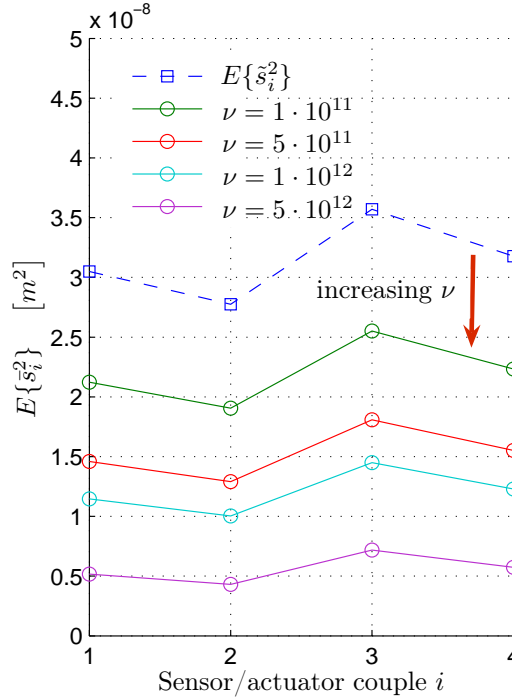
Figure 6.3: Influence of adaptive coupling factors q_i on the displacement variances $E\{\bar{s}_i^2\}$.

converging. One does not know beforehand how the parameters will develop. One knows only that it will converge with every actuator staying active.

The next section shows simulation results in case of varying external forces that is better applicable in real systems.

6.2.3 Varying External Forces

The same mathematical thoughts that are holding in the case of constant forces are also valid for variable external forces, applied on the steel plate during the adaptation process and the equations from Section 6.2.1 are still valid. It turns out that the results in the case of varying forces are the same, compared to the case of constant forces. Although this adaptation is closer to real-life systems as the environment is not expected to be constant during the adaptation but changing. The system is adapting with respect to the environmental variations and emerges towards maximum experienced stiffness.
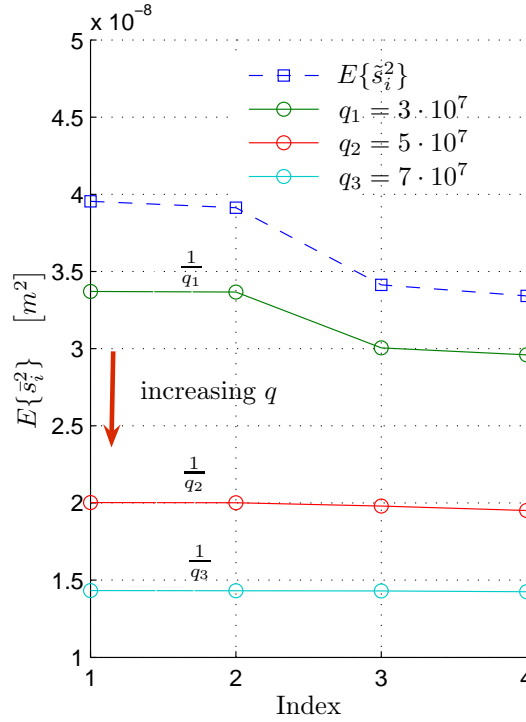
Figure 6.4: Influence of constant q factors on the equalization of the displacement variances of $E\{\bar{s}_i^2\}$ in the case of variable external forces.

As the results of the adaptation are similar to constant forces, the simulations are only performed for constant coupling parameters q and the outcomes are presented.

It is mentioned in Section 4.3 that the breaking criterion for the algorithm has to be changed slightly, if varying external forces are considered for the adaptation of the system. The problems are higher variations for the mapping matrix entries $\phi_{ii}^T(l)$ and therefore past information should be regarded as well for the breaking criterion. The adaptation of the steel plate is finished if the maximum mean value difference over the last w values is smaller than the given mean value $\overline{\Delta\phi}$ and analog to (4.17) the modified breaking criterion is

$$\max_i\left\{\frac{1}{w}\left(\sum_{j=l_b-w+1}^{l_b+1}\phi_{ii}^T(j) - \sum_{k=l_b-w}^{l_b}\phi_{ii}^T(k)\right)\right\} < \overline{\Delta\phi} \qquad i = 1, ..., m \qquad (6.15)$$

Figure 6.4 shows the development of the variances $E\{\bar{s}_i^2\}$ of the adapted system compared to the original undisturbed environmental variances $E\{\tilde{s}_i^2\}$ (blue markers) depending on the coupling coefficient q for the mapping matrix ϕ^T. Here the devel-

opment is analog to Section 6.2.2:

With increasing q factors the variances are getting equalized again. For $q = q_1$ only two actuator variances equal. The other two variances are already lower than the q factor would determine analog to the results for the constant external forces. Once again, these actuator forces are zeroed during the adaptation process. For $q = q_2$ the variances are getting closer together and in case of the third adapted system $q = q_3$ equalization is reached again and all four actuators are occupied.

Figures B.8(a) to B.8(c) show exemplary the development of the sensor displacements $\bar{s}_i(l)$, the actuator forces $\bar{F}_{act,i}(l)$ and the mapping matrix $\phi_{ii}^T(l)$ for the use of $q = q_3$.

In the case of varying external forces, a connection can also be made between the steady states of the displacements $\bar{s}_i$ after the adaptation and the coupling parameter q in the case of equalization of all variances. It turns out that the expectation value of all sensor displacements $E\{\bar{s}_i\}$ can be determined analog to (6.11) as

$$\max_i |E\{\bar{s}_i\}| \leq \sqrt{\frac{1}{q}} \quad \forall i = 1, ..., m \tag{6.16}$$

6.3   Outlook: Non-Localized Learning

In order to manifest the theoretical thoughts on elastic systems, the idea of decentralized learning with communication structures between the sensor/actuator couples is addressed shortly. Firstly, mathematical considerations are presented and simulation results are shown in case of constant coupling parameters q.

6.3.1   Mathematical Considerations

If the sensor/actuator couples in the distributed network take also the signals of the other couples into account, the mapping matrix ϕ^T has cross-correlated entries and can be written as

$$\phi^T = Q \cdot E\{\bar{F}_{act}\bar{s}^T\} = Q \cdot \begin{pmatrix} E\{\bar{F}_{act,1}\bar{s}_1\} & \cdots & E\{\bar{F}_{act,1}\bar{s}_m\} \\ E\{\bar{F}_{act,2}\bar{s}_1\} & \cdots & E\{\bar{F}_{act,2}\bar{s}_m\} \\ \vdots & & \vdots \\ E\{\bar{F}_{act,m}\bar{s}_1\} & \cdots & E\{\bar{F}_{act,m}\bar{s}_m\} \end{pmatrix} \tag{6.17}$$

For the algorithmic implementation every matrix entry $e_{ij}(l) = E\{\bar{F}_{act,i}\bar{s}_j\}$ has to be initialized analog to (4.13) with some small value $\epsilon_{ij} < 0$. The update of the covariance matrix $E\{\bar{F}_{act}\bar{s}^T\}(l)$ is now parallel to (2.19) not for every single sensor/actuator but the complete network at once with the found steady state forces $\bar{F}_{act}(l)$ and displacements $\bar{s}(l)$ of the first order iteration

$$E\{\bar{F}_{act}\bar{s}^T\}(l+1) = \lambda E\{\bar{F}_{act}\bar{s}^T\}(l) + (1-\lambda)\bar{F}_{act}(l)\bar{s}^T(l) \qquad (6.18)$$

Here, the simplifications made for the diagonal learning in Section 6.2.1 are not valid any longer, but (6.2) determines how the emergent system should look like:

The coupling coefficients q_i determine directly the eigenvalues of the covariance matrix of the balanced environment $E\{\bar{s}\bar{s}^T\}$ and the system is truly neocybernetic, if (6.3) holds.

The emergence of a steel plate adaptation in this case is shown in what follows. In order to gain interpretable results, variation for the external forces $F_{ext}(l)$ is taken into account.

6.3.2   Constant Coupling Parameters

An exemplary global learning system is evaluated in the case of constant coupling coefficients and the challenges occuring in the simulation are presented shortly.

Figure 6.5 shows exemplary the eigenvalues of the open loop system, determined by the covariance matrix of the undisturbed environment $E\{\tilde{s}\tilde{s}^T\}$ and the emerging system behavior for different constant coupling coefficients q of the mapping matrix. It can be seen directly that the q factor determines the variation in the balanced environment. With increasing q there is a loss in variation on the one hand but on the other hand, equalization of variations emerges. From (6.3) it can be derived that the coupling coefficient determines the number of modes that are behaving neocybernetic as this equation must hold, if the q factor is high enough to capture all different variations. For $q = q_1$ only the first eigenvalue is determined by the coupling coefficient as the other eigenvalues remain. For $q = q_2$ and $q = q_3$ the first two eigenvalues are equalized, as the coupling coefficient q is chosen high enough that the second eigenvalue of the undisturbed environment covariance matrix becomes neocybernetic
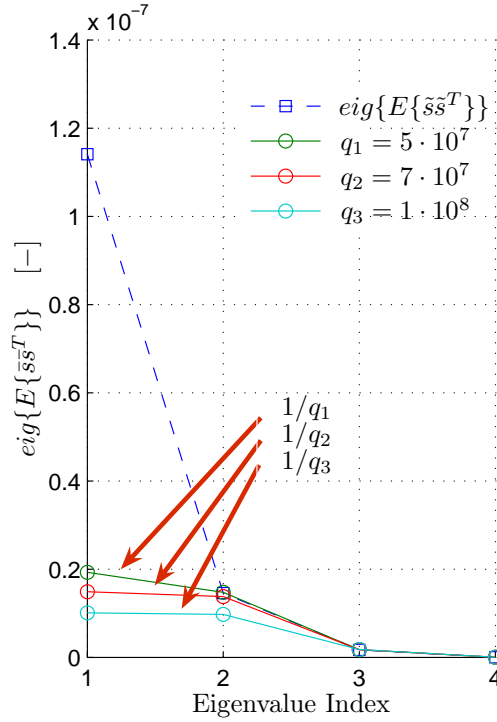
Figure 6.5: Influence of constant q factors on the equalization of the balanced covariance eigenvalues of $E\{\bar{s}\bar{s}^T\}$ in the case of a full matrix ϕ^T.

as well and there holds

$$\lambda_2 \cdot q > 1 \tag{6.19}$$

analog to (6.3). Table 6.1 shows an overview for the eigenvalues of the undisturbed environment covariance matrix and the balanced outcomes and compares the expected values, determined by the factor $1/q_i$, with the real variances in the balanced environment. The occurring error can be explained with numerical uncertainties of the simulation within Femlab and Matlab. The accuracy of the outcomes is still satisfactory.

