Lehrstuhl für Regelungstechnik

Technical Reports on Automatic Control
Vol. TRAC-1, Oct. 2006
Available online at: www.rt.mw.tum.de

TECHNISCHE UNIVERSITÄT MÜNCHEN

A Compromise Between Simplicity & Accuracy of Nonlinear Reduced Order Systems

A. Yousefi [*,†]

Abstract

This report considers the problem of complexity in reduced nonlinear models derived by the so-called system matrices optimization method. The solution suggested here is adding a class of secondary conditions that impose a simpler structure on the reduced model. Distinct choices of these conditions have different impacts on the accuracy of the reduced model, therefore a pioneered search based on genetic algorithms is proposed that finds the optimal choice of conditions. By means of two examples, it is illustrated that this method can effectively compromise between simplicity and accuracy of the reduced model. In order to improve the numerical efficiency of the solution and to speed up the pace of convergence some enhancements are presented.
keywords: Order Reduction, Model Simplification, Nonlinear Systems, Genetic Algorithm, Fitness Function

[*]This work was partially supported by the DFG (German Research Council)
[†]Corresponding author: Tel: +49 89 289 15670, Fax: +49 89 25915659, email: Yousefi@tum.de

Contents

List of Figures

1 Introduction

Typical nonlinear dynamical systems are modeled by means of a set of first order coupled differential equations or a set of partial differential equations. The models which are described with partial differential equations can be also solved numerically by first spatially discretizing them by means of finite element, boundary element and similar methods which lead to a set of ordinary differential equations. In the first case the order of the system (number of state variables) depends on the quality of modeling and complexity of the system, but in the second case it depends on the quality of discretization. Recent advances in hardware and software technology provide this ability to solve very large systems of ordinary differential equations. Nevertheless, typically these calculations need parallel processing which increases the cost of simulation drastically, and as a result, limits the simulation applicability considerably. Therefore the complexity of simulation, analysis and controller design of a system depends directly on the complexity of the corresponding system model. On the one hand high order complex models are more accurate and reliable and on the other hand their cost of simulation and analysis is much higher. In order to face this dilemma, two methodologies are imaginable, *order reduction* and *structure simplification*:

1. The idea behind order reduction is to approximate a dynamic system with a model with less number of state variables. Order reduction methods generally calculate models of lower order but of high inner complexity. In other words they generally result in reduced systems with high number of internal interconnections, i.e. the inner model structures are complex.

2. The idea behind structure simplification is simplifying the relations and coupling among the state variables. This idea is published in [3] and it was developed later in [4, 5, 29] and [28].

The problem that we address through this report is a combination of these two ideas in one algorithm. Starting from an enhanced version of the system matrices optimization method presented in this report, secondary conditions are formulated to calculate reduced systems with simpler structures. One of the methods to find optimal secondary conditions is exploiting genetic algorithm in order to perform a global search within the search space. In this report an effective fitness function is presented, which simplifies the search procedure and enormously reduces the computation effort. Also a method for omitting improper candidates is suggested that accelerates the whole search process by shrinking the search space.

In section 3 some well-known methods of nonlinear order reduction are reviewed. Section 4 gives a summary of Order reduction using system matrices optimization method [15, 16] and it proposes some new enhancements of this method. Section 6 introduces appropriate secondary conditions to simplify the reconstruction matrix and the reduced order system matrices. The effectiveness of the results are demonstrated via two examples in section 7 and section 8 contains concluding remarks.

2 Preliminaries

The notation used in this report is moderately standard. Matrices are represented as bold upper case $(\mathbf{A})$, column vectors as bold lower case $(\mathbf{x})$ and real or complex scalars as italic lower case (t). All vectors are column vectors unless explicitly written as transposed. $\mathbf{A}^{-1}$ and $\mathbf{A}^{T}$ denote respectively the inverse and transpose of A. The notation $\|\mathbf{A}\|$ means the square of the Euclidian (Frobenius) norm of matrix $\mathbf{A}$ and is defined as $\sqrt{\sum diag(\mathbf{A^T A})}$. We consider the nonlinear dynamical systems modeled by means of a set of first-order coupled differential equations (which describe the behavior of the state variables) together with a set of algebraic equations (which describe the dependency of outputs on internal state variables) as follows:

$$\mathcal{S}_1 : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x(t)}, \mathbf{u(t)}) \\ \mathbf{y}(t) = \mathbf{h}(\mathbf{x(t)}, \mathbf{u(t)}) \end{cases} \tag{1}$$

or with more details:

$$\mathcal{S}_2 : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{F}\mathbf{g}(\mathbf{x},\mathbf{u}) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \tag{2}$$