If one has a closer look at the eigenvalues of the open loop system in Figure 6.5 (blue marker) one can see that most of the variance in the system is mainly determined by one dominant eigenvalue. The second one is still high enough to validate the neocybernetic behavior for these two modes of the system as (6.3) is fulfilled for λ_2. The lasting eigenvalues are negligible. This fact can be explained with the rigidity of the researched steel plate. As the shape is very simple the variation of the measured sen-

	$\lambda(10^{-7})$	$\bar{\lambda}_{q_1}(10^{-7})$	$\bar{\lambda}_{q_2}(10^{-7})$	$\bar{\lambda}_{q_3}(10^{-7})$	max. error
1	1.141	0.193	0.149	0.101	4.2%
2	0.146	0.147	0.138	0.097	3.5%
3	0.017	0.017	0.017	0.018	–
4	≈ 0	≈ 0	≈ 0	≈ 0	–
$\frac{1}{q_i}$	–	0.200	0.143	0.100	–
# neocybernetic modes	1	2	2		–

Table 6.1: Comparison of open and closed loop eigenvalues for different coupling coefficients q. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (see Figure C.2), control parameter "Set 1" (Table C.7)

sor deformations is rather limited. In order to prove that more neocybernetic modes could be activated, very high coupling coefficients q would be necessary to direct the system in the balanced area of the remaining small eigenvalues. These higher coupling coefficients are not useful in simulations as the problem of numerical instability introduced in Section 5.1 occurs and the discrepancy between the calculated balanced results and the Femlab simulations is too high.

A solution for this problem would be a research on complexer steel plate shapes, where more variation of data can be expected and the smallest eigenvalue of the undisturbed environment is high enough to be adaptable as well with the correct choice of the coupling coefficient q.

6.4 Interpretations

Finally, the observed simulation results are put in a wider framework and interpreted from three different point of views. The first field of interest is the system theoretic view on elastic systems, followed by the application of these results for control purposes and at last the evolutionary fitness of elastic systems is described closer.

6.4.1 System Theoretic Approach

The idea of elastic systems is rather new and so far existent on paper and validated in simple simulations. In order to manifest the theoretic ideas of Chapter 3 the idea

of global learning is introduced in Section 6.3, meaning that the mapping matrix ϕ^T is full and the sensor/actuator couples forming the distributed network have all information available.

The section shows that the adaptation of the system follows the guidelines introduced for elastic systems. It can be seen that the coupling coefficient q determines directly the number of activated modes in the system and therefore the loss of excitation. In addition it can be seen that there is not only a loss of variation but also an equalization; with proper coupling coefficients q all variations can be equalized. Furthermore it turns out that the control variables can be taken in order to replace the latent system variables as proposed in Section 3.3.3.

In summary it can be said that the theoretic framework of elastic systems can also be proved in complexer simulations as the proposed steel plate adaptation process indicates.

6.4.2 Neocybernetic Control

As one compares the neocybernetic control of the distributed sensor/actuator network to conventional control mechanisms, some advantages of this new thinking can be remarked.

Conventional control purposes for complex systems are mainly based on SISO controls or sophisticated multivariate techniques are applied. Using distributed controlling, there is always the question how the communication of single sensor/actuator couples should be realized in order to reach an emergent goal. The entities could also compete or disturb each other, worsening the overall global result. In the case of the proposed elastic control algorithm, these problems seem to vanish. Emergent behavior in the system can be achieved with only local information available, as the sensor/actuators are communicating implicitly through the environment. The system adaptation is only taking local actuator forces and local deformations into account. In addition, the proposed neocybernetic control does not need explicit knowledge about the actual system dynamics or an appropriate model of the system. As one concentrates more on the pattern view than on the process itself statistical properties can be applied and the multivariate distributed controllers are adapting data-based and learning from

different variations in the observable environment completely locally [8].

Anyhow, the results of this local adaptation processes show emergent behavior in the system. In the case of the steel plate adaptation processes the coupling parameter q determines directly the expectable displacements of the sensors in finally balanced systems and therefore the emergent shape of the plate.

6.4.3   Evolutionary Fitness

Going back to the evolutionary motivation of the control formula, the results of the steel plate adaptations can also be explained in a wider sense.

The proposed algorithm optimizes the steel plate towards its maximum experienced stiffness, meaning that it is adapting in a way where environmental variations are trying to be minimized. From the simulation results it can be seen that sensor/actuator couples with higher measuring displacements are intuitively developing higher forces to counteract these deformations. Reason for higher displacements can be the position of the sensor/actuator couples on the plate and the impact of external forces on these controllers. In this context the fading away of actuator forces during the adaptation process can also be explained more intuitively; Controllers with less significance have not much influence on the overall stiffness of the system. Sensor/actuator couples closer to the "source of concern" or with better position for compensations are developing stronger. This control can be related intuitively to evolutionary fitness in biological systems: the human skin is getting thicker in areas that are used more intensive and more often (finger tips of guitar players, soles of feet, ...), muscles of human beings or animals are getting better evolved, if they are used more intense and so on. In biology this phenomenon is known as hyperplasia and hypertrophy [23].

Even if the ongoing process in the case of the steel plate adaptation is active, the emergent result can be implemented passively, by adding extra layers at these positions, where the actuators evolve high forces during the adaptation. It has to be remarked that this passive control strategy demands different types of sensors (strains or stresses) to make the results more practicable.

Chapter 7

Summary and Outlook

The thesis studied the possibilities of implementing new kinds of sensor/actuator systems in order to control deformable systems. The new approach concentrates on distributed networks where sensor/actuators are acting completely local and still emergent global behavior evolves out of these actions. For this approach neocybernetic tools were applied.

First the general ideas of neocybernetic modeling of complex real-life systems were introduced. The question was, how self-organization and self-regulation can emerge in systems out of local interactions among low level actuators. After introducing the key ideas of the used mathematical tools in this context, Hebbian and anti-Hebbian learning was presented as a possible modeling strategy.

The next chapter considered a new point of view for modeling neocybernetic systems that finally fell back into known domains of self-regulation and self-organization of neocybernetic systems in the sense of stabilization and PSA. Starting point were thoughts about the evolutionary fitness of systems that are in strong interconnection with their environment. Additionally it turned out that active control structures can be implemented for control purposes of elastic systems. This basic idea was evaluated closer in the next chapter.

With the assumption of totally local activities of sensor/actuator couples, a distributed control of deformable systems could be built up and researched closer with

simulations. The used tools were Matlab and Femlab. It turned out that the highly iterative simulation process demanded a new simulation strategy where the strong connection between the used simulation tools Femlab and Matlab was loosened. The next chapter introduced this new algorithm. The advantages of the new algorithm were mainly the time efficiency and the possibility of by-passing problems of numerical instability, which made it possible to research a wider area of setups faster.

The simulation results shown in the next chapter, revealed that global emergent behavior can be seen after the adaptation process of the steel plate, even if the learning strategy of the sensor/actuator couples was completely local and no explicit communication between the sensor/actuator couples was implemented. The outcomes were shown for different coupling parameters and external forces. First only constant deformations were taken into account, as the results are easily expandable to adaptation processes, where varying initial deformations were used. The results show generally a loss of excitation in the system and the remaining variations can be predicted directly by the used coupling parameters. This means that the emergent shape of the steel plate can be determined beforehand with the coupling parameters. As the steel plate adaptation was also used to strengthen the theoretical framework about elastic systems some more simulations were made, with assumed communication between the sensor/actuator couples. Finally, the results were interpreted from the system theoretic point of view, for control purposes and — in a wider sense — from the evolutionary point of view.

As the adapted steel plate results can be used for passive control by adding for example some extra layers at the proper positions in order to increase the stiffness of the system, some more simulations can be made with different types of sensors (for example stresses and strains) to gain more intuitions. In addition to that more complex real applicable steel structures could be researched. Thinking of real-life applications, it is conceivable to use the distributed sensor/actuator network directly in practical applications in order to perform fatigue or vibration analysis for mechanical systems.

Appendix A

Charts for Algorithm Validation
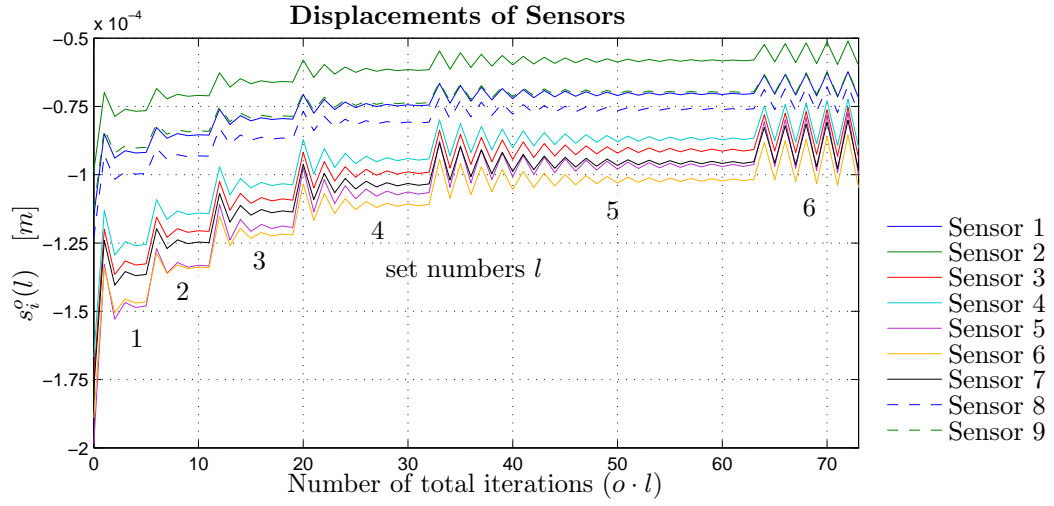
(a) Total Displacements of sensors during adaptation process.
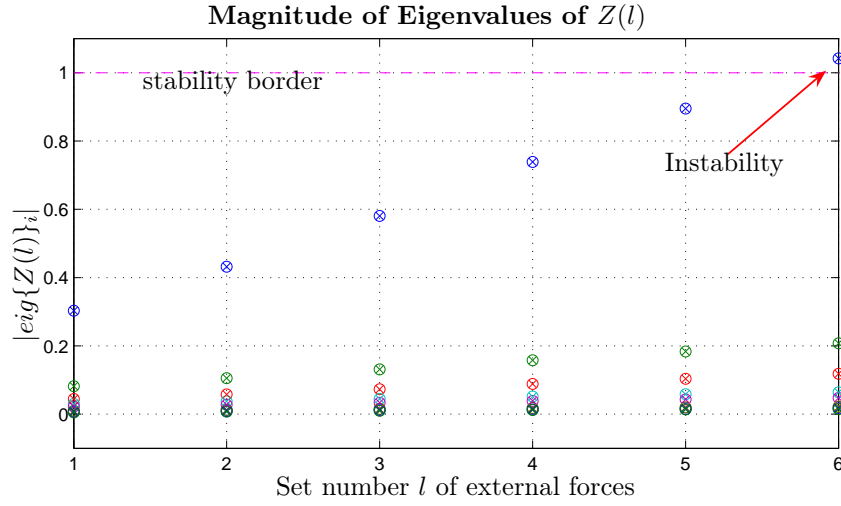
(b) Magnitude of eigenvalues of matrix $Z(l)$ during adaptation process.

Figure A.1: Predictability of first order instability with evaluation of eigenvalues of matrix $Z(l)$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.1), control parameters "Set 1" from Table C.3.
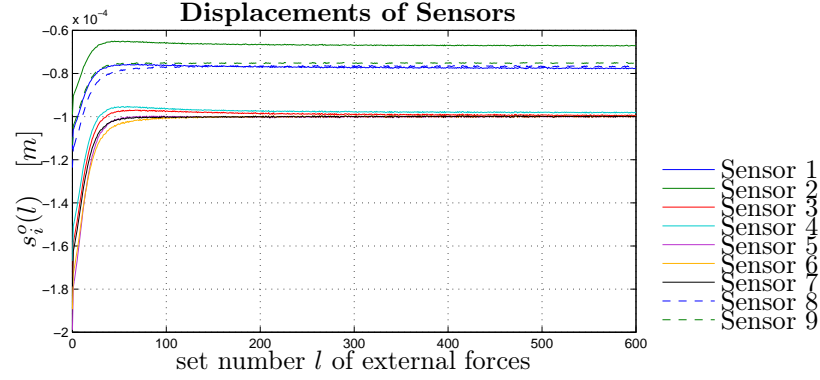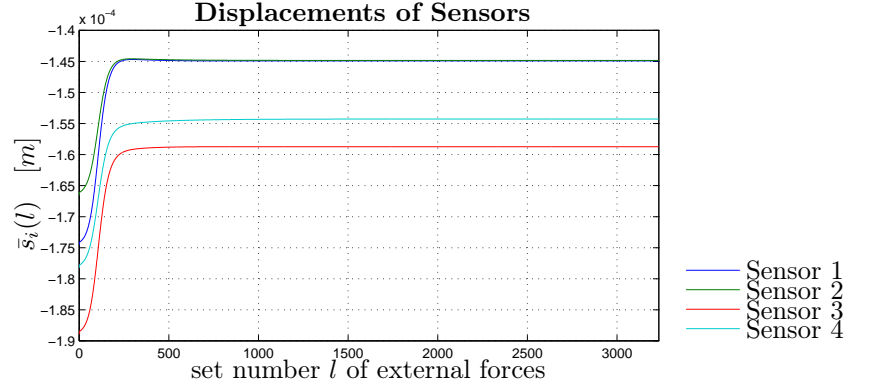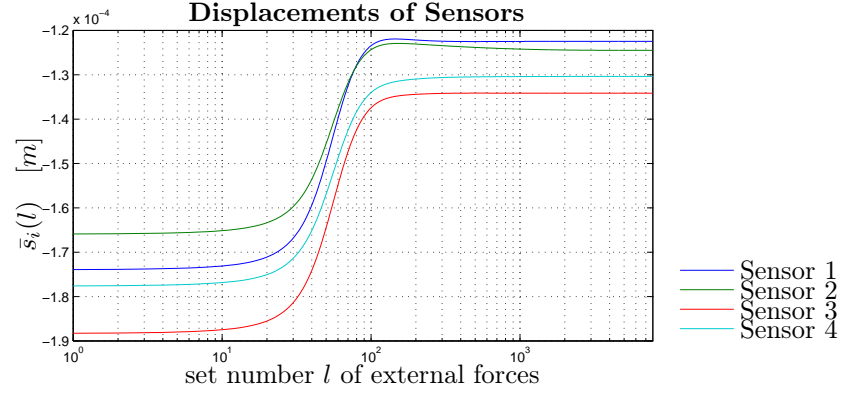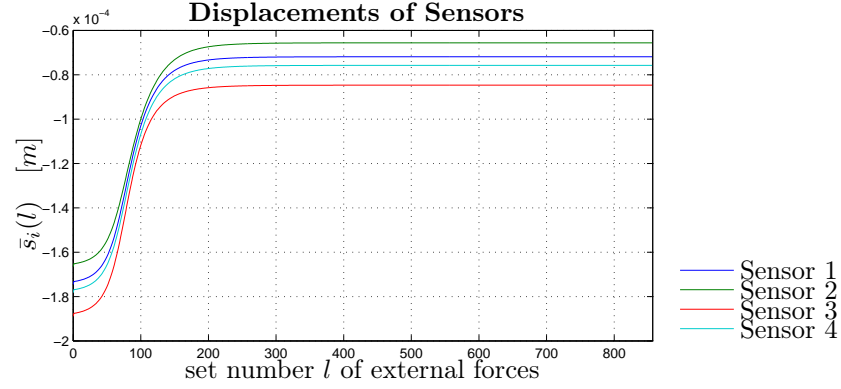
(a) Measured displacements of sensors.
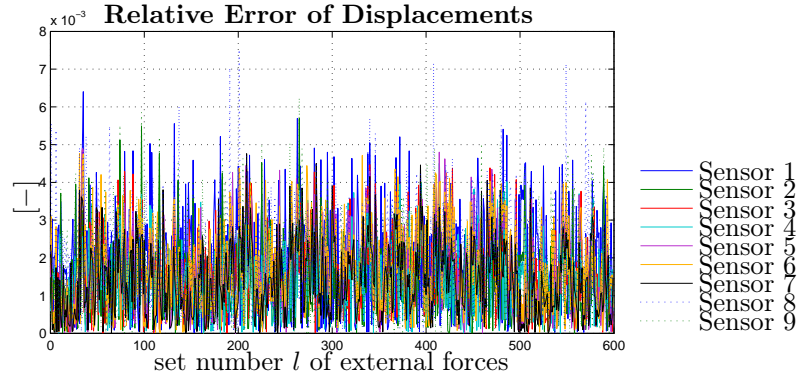
(b) Relative error of measured and calculated sensor displacements .
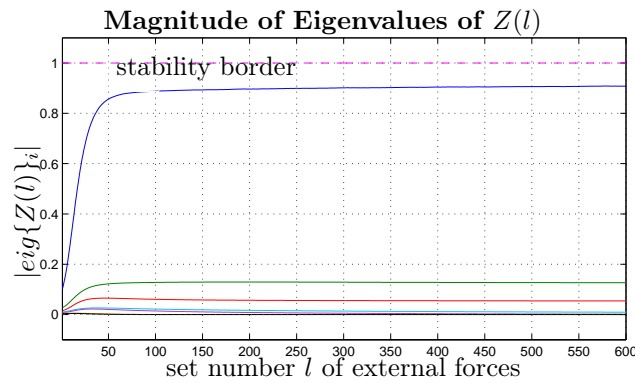
(c) Magnitude of eigenvalues of matrix $Z(l)$.

Figure A.2: Comparison of conventional adaptation algorithm (flowchart 4.3) with new adaptation (flowchart 5.4) for a stable process. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.1), control parameter "Set 2" (Table C.3).
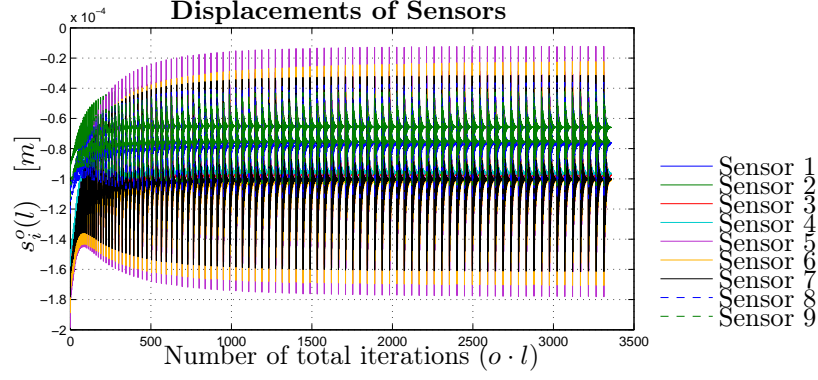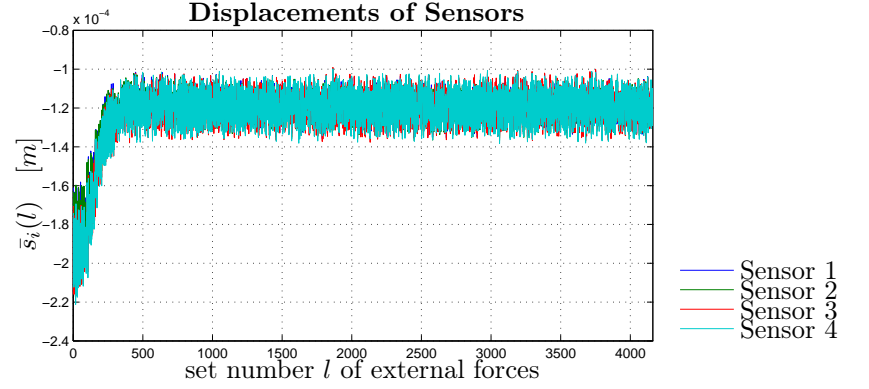
(a) Total displacements of sensors with $s^0(l) = s_g + HNF_{ext}(l)$.
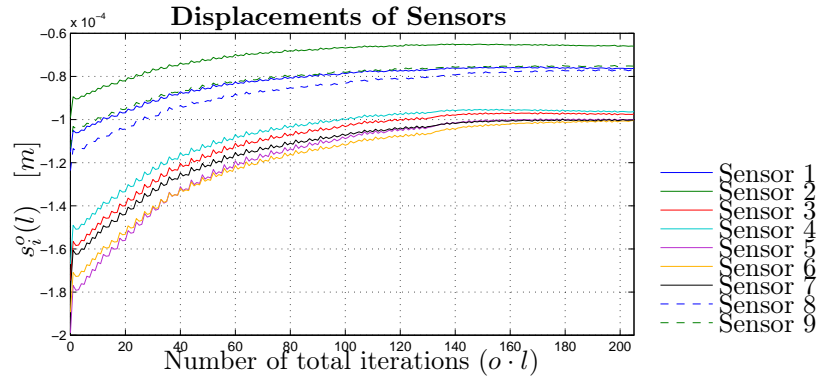
(b) Total displacements of sensors with $s^0(l) = \bar{s}(l-1)$.
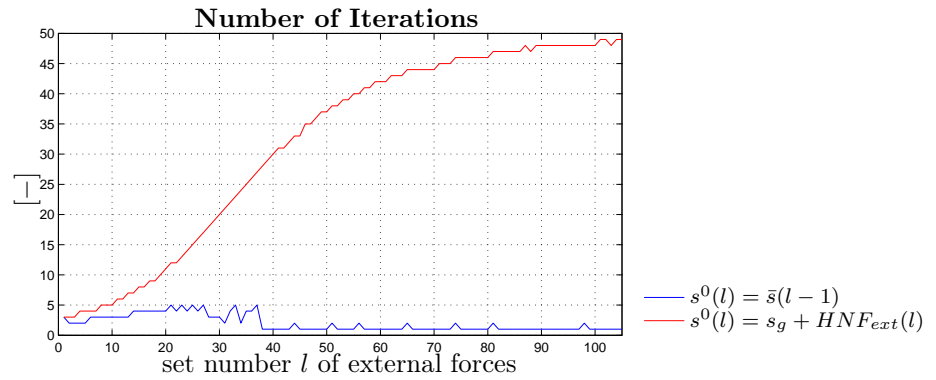
(c) Number of First Order Iteration Steps.