In this representation the vector $\mathbf{g}(\mathbf{x},\mathbf{u})$ exclusively includes the nonlinear summands of the elements $\mathbf{f}(\mathbf{x},\mathbf{u})$ in the system representation (1). This special representation (2) is based on the assumptions that some of the state equations of many technical systems contain linear terms or may even be completely linear, resulting zero-rows in matrix $\mathbf{F}$. The problem that we address here is to simplify or approximate the original nonlinear system with another one with smaller number of states and simple structure.

3 Review of known techniques

In this section we will review briefly some well known methods of nonlinear order reduction and we discuss why none of the existing methods fulfills our expectation of computation efficiency and simplicity of the reduced order system.

Singular Perturbation ([13, 23]): This method is based on the assumption that the system equations can be separated into fast and slow modes as follows:

$$\mathcal{S}_{slow-fast} : \begin{cases} \dot{\mathbf{x}}(t) & = \mathbf{f}(\mathbf{x},\mathbf{z},\mathbf{u},t,\mu) \\ \mu\dot{\mathbf{z}} & = \mathbf{g}(\mathbf{x},\mathbf{z},\mathbf{u},t,\mu) \end{cases}$$

where $\mu > 0$ is a scalar and $\mathbf{x},\mathbf{z}$ and $\mathbf{u}$ are vectors. This method decreases the order of the model, first by ignoring the fast modes of the system and keeping the slower modes. Assuming μ close to zero and substituting the steady state value of $\mathbf{z}$ (denoted by $\tilde{\mathbf{z}}$) in the original system, the following results will be achieved.

$$\tilde{\mathbf{z}} = \phi(\tilde{\mathbf{x}},\mathbf{u},t),$$
$$\mathcal{S}^{SingPer.}_{reduced} : \dot{\tilde{\mathbf{x}}}(t) = \mathbf{f}(\tilde{\mathbf{x}},\phi(\tilde{\mathbf{x}},\mathbf{u},t),\mathbf{u},t)$$

It should also be noted that it is not always easy to find the function ϕ and the combinations $\mathbf{f}(\tilde{\mathbf{x}},\phi,\mathbf{u},t)$ has normally a very complicated structure.

Proper Orthogonal Decomposition ([26]): Consider the nonlinear system represented by (1). For a fixed input u, the state trajectory at certain instances of time t_k is measured as follows:

$$\chi = [\mathbf{x}(t_1)\mathbf{x}(t_2)\cdots\mathbf{x}(t_N)] \tag{3}$$

This is called a matrix of *snapshots* of the states. If the singular values of (3) decrease rapidly, it can be proved that the optimal approximation of χ in the sense of ℓ_2 norm, evolves on a low-dimensional space which is spanned by the first k leading columns of $\mathbf{U}$ as shown below:

$$\chi = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}} \approx \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T, \quad k \ll n$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and $\mathbf{\Sigma} = diag(\sigma_1,...,\sigma_n)$ is diagonal [7, 9, 10]. Then the reduced order system will be obtained as follows:

$$\mathcal{S}^{POD}_{reduced} : \begin{cases} \dot{\tilde{\mathbf{x}}}(t) = \mathbf{U}_k^T\mathbf{f}(\mathbf{U}_k\tilde{\mathbf{x}}(t),\mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{h}(\mathbf{U}_k\tilde{\mathbf{x}}(t),\mathbf{u}(t)) \end{cases}$$

In the POD procedure the effect of states and inputs on outputs is not taken into account and this confines its application, for more details and examples refer to [20, 21].