Figure A.3: Comparison of different inital first order iteration conditions. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.1), control parameter "Set 2" (Table C.3).
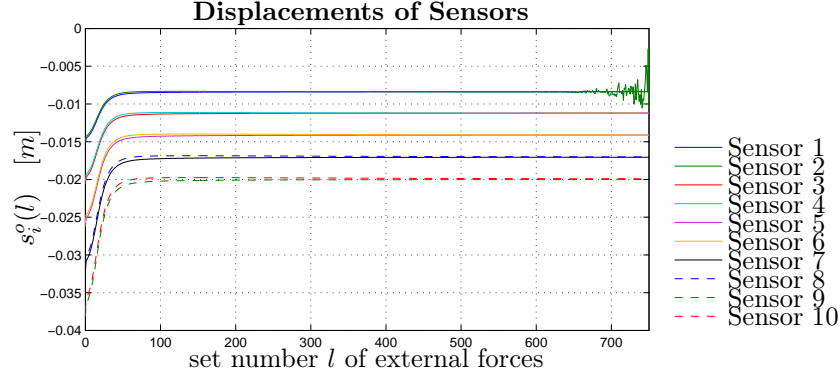
(a) Displacements of sensors.
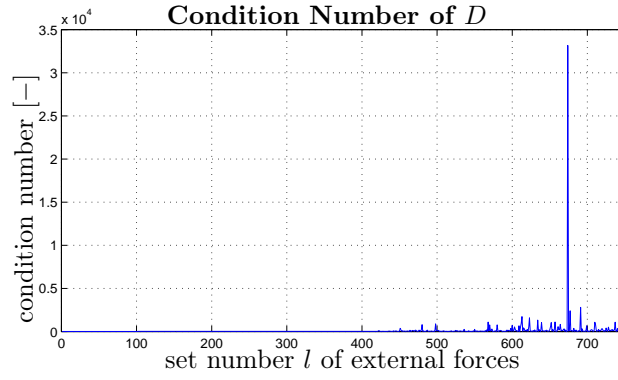
(b) Condition number of transformation matrix D.
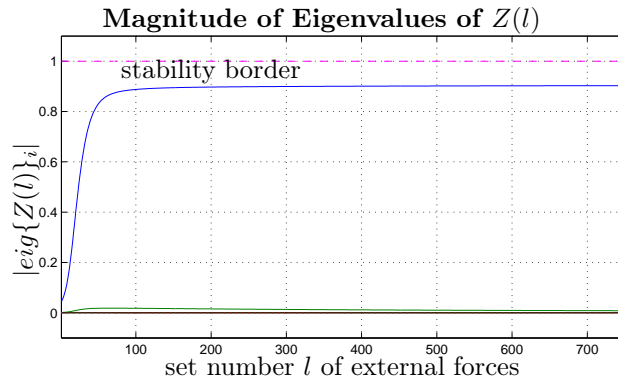
(c) Magnitude of eigenvalues of matrix $Z(l)$.

Figure A.4: Inaccuracy of closed form first order iteration. Simulation setup: "Spring Board" (Table C.2), sensor/actuator and external forces positions (Figure C.3), control parameter "Set 3" (Table C.3).

PSfrag replacements

(a) Displacements of sensors.

PSfrag replacements
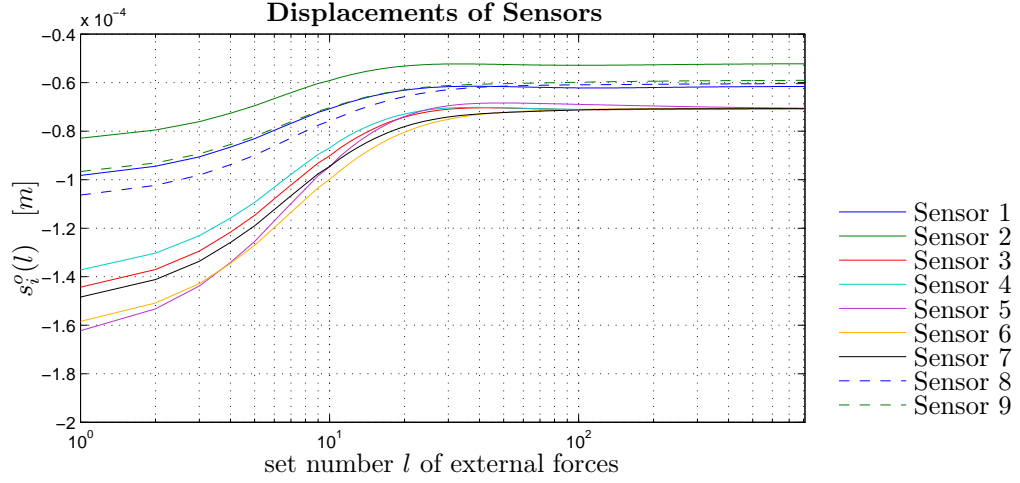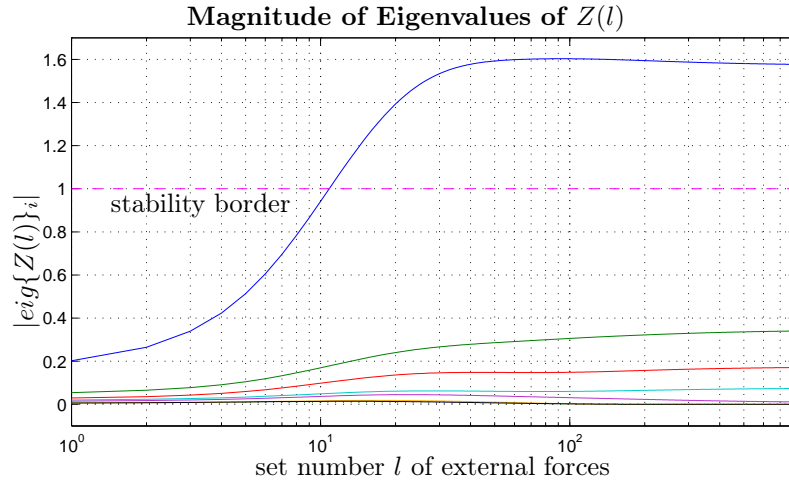
(b) Magnitude of eigenvalues of matrix $Z(l)$.

Figure A.5: Extension of the conventional algorithm for instable first order iterations. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.1), control parameter "Set 4" (Table C.3).

PSfrag replacements

Displacements of Sensors

(a) Displacements of sensors.

PSfrag replacements
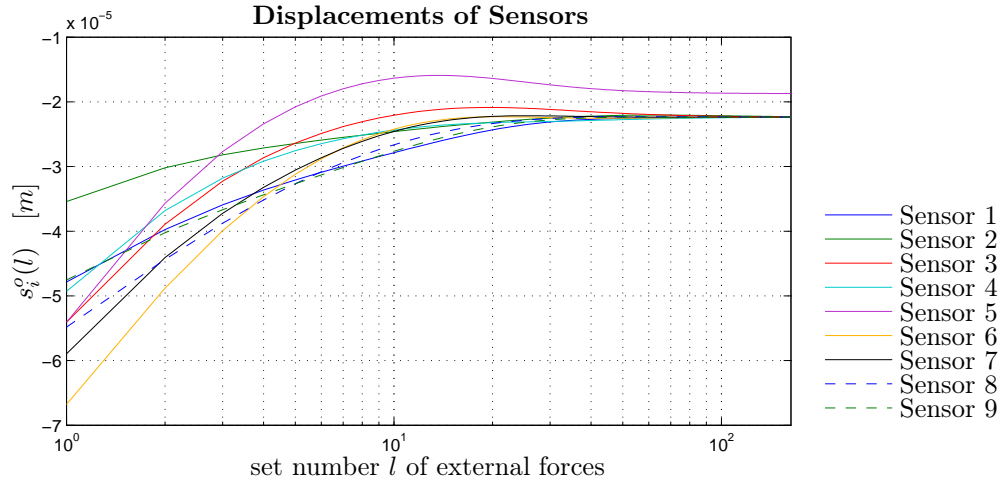
Magnitude of Eigenvalues of $Z(l)$

(b) Magnitude of eigenvalues of matrix $Z(l)$.

Figure A.6: Extension of the conventional algorithm for instable first order iterations. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.1), control parameter "Set 5" (Table C.3).

Appendix B

Simulation Charts

PSfrag replacements

(a) Displacements of sensors.

PSfrag replacements

(b) Forces of actuators.

PSfrag replacements
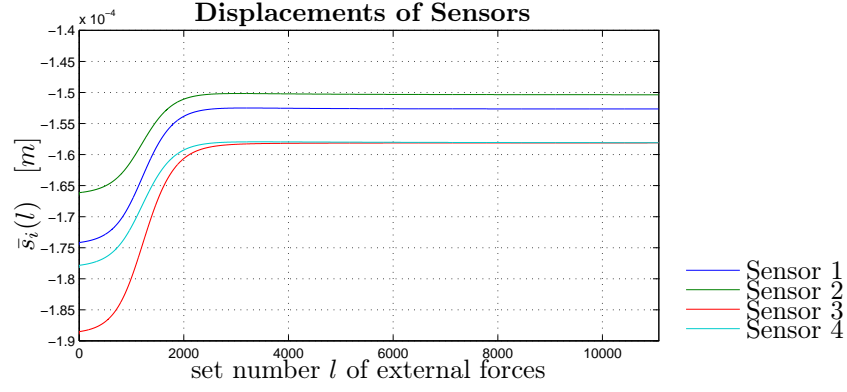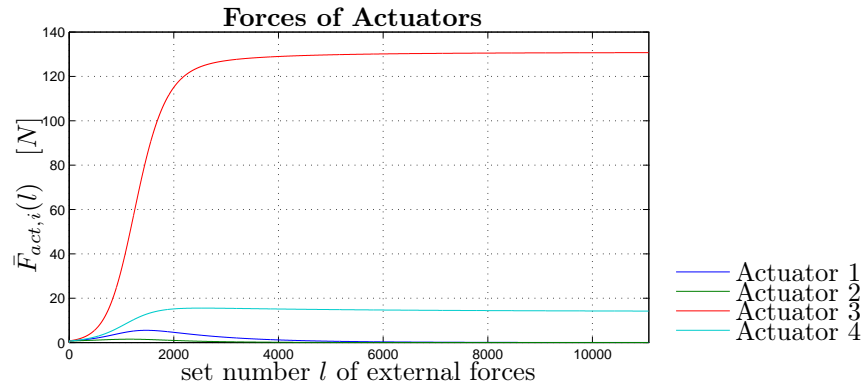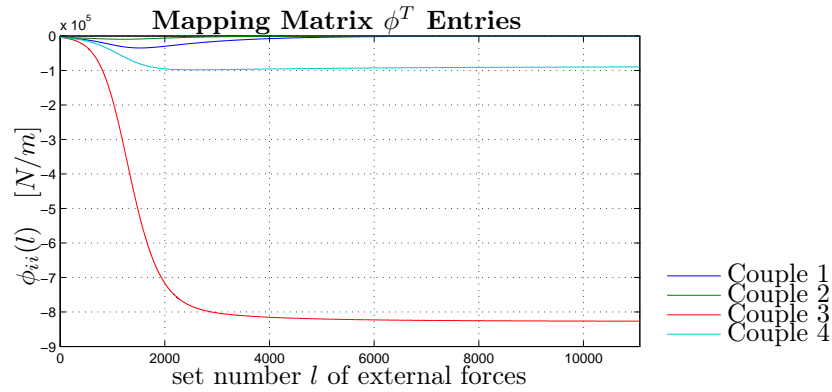
(c) Mapping matrix elements.

Figure B.1: Adaptation process of the steel plate for $q = 4 \cdot 10^7$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.2), control parameter "Set 6" (Table C.3)

PSfrag replacements

(a) Displacements of sensors.

PSfrag replacements

(b) Forces of actuators.

PSfrag replacements
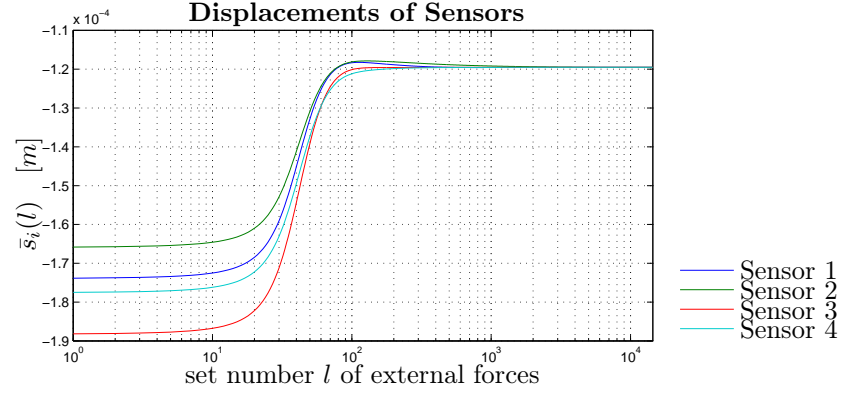
(c) Mapping matrix elements.

Figure B.2: Adaptation process of the steel plate for $q = 7 \cdot 10^7$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.2), control parameter "Set 7" (Table C.3)

PSfrag replacements

(a) Displacements of sensors.

PSfrag replacements

(b) Forces of actuators.

PSfrag replacements

(c) Mapping matrix elements.

Figure B.3: Adaptation process of the steel plate for $q = q_{set,2}$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.2), control parameter "Set 1" (Table C.4)

PSfrag replacements

(a) Displacements of sensors.

PSfrag replacements

(b) Forces of actuators.

PSfrag replacements
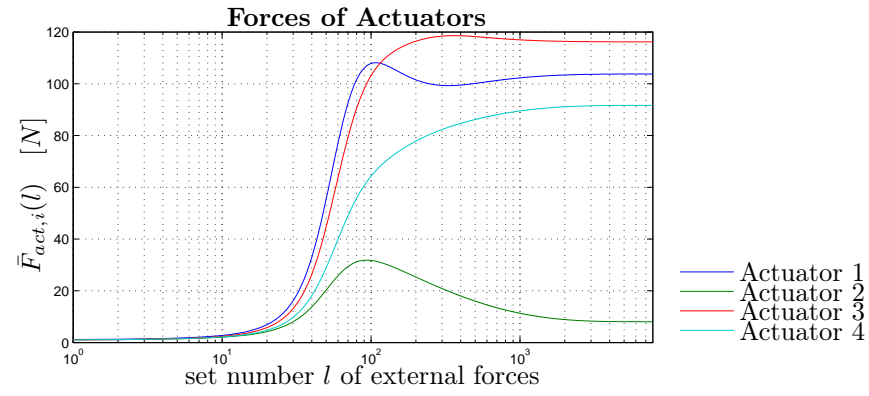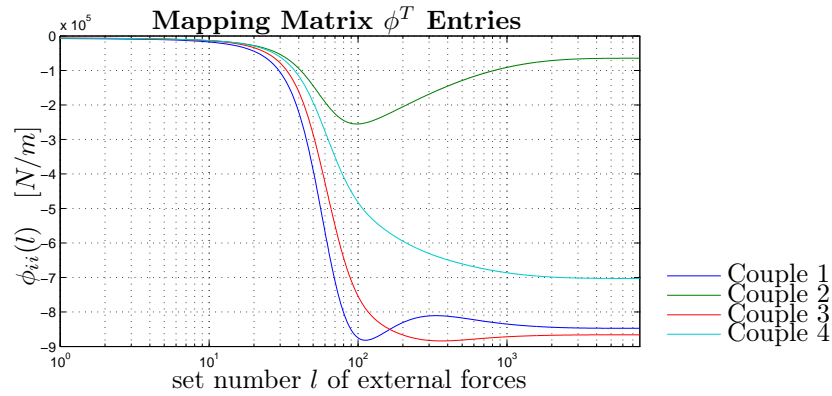
(c) Mapping matrix elements.

Figure B.4: Adaptation process of the steel plate for $q = q_{set,3}$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.2), control parameter "Set 2" (Table C.4)

PSfrag replacements

(a) Displacements of sensors.

PSfrag replacements

(b) Forces of actuators.
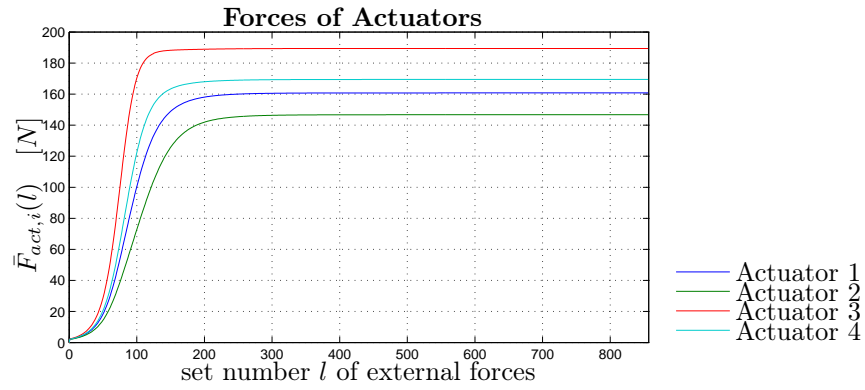
PSfrag replacements
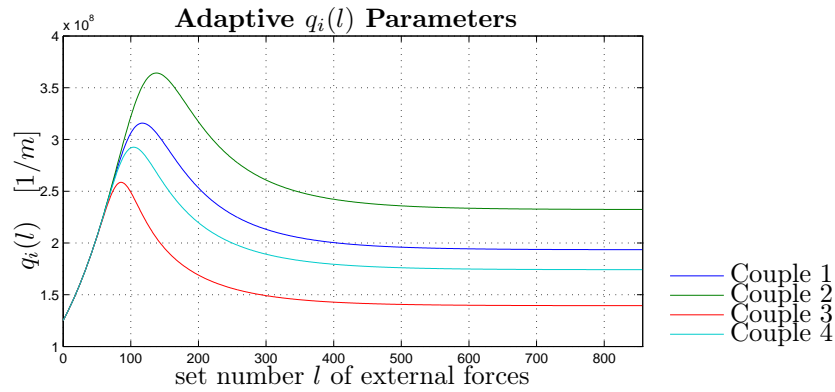
(c) Adaptive coupling parameters.

Figure B.5: Adaptation process of the steel plate for adaptive coupling parameters and $\nu = 5 \cdot 10^{12}$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.2), control parameter "Set 2" (Table C.5)

PSfrag replacements
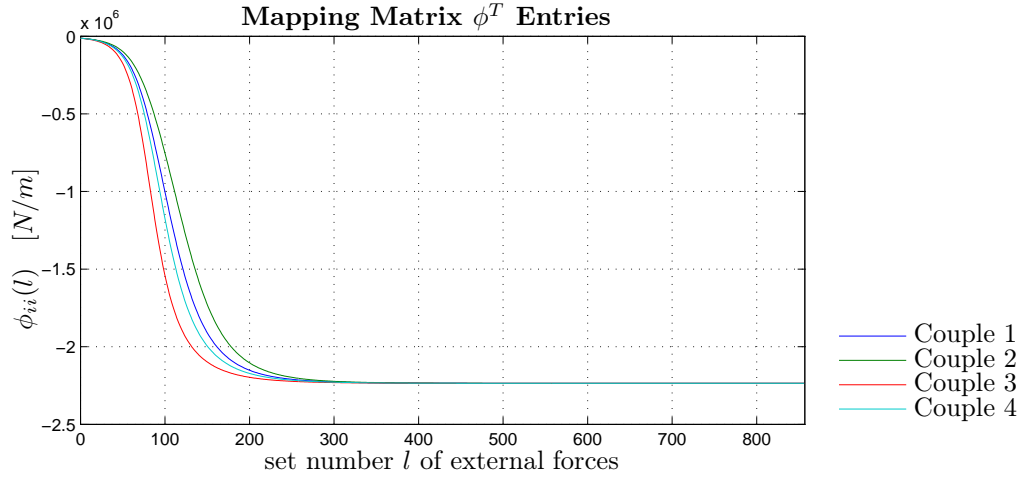
Figure B.6: Adaptation of mapping matrix entries $\phi_{ii}^{T}(l)$ $\nu = 5 \cdot 10^{12}$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.2), control parameter "Set 2" (Table C.5)

PSfrag replacements
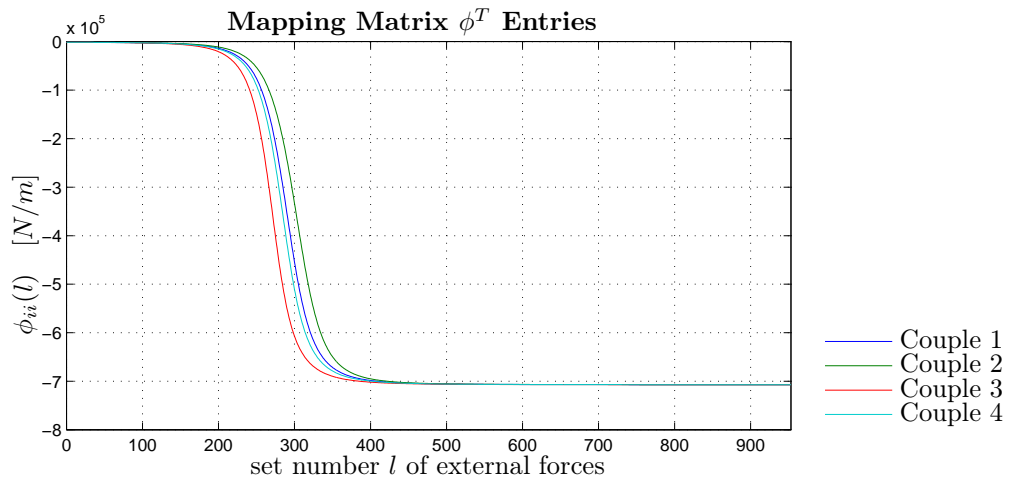
Figure B.7: Adaptation of mapping matrix entries $\phi_{ii}^{T}(l)$ for $\nu = 5 \cdot 10^{11}$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.2), control parameter "Set 1" (Table C.5)

PSfrag replacements

(a) Displacements of sensors.

PSfrag replacements

(b) Forces of actuators.