Nonlinear Balancing ([25]): This method is an extension of balancing for linear systems in the sense that it is based on extended definition of balancing and Hankel singular functions which was first introduced in by Scherpen [24, 25]. This method is applicable to nonlinear systems with the following representation:

$$\mathcal{S}_2 : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) \end{cases}$$

The main objective involved in balancing theory are the controllability and observability energy functions, which are defined as follows:

$$L_c(\mathbf{x}_0) = \min_{x(-\infty)=0, x(0)=x_0} \frac{1}{2} \int_{-\infty}^{0} \|\mathbf{u}(t)\|^2 dt \tag{4}$$

$$L_o(\mathbf{x}_0) = \frac{1}{2} \int_{-\infty}^{0} \|\mathbf{y}(t)\|^2 dt,\ \mathbf{x}(0) = \mathbf{x}_0, \mathbf{u}(t)\Big|_{0<t<\infty} = 0 \tag{5}$$

This method similar to the concept of balancing for linear systems finds a coordinate transformation in form of $\mathbf{x} = \psi(\mathbf{z})$ that balances the system due to extended definitions of balancing for nonlinear systems, which results in

$$\mathcal{S}_{balanced} : \begin{cases} \dot{\mathbf{z}}(t) = \tilde{\mathbf{f}}(\mathbf{z}) + \tilde{\mathbf{g}}(\mathbf{z})\mathbf{u}(t) \\ \mathbf{y}(t) = \tilde{\mathbf{h}}(\mathbf{z}(t)) \end{cases} .$$

The reduced model is obtained by trimming the states corresponding to small Hankel singular values [25]. Computation of L_c via (4) requires solution of an optimal control problem, which presents computational difficulties and restricts its application to very low order nonlinear systems.

4 System matrices optimization method

Since the *system matrices optimization* method is the core of our discussions, it is described in more details in this section. The system matrices optimization method [15, 16, 17] can be exploited for nonlinear systems with the representation in (2). Starting from (2) the task of order reduction in system matrices optimization method is to find a system of lower order $\tilde{n}$ which delivers an approximation ($\hat{\mathbf{x}}$) of the dominant state variables ($\mathbf{x_{do}}$) as follows:

$$\mathcal{S}_{reduced} : \begin{cases} \dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}\mathbf{u}(t) + \tilde{\mathbf{F}}\mathbf{g}(\mathbf{W}\tilde{\mathbf{x}}, \mathbf{u}) \\ \mathbf{y}(t) = \tilde{\mathbf{C}}\tilde{\mathbf{x}}(t) \end{cases} \tag{6}$$

These dominant state variables are chosen by the designer and are combined in the vector $\mathbf{x_{do}}$ which is related to the original vector $\mathbf{x}$ by

$$\mathbf{x_{do}} = \mathbf{R}\mathbf{x}. \tag{7}$$

Based on the given system (2) and the definition of dominant state variables, the system matrices optimization method calculates the matrices that describe the reduced order system ($\mathbf{E} = [\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{F}}]$ and $\mathbf{W}$) such that they optimally fit the snapshots of the dominant state variables of the original system in the sense of Euclidean norm [16, 17]. Assume that matrices $\chi,\ \chi_{\mathbf{do}},\ \dot{\chi}_{\mathbf{do}}$, $\mathbf{\Psi}$ and $\mathbf{\Gamma}$ are the snapshots of the original system for typical input signals that respectively show the numerical values of state variables, dominant state variables, derivative of dominant state variables, inputs and nonlinear part as shown below:

$$\begin{aligned} \chi_{\mathbf{do}} &= [\mathbf{x_{do}}(t_1)\mathbf{x_{do}}(t_2)\cdots\mathbf{x_{do}}(t_N)],\ \mathbf{\Psi} = [\mathbf{u}(t_1)\mathbf{u}(t_2)\cdots\mathbf{u}(t_N)] \\ \dot{\chi}_{\mathbf{do}} &= [\dot{\mathbf{x}}_{\mathbf{do}}(t_1)\dot{\mathbf{x}}_{\mathbf{do}}(t_2)\cdots\dot{\mathbf{x}}_{\mathbf{do}}(t_N)],\ \mathbf{\Gamma} = [\mathbf{g}(t_1)\mathbf{g}(t_2)\cdots\mathbf{g}(t_N)] \\ \chi &= [\mathbf{x}(t_1)\mathbf{x}(t_2)\cdots\mathbf{x}(t_N)]. \end{aligned}$$

To find $\mathbf{E}$ and $\mathbf{W}$, the following optimization problems are solved:

$$\min_{\mathbf{E}} \|\dot{\chi}_{\mathbf{do}} - \underbrace{[\tilde{\mathbf{A}} \quad \tilde{\mathbf{B}} \quad \tilde{\mathbf{F}}]}_{\mathbf{E}} \underbrace{\begin{bmatrix} \chi_{\mathbf{do}} \\ \mathbf{\Psi} \\ \mathbf{\Gamma} \end{bmatrix}}_{\mathbf{M}} \| \Rightarrow \mathbf{E_{nc}} \tag{8}$$

$$\min_{\mathbf{W}} \|\chi - \mathbf{W}\chi_{\mathbf{do}}\| \Rightarrow \mathbf{W_{nc}} \tag{9}$$

The optimal solution can be evaluated using (10).

$$\mathbf{E_{opt}} = \dot{\chi}\mathbf{M^T}(\mathbf{MM^T})^{-1},\ \mathbf{W_{opt}} = \chi\chi_{\mathbf{do}}^{\mathbf{T}}(\chi_{\mathbf{do}}\chi_{\mathbf{do}}^{\mathbf{T}})^{-1} \tag{10}$$

Exploiting the result of (10) the reduced system is completely determined and in addition to dominant state variables the non-dominant state variables are approximated using $\mathbf{W}$. Accordingly, the vector $\mathbf{g}$ of the nonlinearities is taken over from the original system (2) into the reduced order system and no additional nonlinearities are introduced. For more details on the method, for instance how to choose $\mathbf{R}$ in (7) and the input signals refer to [15, 16, 17].

4.1 New development: row by row reduction

In following theorem we show that the original optimization problem can be split into smaller row by row optimization problems (see application in section 6.1).

Theorem 1. In system matrices optimization method solving the following optimization problem

$$\min_{\mathbf{E}} \|\dot{\chi}_{\mathbf{do}} - \underbrace{[\tilde{\mathbf{A}} \quad \tilde{\mathbf{B}} \quad \tilde{\mathbf{F}}]}_{\mathbf{E}} \underbrace{\begin{bmatrix} \chi_{\mathbf{do}} \\ \mathbf{\Psi} \\ \mathbf{\Gamma} \end{bmatrix}}_{\mathbf{M}} \| \tag{11}$$

is equivalent to solving $\tilde{n}$ independent optimization problems as follow:

$$\min_{\mathbf{e_i^T}} \| \dot{\mathbf{x}}_{\mathbf{do_i}}^{\mathbf{T}} - \underbrace{[\tilde{\mathbf{A}}_{\mathbf{i}} \quad \tilde{\mathbf{B}}_{\mathbf{i}} \quad \tilde{\mathbf{F}}_{\mathbf{i}}]}_{\mathbf{e_i^T}} \begin{bmatrix} \chi_{\mathbf{do}} \\ \mathbf{\Psi} \\ \mathbf{\Gamma} \end{bmatrix} \|,\ i = 1, 2, \ldots, \tilde{n}$$

where $\dot{\mathbf{x}}_{\mathbf{do_i}}^{\mathbf{T}}$ is the snapshots of the i_{th} states derivative and $\tilde{\mathbf{A}}_{\mathbf{i}},\ \tilde{\mathbf{B}}_{\mathbf{i}}\ and\ \tilde{\mathbf{F}}_{\mathbf{i}}$ are the i_{th} row of the reduced order system matrices $\tilde{\mathbf{A}},\ \tilde{\mathbf{B}}\ and\ \tilde{\mathbf{F}}$ respectively, that should be calculated.

Proof: See the Appendix.

Using Theorem 1 the optimization problems in (8,9) are equivalent to $n + \tilde{n}$ independent optimization problems as follows:

$$\text{for } i = 1, 2, \ldots, n \quad \min_{\mathbf{w_i}} \| \mathbf{x_i^T} - \mathbf{w_i^T}\chi_{\mathbf{do}}\|, \Rightarrow \mathbf{W_{nc}} \tag{12}$$

$$\text{for } i = 1, 2, \ldots, \tilde{n}$$

$$\min_{\mathbf{e_i^T}} \| \dot{\mathbf{x}}_{\mathbf{do_i}}^{\mathbf{T}} - \underbrace{[\tilde{A}_i \quad \tilde{B}_i \quad \tilde{F}_i]}_{\mathbf{e_i^T}} \underbrace{\begin{bmatrix} \chi_{do} \\ \Psi \\ \Gamma \end{bmatrix}}_{\mathbf{M}} \| \Rightarrow \mathbf{E_{nc}}, \tag{13}$$