PSfrag replacements
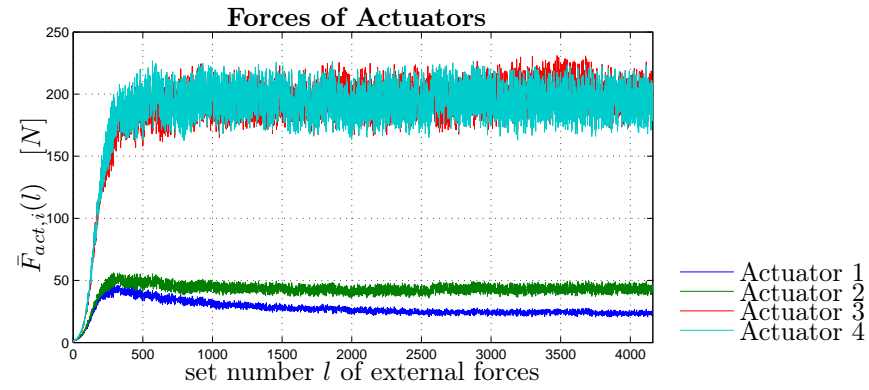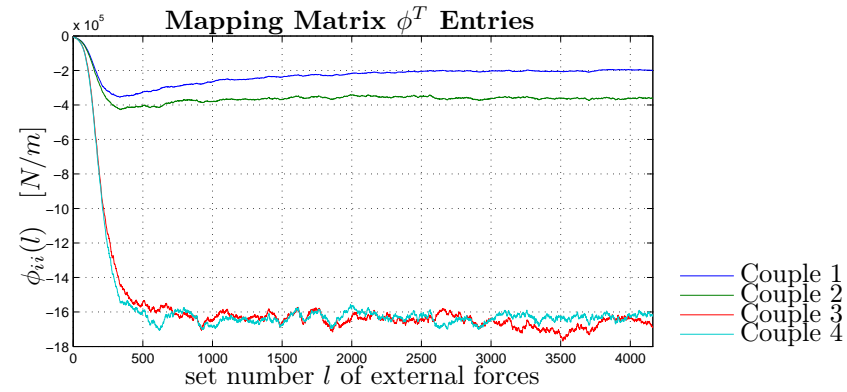
(c) Mapping matrix elements.

Figure B.8: Adaptation process of the steel plate for $q = 7 \cdot 10^7$ and variable external forces $F_{ext}(l)$. Simulation setup: "Quadratic Steel Plate" (Table C.2), sensor/actuator and external forces positions (Figure C.2), control parameter "Set 1" (Table C.6)

Appendix C

Layouts and Parameter Sets

Distribution of sensor/actuator couples and external forces
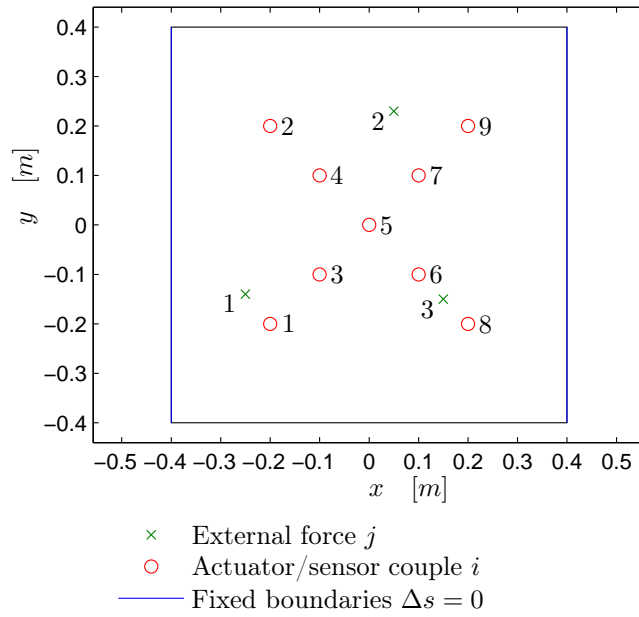
Figure C.1: Steel Plate set with nine sensor/actuator pairs and three external applied forces positions.

PSfrag replacements

Distribution of sensor/actuator couples and external forces

PSfrag replacements

Figure C.2: Steel Plate set with four sensor/actuator pairs and three external applied forces positions.

Distribution of sensor/actuator couples and external forces
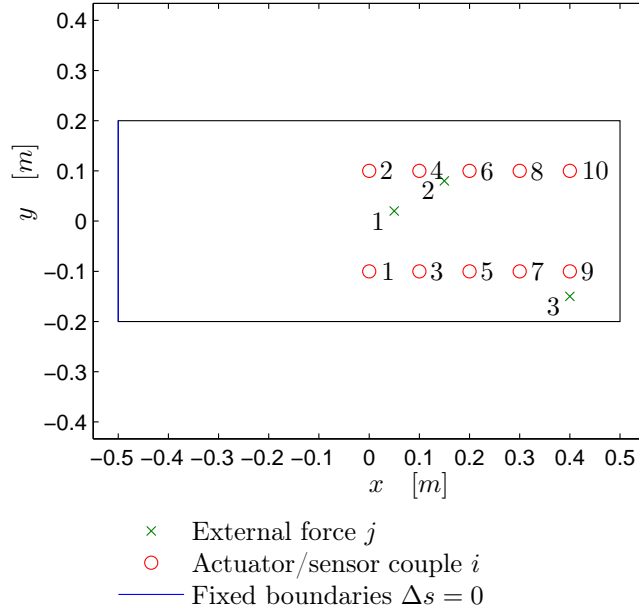
Figure C.3: Spring Board set with ten sensor/actuator couples and three external applied forces positions.

For simulation purposes the steel plate is considered to be structural steel. The following parameters are used for Femlab simulations and parameter identification:

Structural Steel			
Description	Parameter	Value	Unit
Youngs' modulus	E	$200 \cdot 10^9$	$\left[\frac{N}{m^2}\right]$
Poisson's ratio	ν	0.33	$[-]$
Density	ρ	7850	$\left[\frac{kg}{m^3}\right]$
Thermal Expansion Coeff.	α	$12.3 \cdot 10^{-6}$	$\left[\frac{1}{K}\right]$
Heat Capacity	C	475	$\left[\frac{J}{K}\right]$
Thermal Conductivity	k	44.5	$\left[\frac{W}{m \cdot K}\right]$
Conductivity	σ	$4.032 \cdot 10^6$	$\left[\frac{S}{m}\right]$
Relative Permittivity	ϵ_r	1	$[-]$
Relative Permeability	μ_r	1	$[-]$

Table C.1: Structural Steel parameters.

	Quadratic Steel Plate	Spring Board
Length $[m]$	0.80	1.0
Width $[m]$	0.80	0.40
Depth $[m]$	0.01	0.01

Table C.2: Dimensions of steel plates.

Parameter	Unit	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7
q	$\left[\frac{1}{m^2}\right]$	$3 \cdot 10^8$	$1 \cdot 10^8$	$2.5 \cdot 10^3$	$2 \cdot 10^8$	$2 \cdot 10^9$	$4 \cdot 10^7$	$7 \cdot 10^7$
$e_i(1),\ i = 1, ..., m$	$[Nm]$	$-1 \cdot 10^{-3}$		-0.1	$-1 \cdot 10^{-3}$		$-1 \cdot 10^{-4}$	
λ	$[-]$	0.90					0.99	0.90
Δs	$[m]$	$7 \cdot 10^{-7}$		$1 \cdot 10^{-4}$	$7 \cdot 10^{-7}$			
$\Delta \phi$	$\left[\frac{N}{m}\right]$	230	230	0.25	200	2000	0.3	0.01
$F_{ext,1}$	$[N]$	-500						
$F_{ext,2}$	$[N]$	-400						
$F_{ext,2}$	$[N]$	-500						

Table C.3: Control parameter sets for constant external forces F_{ext} and constant coupling coefficients $q_i = q = const.$

Parameter	Unit	Set 1	Set 2
q_1	$\left[\frac{1}{m^2}\right]$	$4.76 \cdot 10^7$	$6.67 \cdot 10^7$
q_2	$\left[\frac{1}{m^2}\right]$	$4.61 \cdot 10^7$	$6.45 \cdot 10^7$
q_3	$\left[\frac{1}{m^2}\right]$	$3.97 \cdot 10^7$	$5.56 \cdot 10^7$
q_4	$\left[\frac{1}{m^2}\right]$	$4.20 \cdot 10^7$	$5.88 \cdot 10^7$
$e_i(1),\ i = 1, ..., m$	$[Nm]$	$-1 \cdot 10^{-4}$	
λ	$[-]$	0.90	
Δs	$[m]$	$7 \cdot 10^{-7}$	
$\Delta \phi$	$\left[\frac{N}{m}\right]$	0.01	
$F_{ext,1}$	$[N]$	-500	
$F_{ext,2}$	$[N]$	-400	
$F_{ext,2}$	$[N]$	-500	

Table C.4: Control parameter sets for constant external forces F_{ext} and individual coupling coefficients q_i.

Parameter	Unit	Set 1	Set 2
ν	$\left[\frac{N^2}{m^2}\right]$	$5 \cdot 10^{11}$	$5 \cdot 10^{12}$
$e_{q,i}(1),\ i = 1, ..., m$	$\left[N^2\right]$	$4 \cdot 10^4$	
$e_i(1),\ i = 1, ..., m$	$[Nm]$	$-1 \cdot 10^{-4}$	
λ	$[-]$	0.99	
λ_q	$[-]$	0.99	
Δs	$[m]$	$7 \cdot 10^{-7}$	
$\Delta \phi$	$\left[\frac{N}{m}\right]$	0.1	
$F_{ext,1}$	$[N]$	-500	
$F_{ext,2}$	$[N]$	-400	
$F_{ext,2}$	$[N]$	-500	

Table C.5: Control parameter sets for constant external forces F_{ext} and adaptive coupling coefficients $q_i(l)$.

Parameter	Unit	Set 1
q	$\left[\frac{1}{m^2}\right]$	$7 \cdot 10^7$
$e_i(1), \; i = 1, ..., m$	$[Nm]$	$-1 \cdot 10^{-4}$
λ	$[-]$	0.98
Δs	$[m]$	$7 \cdot 10^{-7}$
$\overline{\Delta\phi}$	$\left[\frac{N}{m}\right]$	10
w	$[-]$	1000
uniformly distributed		
$F_{ext,1}(l)$	$[N]$	-400 to -600
$F_{ext,2}(l)$	$[N]$	-400 to -600
$F_{ext,2}(l)$	$[N]$	-400 to -600

Table C.6: Control parameter sets for variable external forces $F_{ext}(l)$ and constant coupling coefficients $q_i = q = const$.