where $\dot{\mathbf{x}}^{\mathbf{T}}_{\mathbf{do_i}}$ is the snapshots of derivative of the i_{th} state variable, $\tilde{\mathbf{A}}_{\mathbf{i}}, \tilde{\mathbf{B}}_{\mathbf{i}}$ and $\tilde{\mathbf{F}}_{\mathbf{i}}$ are the i_{th} row of the reduced order system matrices $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$ and $\tilde{\mathbf{F}}$ respectively, $\mathbf{x}^{\mathbf{T}}_{\mathbf{i}}$ is the snapshots of i_{th} state variable and $\mathbf{w}^{\mathbf{T}}_{\mathbf{i}}$ is the i_{th} row of matrix $\mathbf{W}$. The index *nc* means no constraints is applied to the optimization problem. In fact $\mathbf{E_{nc}}$ and $\mathbf{W_{nc}}$ are the optimal answers in the sense of (8) and (9). The optimal solution can be evaluated explicitly using (14).

$$\mathbf{e}^{\mathbf{T}}_{\mathbf{opt_i}} = \dot{\mathbf{x}}^{\mathbf{T}}_{\mathbf{do_i}}\mathbf{M}^{\mathbf{T}}(\mathbf{M}\mathbf{M}^{\mathbf{T}})^{-\mathbf{1}}, \mathbf{w}^{\mathbf{T}}_{\mathbf{opt_i}} = \mathbf{x}^{\mathbf{T}}_{\mathbf{i}}\chi^{\mathbf{T}}_{\mathbf{do}}(\chi_{\mathbf{do}}\chi^{\mathbf{T}}_{\mathbf{do}})^{-\mathbf{1}} \tag{14}$$

Exploiting the result of (14) the reduced system is completely determined and the reduced system is set up as in (6).

5 Dominant State Finder Algorithm (DSFA)

The choice of matrix $\mathbf{R}$ in (7) can be a difficult task. Therefore, this section proposes a *Dominant State Finder Algorithm* (*DSFA*) that enables *automatic selection of dominant states* $\mathbf{x_{do}}$ in nonlinear systems, without losing their physical meanings. Our algorithm is based on the snapshot trajectories of the original system as shown below:

$$\begin{aligned}
&\chi{=}[\mathbf{x}(t_1)\mathbf{x}(t_2)\cdots\mathbf{x}(t_N)], \Psi{=}[\mathbf{u}(t_1)\mathbf{u}(t_2)\cdots\mathbf{u}(t_N)]\\
&\dot{\chi}{=}[\dot{\mathbf{x}}(t_1)\dot{\mathbf{x}}(t_2)\cdots\dot{\mathbf{x}}(t_N)], \Gamma{=}[\mathbf{g}(t_1)\mathbf{g}(t_2)\cdots\mathbf{g}(t_N)]\\
&\Xi{=}[\mathbf{y}(t_1), \mathbf{y}(t_2), \cdots, \mathbf{y}(t_N)]
\end{aligned} \tag{15}$$

We define a set of states $\mathbf{x_{do}}$ as dominant state variables if there exist matrices $\tilde{\mathbf{C}}, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{F}}$ and $\mathbf{W}$ such that the following errors are small

$$\begin{aligned}
e_1 = \|\mathbf{y}(t) - \tilde{\mathbf{C}}\mathbf{x_{do}}(t)\|, \quad e_2 = \|\mathbf{g}(\mathbf{x},\mathbf{u}) - \mathbf{g}(\mathbf{W}\mathbf{x_{do}},\mathbf{u})\|,\\
e_3 = \|\dot{\mathbf{x}}_{\mathbf{do}} - (\tilde{\mathbf{A}}\mathbf{x_{do}} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{F}}\mathbf{g}(\mathbf{W}\mathbf{x_{do}},\mathbf{u}))\|.
\end{aligned} \tag{16}$$

The idea here is to approximate the error in (16) by exploiting the snapshot matrices. The points of interest are marked by dashed boxes in the block diagram in Fig. 1, where $(a.), (b.)$ and $(c.)$ show output, nonlinear part and derivative of states respectively.

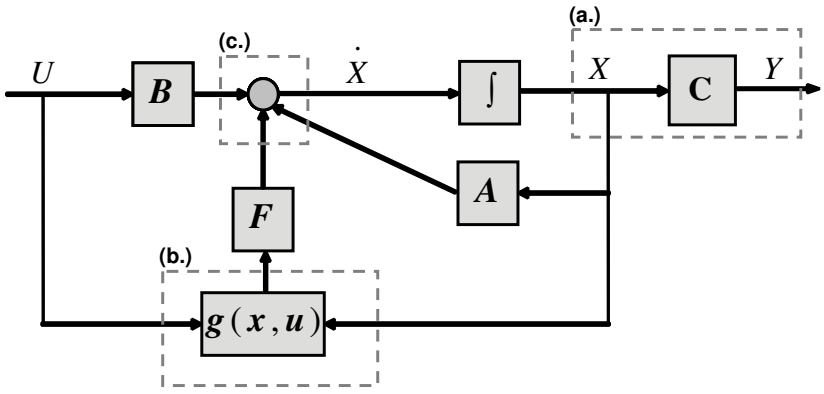

Figure 1: **a.:** System output Y; **b.:** The nonlinear function $g(X_g, U)$; **c.:** $\dot{X}$ is constructed by $X, g(X, U)$ and U

Considering the snapshot matrices

$$\chi_{\mathbf{do}}{=}[\mathbf{x_{do}}(t_1)\mathbf{x_{do}}(t_2)\cdots\mathbf{x_{do}}(t_N)], \quad \dot{\chi}_{\mathbf{do}}{=}[\dot{\mathbf{x}}_{\mathbf{do}}(t_1)\dot{\mathbf{x}}_{\mathbf{do}}(t_2)\cdots\dot{\mathbf{x}}_{\mathbf{do}}(t_N)]$$

$\mathbf{x_{do}}$ represents the dominant states if the following three conditions are satisfied

1. subspace Ξ is close to subspace $\chi_{\mathbf{do}}$ ($\mathbf{y}$ can be approximated by $\mathbf{x_{do}}$)
2. subspace χ is close to subspace $\chi_{\mathbf{do}}$ ($\mathbf{g}(\mathbf{x},\mathbf{u})$ can be approximated by $\mathbf{g}(\mathbf{W}\mathbf{x_{do}},\mathbf{u})$)
3. subspace $\dot{\chi}_{\mathbf{do}}$ is close to the union of subspaces χ_{do}, Γ and Ψ ($\dot{\mathbf{x}}_{\mathbf{do}}$ can be approximated by $\mathbf{x_{do}}, \mathbf{g}(\mathbf{x_{do}},\mathbf{u})$ and $\mathbf{u}$)

The closeness of two subspaces can be interpreted mathematically by defining the angle between them as presented in [12]. If the angle is small, the two subspaces are close to each other and nearly linearly dependent. In our case, suppose A describes the complete snapshots of a physical model, and B describes a subset of it, then the angle between these two spaces gives a measure of the amount of information afforded by the full snapshot not associated with the reduced snapshot. Suppose two subspaces are specified by the columns of orthonormal matrices A and B, the angle between these two subspaces, according to [2], is calculated as follows:

$$\begin{aligned}&\texttt{for } k = 1 : size(A)\\ &\qquad B = B - A_k A_k^T B\\ &\texttt{end}\end{aligned}$$

where A_k denotes the k_{th} column of A. If $\|B\| \leq 1$

$$\angle(A,B) = sin^{-1}(\|B\|)$$

otherwise $\angle(A,B) = \frac{\pi}{2}$.The angle θ is contained by the interval $[0, \frac{\pi}{2}]$. Where the angle $\theta = 0$ of two bases A and B means, that A is subspace to B and for $\theta = \frac{\pi}{2}$ no correlation between the two bases is given.

Based on above definitions of dominant states and angle between subspaces, we propose the following algorithm for finding the dominant states:

1. Choose an upper bound for the maximum allowed angle (θ_{max})
2. Sort the states according to the angles between their snapshots and output snapshot matrix defined as follows:

$$\angle(\chi_i, \Xi), \quad i = 1, \cdots, n \text{ where } \chi_i = [\mathbf{x_i}(t_1), \cdots, \mathbf{x_i}(t_N)]$$

3. Find the minimum number of states (m_1) such that

$$\angle(\chi_{1:m_1}, \Xi) < \theta_{max} \text{ where } \chi_{1:m_1} \text{ is the union of } \chi_i\text{s, } i = 1, \cdots, m_1 \tag{17}$$