Parameter	Unit	Set 1
$E\{\bar{F}_{act}\bar{s}^T\}$	$[Nm]$	$-1 \cdot 10^{-4} \cdot \begin{pmatrix} 1.0 & 0.5 & 0.5 & 0.3 \\ 0.5 & 1.0 & 0.3 & 0.5 \\ 0.5 & 0.3 & 1.0 & 0.5 \\ 0.3 & 0.5 & 0.5 & 1.0 \end{pmatrix}$
λ	$[-]$	0.99
Δs	$[m]$	$7 \cdot 10^{-7}$
$\overline{\Delta\phi}$	$\left[\frac{N}{m}\right]$	1
w	$[-]$	1000
uniformly distributed between		
$F_{ext,1}(l)$	$[N]$	-1000 to -800 and 800 to 1000
$F_{ext,2}(l)$	$[N]$	-1000 to -800 and 800 to 1000
$F_{ext,2}(l)$	$[N]$	-1000 to -800 and 800 to 1000

Table C.7: Control parameter sets for variable external forces $F_{ext}(l)$ and constant coupling coefficients $q_i = q = const$ in the case of a full mapping matrix ϕ^T.

Appendix D

Alternating Sequences with Exponential Growth

Generally, an alternating sequence (see Figure D.1) with exponential growth can be described as follows

$$s^i = \bar{s} + \alpha(-1)^i e^{\gamma i} \quad i \in \mathbf{N}_0 \tag{D.1}$$

Here, γ is the exponential growth factor ($\gamma > 0$) and α can be considered to be a scaling factor.

In (D.1) it can easily be seen that the inverted alternating sequence converges towards $\bar{s}$. Inversion means that the indices for i are non-positive and there follows

$$lim_{e\to-\infty} s^i = \bar{s} \tag{D.2}$$

As there are the unknown parameters α,γ and $\bar{s}$ in the sequence s^i, three equations are needed to solve them in dependency of three succeeding parts s^j, s^{j+1}, s^{j+2} of the sequence s^i. For the derivation, the first three indices $i = 0, 1, 2$ are used. The three first elements of the sequence are

$$s^0 = \bar{s} + \alpha \tag{D.3}$$

$$s^1 = \bar{s} - \alpha e^{\gamma} \tag{D.4}$$

$$s^2 = \bar{s} + \alpha e^{2\gamma} \tag{D.5}$$

In order to calculate the exponential growth rate γ it suffices if one calculates the absolute differences between the succeeding elements s^2 and s^1 as well all as the
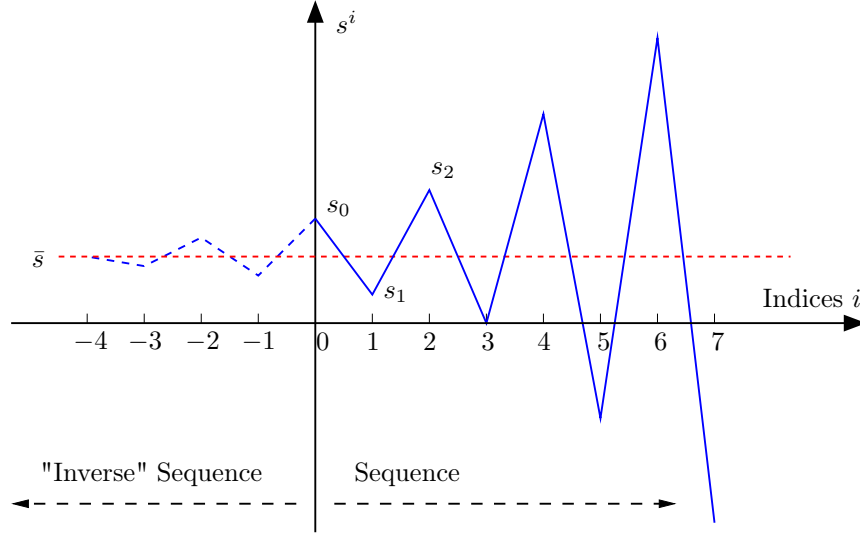
Figure D.1: Alternating Sequence with Exponential Growth.

elements s^1 and s^0. Out of (D.3) to (D.5) results

$$|s^2 - s^1| = \alpha(e^{2\gamma} + e^{\gamma}) \tag{D.6}$$

$$|s^1 - s^0| = \alpha(e^{\gamma} + 1) \tag{D.7}$$

If one divides the two equations and applies the logarithm to the quotient it leads to

$$\ln \frac{|s^2 - s^1|}{|s^1 - s^0|} = \ln \frac{e^{2\gamma} + e^{\gamma}}{e^{\gamma} + 1} = \gamma \tag{D.8}$$

In a second step the value α can be calculated from (D.3) and (D.4)

$$\alpha = \frac{s^0 - s^1}{e^{\gamma} + 1} \tag{D.9}$$

Finally the steady state $\bar{s}$ can be calculated from (D.3) with (D.8) and (D.9)

$$\bar{s} = s^0 - \alpha \overset{(D.9)}{=} s^0 + \frac{s^1 - s^0}{e^{\gamma} + 1} \overset{(D.8)}{=} s^0 + \frac{s^1 - s^0}{\frac{|s^2 - s^1|}{|s^1 - s^0|} + 1} \tag{D.10}$$

Appendix E

About Linear Spaces

This Appendix is taken from [12] and should be a very brief summary of basic linear algebra.

The set of all possible real-valued vectors u of dimension d constitutes the linear space $\mathbf{R}^d$. If $\mathbf{S} \in \mathbf{R}^d$ is a set of vectors, a subspace spanned by $\mathbf{S}$, or $\mathbf{L}(\mathbf{S})$, is the set of all linear combinations of the vectors in $\mathbf{S}$. A set of linearly independent vectors θ_i spanning a subspace is called a basis for that subspace.

The number of linearly independent vectors in the subspace basis determines the dimension of the subspace. The basis vectors θ_1 to θ_c of a c dimensional subspace can conveniently be represented in a matrix form:

$$\theta = (\theta_1 \,|\, \cdots \,|\, \theta_c) \tag{E.1}$$

Given a basis of a subspace, all points u_θ in that subspace have a unique representation. The basis vectors θ_i can be interpreted as coordinate axes in the subspace, and weights of the basis vectors, denoted now z_i, determine the corresponding coordinate values, or scores, of the point:

$$u_\theta = \sum_{i=1}^{c} z_i \theta_i \tag{E.2}$$

or in matrix form

$$u_\theta = \theta z \tag{E.3}$$

If $d > c$, an arbitrary data point u cannot necessarily be represented in the new basis. Using the least squares technique an approximation can be found that minimizes the

projection error

$$\hat{z} = \left(\theta^T \theta\right)^{-1} \theta^T u \tag{E.4}$$

If the basis vectors are orthonormal so that there holds $\theta^T \theta = I_c$, it is evident that (E.4) gives a simple solution:

$$\hat{z} = \theta^T u \tag{E.5}$$

Appendix F

Principal Components

This Appendix is taken from [12]. More about PCA can be read in [9].

PCA is a mathematical procedure for determining a subspace that optimally captures the variation in the higher-dimensional data. The easiest way to determine the principal components is by analysis of the data covariance matrix. For understanding principal components, knowledge of eigenvalues and eigenvectors is necessary. The eigenvector Θ_i and the corresponding eigenvalue λ_i of the data covariance matrix fulfill

$$E\{uu^T\}\Theta_i = \lambda_i\Theta_i \tag{F.1}$$

There are d distinct eigenvectors and corresponding eigenvalues. If one collects the eigenvectors of $E\{uu^T\}$ in the $d \times d$ dimensional matrix Θ, and the corresponding eigenvalues on the diagonal of $d \times d$ dimensional Λ, so that

$$\Theta = (\Theta_1 | \cdots | \Theta_c) \quad \text{and} \quad \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_c \end{pmatrix} \tag{F.2}$$

the covariance matrix can be decomposed as

$$E\{uu^T\} = \Theta\Lambda\Theta^{-1} = \Theta\Lambda\Theta^T \tag{F.3}$$

This comes from the fact, that eigenvectors of the symmetric matrix are orthogonal. Because of the construction of the covariance matrix, the eigenvalues are real and non-negative; one can order the eigenvectors in the order of decreasing significance, so that

$\lambda_1 \geq \cdots \geq \lambda_c \geq 0$. The eigenvalues λ_i reveal how much of the data variation takes place in the direction of the eigenvector Θ_i. This gives a practical way to compress data - select only the d most significant eigenvectors to constitute an orthonormal basis θ:

$$\theta = (\Theta_1 | \cdots | \Theta_d) \qquad \text{with } d < c \tag{F.4}$$

When the data u is projected onto this subspace, so that $z = \theta^T u$, one receives the latent variables or principal components that are mutually uncorrelated and capture the variation in u in an optimal way.

Appendix G

Table of Symbols

In this work appearing symbols in alphabetical order:

Chapter 2

Symbol	Description
A	Matrix of synaptic weights between neurons
a	Index
B	Matrix of weights between input and neurons
b	Index
C	Matrix of weights between neurons and output
c	Number of neurons
d	Number of inputs
$E\{\bar{x}u^T\}$	Covariance matrix of steady state neural activity $\bar{x}$ and input u
$E\{\bar{x}\bar{x}^T\}$	Covariance matrix of steady state neural activity $\bar{x}$
$E\{uu^T\}$	Covariance matrix of input u
h	Discrete time step
I_c	Identity matrix of dimension $c \times c$
$J(x)$	Cost criterion
l	Sample index
R	Vector of neural synaptic weights
$R(k)$	Vector of sampled neural synaptic weights
$R_{\bar{x}\bar{x}}$	Covariance matrix of steady state neural activity $\bar{x}$
$R_{\bar{x}u}$	Covariance matrix of steady state neural activity $\bar{x}$ and input u
$r_{ab}(k)$	Sampled synaptic weight between neuron a and input b
t	Time

Symbol	Description
u	Input data vector
$u(k)$	Sampled input data vector
$u_b(k)$	Input b
W	Weighting matrix for cost criteria
x	Vector of neural activities
$x(k)$	Sampled vector of neural activities
$\bar{x}(k)$	Vector of neural activity steady-states
$\bar{x}_a(k)$	Steady-state neural activity of neuron a
$x_{cont}(t, k)$	Continuous time vector of neural activity
$x_{cont}(\kappa h, k)$	Sampled time vector of neural activity
$x(\kappa, k)$	Discrete-time neural activity vector
$y(k)$	Data output vector
$\hat{y}(k)$	Vector of estimated data output
ϵ	Scaling factor
β	Factor for Steepest Descent Method
κ	Discrete-time factor
λ	Forgetting factor for covariance matrix updates
μ	Step size factor
ϕ	Mapping matrix
φ	Mapping matrix
ρ	Factor of synaptic weights between data input and neurons
τ	Time constant
θ	Matrix of PCA eigenvectors

Chapter 3, 4, 5, 6

Symbol	Description
A	Linear system matrix
B	Input matrix
$\tilde{B}$	Coupling coefficient matrix for evolutionary fitness
b	Coupling coefficient for evolutionary fitness
b_i	Individual coupling coefficient for sensor/actuator i
D	Orthogonal transformation matrix
$E\{\bar{x}\bar{u}^T\}$	Covariance matrix of balanced system $\bar{x}$ and balanced environment $\bar{u}$