4. Sort the remaining states ($(m_1+1) : n$) according to the angles between their snapshots and original states snapshot matrix defined as follows:

$$\angle(\chi_i, \chi), \quad i = (m_1+1), \cdots, n$$

5. Add the minimum number of states (m_2) to the last selected set such that

$$\angle(\chi_{1:(m_1+m_2)}, \chi) < \theta_{max} \tag{18}$$

6. Sort the remaining states ($(m_1+m_2+1) : n$) according to the angles between their snapshots and the union of snapshot matrices χ, Γ and Ψ defined as follows:

$$\angle(\chi_i, (\chi \cup \Gamma \cup \Psi)), \quad i = (m_1+m_2+1), \cdots, n$$

7. Add the minimum number of states (m_3) to the last selected set such that

$$\angle(\chi_{1:(m_1+m_2+m_3)}, (\chi \cup \Gamma \cup \Psi)) < \theta_{max} \tag{19}$$

8. The selected $m_1 + m_2 + m_3$ states indicate the dominant state variables.

The inequalities (17), (18) and (19) are connected to the errors e_1, e_2 and e_3 in (16) respectively. By proper choice of θ_{max}, the errors in (16) can be kept deliberately small. The algorithm has been applied successfully to different models resulting automatic selection of dominant states. For instance in section 7 by setting $\theta_{max} = \frac{\pi}{100}$ for a mechatronic protection valve, the algorithm identifies 8 state variables as dominant states. For further discussions and simulation results refer to section 7.

6 Structure simplification

Most of the models based on physical phenomena have simple inner structures and consist of sparse matrices, but after applying the system matrices optimization method, normally the system matrices of the reduced order system are full of nonzero elements. This corresponds to a high model complexity, since each non-zero element represents one internal coupling within the system.

By observing precisely the system matrices, it can be perceived that usually some of their elements are very small. Thus the conjecture lead us to the question that "Are they really valid numbers or they are just results of some errors or round off in numerical computations?". The authenticity of this guess can be checked by forcing the related elements to zero by *resolving the original optimization problem with appropriate constraints*. The question arises is "which elements are not significant and can be replaced by zeros with negligible impact on the approximation error?". To verify the significance of each element and its effect on the performance and quality of the reduced order system, a cost function is required. Afterward we should search between the possible options to find a set of elements that minimize the cost function. Often there are a large number of different choices to carry out this task and this number is related directly to the size and complexity of the original system. In fact it is not usually possible to check every single option independently and find the best solution, therefore some methods for pioneered searching such as genetic algorithm (GA) is demanded. These ideas and their mathematical background is presented in the succeeding subsections.

6.1 Secondary conditions

The general linear equality constrained minimization problem can be written as follows:

$$\textit{Find } X \textit{ such that it minimizes } \|\mathbf{A}X - \mathbf{B}\| \textit{ and fulfills the equality } \mathbf{C}X = \mathbf{D}$$

where $\mathbf{A}$ is an m-by-n matrix ($m \leq n$) and $\mathbf{C}X = \mathbf{D}$ defines a linear equality constraint. In [8, 14] some methods for solving this optimization problems are proposed. The ability to solve optimization problems with constraints can be used to combine some additional features to system matrices optimization method for order reduction and structure simplification. In [17] this basic idea is used for improving the steady state performance.

Application of Secondary Conditions to Structure Simplification: Since each non-zero element represents one internal coupling within the system, it is therefore appropriate to not only reduce the system order but also to keep the reduced system simple by aiming at a significant number of zero elements in $\mathbf{E}$ and $\mathbf{W}$. In order to achieve this, we first formulate complexity constraints on the reduced model (6) by the following secondary conditions:

$$\mathbf{e_i^T h_{e,i}} - \mathbf{l_{e,i}} = \mathbf{0^T}, \quad \mathbf{w_i^T h_{w,i}} - \mathbf{l_{w,i}} = \mathbf{0^T} \tag{20}$$

where $\mathbf{e_i^T}$ is the i_{th} row of matrix $\mathbf{E}$ and $\mathbf{w_i^T}$ is the i_{th} row of matrix $\mathbf{W}$. For instance for forcing the first element in the second row of matrix $\mathbf{A}$ (a_{21}) to zero, we can choose the following secondary conditions for the second row of matrix $\mathbf{E}$ ($\mathbf{e_2^T}$):

$$\mathbf{h_{e,2}} = [1, 0, \ldots, 0]^T,\ \mathbf{l_{e,2}} = [0]$$

The optimization problems (13) with secondary conditions of type (20) results in optimal solutions (21):