Symbol	Description
$E\{\bar{x}\bar{x}^T\}$	Covariance matrix of balanced system $\bar{x}$
$E\{\bar{x}\Delta u^T\}$	Covariance matrix of balanced system $\bar{x}$ and environmental difference Δu
$E\{\Delta u\Delta u^T\}$	Covariance matrix of environmental differences Δu
$E\{uu^T\}$	Covariance matrix of undisturbed environment u
$E\{\bar{x}_i^2\}$	Variance of balanced system variable $\bar{x}_i$
$E\{\bar{x}u^T\}$	Covariance matrix of balanced system and undisturbed environment
$E\{\bar{x}'\bar{u}^T\}$	Covariance matrix of balanced system $\bar{x}'$ (variables free) and balanced environment $\bar{u}$
$E\{\bar{F}_{act}\bar{s}^T\}$	Covariance matrix of actuator forces $\bar{F}_{act}$ and sensor displacements $\bar{s}$
$E\{\bar{F}_{act,i}\bar{s}_i^T\}$	Covariance of sensor/actuator couple i
$E\{\bar{F}_{act,i}^2\}$	Variance of actuator force $\bar{F}_{act,i}$
$E\{\tilde{s}\tilde{s}^T\}$	Covariance matrix of undisturbed sensor displacements $\tilde{s}$
$E\{\tilde{s}_i^2\}$	Variance of undisturbed displacement $\tilde{s}_i$
$E\{\bar{s}\bar{s}^T\}$	Covariance matrix of balanced sensor displacements $\bar{s}$
$E\{\bar{s}_i^2\}$	Variance of balanced sensor displacement $\bar{s}_i$
$e_i(l)$	Sampled covariance of sensor/actuator couple i, $E\{\bar{F}_{act,i}\bar{s}_i\}(l)$
$e_{q,i}(l)$	Sampled variance of actuator force i, $E\{\bar{F}_{act,i}^2\}(l)$
F	Fixed mapping for evolutionary fitness
$F_{act,i}$	Actuator force i
$F_{act,i}(l)$	Actuator force i during applied external forces set l
$F_{act}^o(l)$	Vector of actuator forces during ongoing adaptation process
$F_{act,i}^o(l)$	Actuator force i during ongoing adaptation process
$\bar{F}_{act}$	Vector of balanced actuator forces
$\bar{F}_{act,i}$	Balanced actuator force of actuator i
$\bar{F}_{act,i}(l)$	Balanced actuator force i during adaptation process
$\bar{F}_{act}(l)$	Vector of balanced actuator forces during adaptation process
$F_{ext,j}(l)$	External force j during ongoing adaptation process
$F_{ext}(l)$	Vector of external forces during ongoing adaptation process
F_s	Applied spring force
F_{test}	Defined test force for identification purposes
H	Matrix of damping factors for external forces
I_m	Identity matrix dimension $m \times m$
I_n	Identity matrix dimension $n \times n$

Symbol	Description
i	Index
$J(x, u)$	Cost criterion
j	Index
$\mathcal{J}(x, u)$	Cost criterion
k	Spring constant
l	Set of external forces
l_b	Index for second order iteration stop
M	Matrix of inverse spring constants for actuators
m	Number of environmental variables
m_j	Inverse spring constant of actuator force j
N	Matrix of inverse spring constants for external forces
n	Number of system variables
n_k	Inverse spring constant of external force k
o	Iteration index for first order iteration
o_b	Index for first order iteration stop
Q	Coupling coefficient matrix for evolutionary fitness
q	Coupling coefficient for evolutionary fitness
q_i	Individual coupling coefficient for sensor/actuator i
$q_i(l)$	Coupling coefficient for sensor/actuator couple i during adaptation process
$\bar{s}$	Vector of balanced sensor displacements
$\bar{s}_i$	Balanced sensor displacement of sensor i
$\bar{s}_i(l)$	Balanced sensor displacement of sensor i during adaptation process
$\bar{s}(l)$	Vector of balanced sensor displacements during adaptation process
$\tilde{s}(l)$	Vector of initial displacements of sensors with applied set l of external forces
$\tilde{s}_i(l)$	Initial displacement of sensor i with applied set l of external forces
s_j^{ext}	Displacement of external forces position j
$s_{g,i}$	Sensor displacement i due to gravity
s_g	Vector of sensor displacements due to gravity
s_g^{ext}	Vector of displacements of external forces positions due to gravity
s_i	Sensor displacement for sensor i
$s_i(l)$	Sensor displacement for sensor i and applied external forces set l
$s^o(l)$	Vector of sensor displacements during ongoing adaptation

Symbol	Description
$s_i^o(l)$	Sensor displacement for sensor i during ongoing adaptation
s_*	Vector of transformed sensor displacements (remark: all mentioned sensor displacements exist also in transformed coordinates)
s_s	Displacement of spring
T	Time
u	Vector of environmental variables
Δu	Vector of difference between undisturbed and balanced environment
$\tilde{u}$	Vector of changed environment
$\bar{u}$	Vector of balanced environmental variables
W	Weighting matrix for cost criteria
$W_{ext}(s_s)$	External spring energy
$W_{int}(s_s, F_s)$	Internal Spring energy
w	Window size for mean value calculation of mapping matrix elements ϕ_{ii}^T
x	Vector of system variables
$\bar{x}$	Vector of system steady-states
x'	Vector of free system variables
$\bar{x}'$	Vector of balanced free system variables
$Z(l)$	Sequence matrix for stability check during adaptation
α	Scaling factor
ϵ_i	Initial value for $e_i(l)$
η_{ik}	damping from external force k to sensor i
γ	Scaling factor
$\Lambda(l)$	Eigenvalue matrix of transformed sequence matrix $Z(l)$
λ	Forgetting factor for covariance matrix of evolutionary fitness
λ_i	Eigenvalue i of undisturbed environmental covariance matrix
$\lambda_i(l)$	Eigenvalue i of sequence matrix $Z(l)$
$\bar{\lambda}_i$	Eigenvalue of balanced environmental covariance matrix
λ_q	Forgetting factor for adaptive coupling coefficients
ν	Controlling parameter in case of adaptive learning coupling coefficients q_i
ν_i	Initial value for $e_{q,i}(l)$
Φ	Mapping matrix
ϕ^T	Mapping matrix

Symbol	Description
$\phi^T(l)$	Mapping matrix during ongoing adaptation
$\Delta\phi$	Breaking criterion for second order iteration
$\overline{\Delta\phi}$	Breaking criterion for second order iteration in case of varying external forces
ϕ_{ii}^T	Diagonal mapping matrix entry
$\phi_{ii}^T(l)$	Diagonal mapping matrix entry for ongoing adaptation
Δs	Breaking criterion for first order iteration
θ	Matrix of eigenvectors for PCA
$\bar{\theta}$	Mapping matrix
θ_i	Eigenvector i of matrix θ
φ	Mapping matrix
Ξ	Matrix of damping factors of actuators
ξ_{ij}	damping from actuator force j to sensor i

List of Figures

List of Tables

Bibliography

[1] Comsol AB. *Femlab 3.1 Modeling Guide.* Femlab AB, Los Angeles, California, 2004.

[2] Comsol AB. *Femlab 3.1 User's Guide.* Femlab AB, Los Angeles, California, 2004.

[3] Boßle, Höllig, Streit. *Konvergenz einer Vektor-Folge.* 2006. `http://mo.mathematik.uni-stuttgart.de/inhalt/aussage/aussage522/`.

[4] J. Clark. *Le Chatelier's Principle.* 2002. `http://www.chemguide.co.uk/physical/equilibria/lechatelier.html`.

[5] Comsol, Inc. *COMSOL Multiphysics^{TM}.* 2006. `http://www.comsol.com/products/multiphysics/`.

[6] S. Haykin. *Neural Networks - A Comprehensive Foundation.* Prentice-Hall, Upper Saddle River, NJ, 1999.

[7] D.O. Hebb. *The Organization and Behaviour: A Neuropsychological Theory.* John Wiley and Sons, New York, NY, 1949.

[8] H. Hyötyniemi. *Presentation: Towards "Cybernetic Control".* 07.10.2006.

[9] H. Hyötyniemi. *Multivariate Regression - Techniques and Tools.* 2001.

[10] H. Hyötyniemi. Cybernetics: Towards a Unified Theory. *Finnish Artificial Intelligence Conference SteP'04*, 2004.

[11] H. Hyötyniemi. Emergent Coordination in Distributed Sensor Networks. *Finnish Artificial Intelligence Conference SteP'04*, 2004.

[12] H. Hyötyniemi. Hebbian Neuron Grids: System Theoretic Approach. Technical report, Helsinki University of Technology, Control Engineering Laboratory, Report 144, 2004.

[13] H. Hyötyniemi. Information and Entropy in Cybernetic Systems. *Finnish Artificial Intelligence Conference SteP'04*, 2004.

[14] H. Hyötyniemi. *Cybernetics, it is here, it is hot.* 2006. `http://control.hut.fi/Research/cybernetics/vision.html`.

[15] H. Hyötyniemi. Neocybernetics in Biological Systems. Technical report, Helsinki University of Technology, Control Engineering Laboratory, 2006.

[16] I.F. Akyldiz, I. H. Kasimoglu. Wireless Sensor and Actor Networks: Research Challenges. Technical report, Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, April 2004.

[17] K. Meyberg, P. Vachenauer. *Höhere Mathematik 1.* Springer Verlag, Berlin, Heidelberg, New York, 1999.

[18] K.I. Diamantaras, S.Y. Kung. *Principal Components Neural Networks: Theory and Applications.* Wiley, New York, NY, 1996.

[19] E. Oja. Principal Components, Minor Components and Linear Neural Networks. *Neural Networks*, 5:927–935, 1992.

[20] R.D Cook, D.S. Malkus, M.E. Plesha and R.J. Witt. *Concepts and Applications of Finite Element Analysis, 4th edition.* Wiley and Sons, New York, NY, 2001.

[21] S. Russell, P. Norvig. *Artificial Intelligence - A Modern Approach.* Prentice Hall, New Jersey, 1995.

[22] The Mathworks Inc. *Matlab Function Reference - cond.* 1994-2006. `http://www.mathworks.com/access/helpdesk/help/techdoc/ref/cond.html`.

[23] U. Riede, M. Werner, H. Schäfer. *Allgemeine und Spezielle Pathologie.* Thieme Verlag, 5. Auflage, 2004.

[24] Eric W. Weisstein. *Method of Steepest Descent. From MathWorld–A Wolfram Web Resource.* 2006. `http://mathworld.wolfram.com/MethodofSteepestDescent.html`.

[25] N. Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine.* Wiley, New York, NY, 1948.

[26] Wikipedia - Die freie Enzyklopädie. *Kovarianzmatrix.* 2006. `http://de.wikipedia.org/wiki/Kovarianzmatrix`.

[27] Wikipedia - Die freie Enzyklopädie. *Minimum Total Potential Energy Principle.* 2006. `http://en.wikipedia.org/wiki/Minimum_total_potential_energy_principle`.

[28] Wikipedia - The free Encyclopedia. *Diagonalmatrix.* 2006. `http://de.wikipedia.org/wiki/Diagonalmatrix`.

[29] Wikipedia - The free Encyclopedia. *Trace (linear algebra).* 2006. `http://en.wikipedia.org/wiki/Matrix_trace`.