$$\begin{aligned}\mathbf{e_{opt_i}^T} &= \dot{\mathbf{x}}_{\mathbf{do_i}}^{\mathbf{T}}\mathbf{M^T}(\mathbf{MM^T})^{-1} + (\mathbf{l_{e,i}} - \dot{\mathbf{x}}_{\mathbf{do_i}}^{\mathbf{T}}\mathbf{M^T}(\mathbf{MM^T})^{-1}\mathbf{h_{e,i}})\cdot \\ &\qquad \cdot(\mathbf{h_{e,i}^T}(\mathbf{MM^T})^{-1}\mathbf{h_{e,i}})^{-1}\mathbf{h_{e,i}^T}(\mathbf{MM^T})^{-1} \\ \mathbf{w_{opt_i}^T} &= \mathbf{x_i^T}\chi_{\mathbf{do}}^{\mathbf{T}}(\chi_{\mathbf{do}}\chi_{\mathbf{do}}^{\mathbf{T}})^{-1} + (\mathbf{l_{w,i}} - \mathbf{x_i^T}\chi_{\mathbf{do}}^{\mathbf{T}}(\chi_{\mathbf{do}}\chi_{\mathbf{do}}^{\mathbf{T}})^{-1}\mathbf{h_{w,i}})\cdot \\ &\qquad \cdot(\mathbf{h_{w,i}^T}(\chi_{\mathbf{do}}\chi_{\mathbf{do}}^{\mathbf{T}})^{-1}\mathbf{h_{w,i}})^{-1}\mathbf{h_{w,i}^T}(\chi_{\mathbf{do}}\chi_{\mathbf{do}}^{\mathbf{T}})^{-1}\end{aligned} \tag{21}$$

Using secondary conditions presented in (20), we can force any element of system matrices to zero deliberately. But the problem is that we don't know which elements are not significant and can be replaced by zeros. Therefore we should search between the possible options to find the optimal places that can be replaced by zeros. If the suitable complexity constraints $\mathbf{l}$ and $\mathbf{h}$ are found, the optimization problems (12,13) with secondary conditions of type (20) result in a reduced simplified model.

6.2 Method for simplifying the reconstruction matrix $\mathbf{W_{nc}}$

Each row of $\mathbf{W_{nc}}$ shows the optimal estimation of the corresponding state variable in the original system based on the state variables of the reduced order system. In the case that the original state variable be one of the dominant state variables, the corresponding row of $\mathbf{W}$ has very simple structure as follows:

$$\mathbf{w_{i_{do}}^T} = \begin{pmatrix} 0 & \ldots & 0 & \underbrace{1}_{i_{th}col.} & 0 & \ldots & 0 \end{pmatrix}$$

therefore no further simplification is applicable. But other rows need to be checked for the possibility of simplification. For instance if one of the rows of $\mathbf{W_{nc}}$ looks like the following row vector:

$$\mathbf{w_j^T} = \begin{pmatrix} 4 & 3\times 10^{-17} & -7 & 3\times 10^{-14} & 10^{-14} \end{pmatrix}$$

our method tries to replace the very small elements (number 2,4 and 5) with zeros and simultaneously examine precisely the effect of this replacement on the approximation error. In order to carry out this task we define an acceptable error range (accuracy criterion) for each row as follows:

$$[\mathbf{Er_{Wi}},\ (1+k)\mathbf{Er_{Wi}}] \tag{22}$$

where $\mathbf{Er_{Wi}} = \|\ \mathbf{x_i^T} - \mathbf{w_{nc_i}^T}\chi_{\mathbf{do}}\|$ and parameter k can be any value greater than zero. The typical value of k is around 0.1 which shows losing the accuracy not more than 10 percent of the optimum answer, of course this value can be changed with regard to the application. This idea is graphically presented in Fig. 6.2. It means that we search for the simplest model among the simplified reduced order models that have acceptable performance (set A in the figure) defined by (22). To do so our algorithm replaces some elements with zeros by adding the corresponding (zero forcing) secondary conditions to (12) and solving (21) for a candidate $\tilde{\mathbf{w}}_\mathbf{i}^\mathbf{T}$. Then if the added secondary condition results in an approximation error ($\mathbf{Er}_{\tilde{\mathbf{W}}_\mathbf{i}}$) in the *Error Range* $[\mathbf{Er_{Wi}},\ (1+k)\mathbf{Er_{Wi}}]$, it calculates the following simplicity cost function:

$$F = \textit{number of nonzero elements} \tag{23}$$

otherwise F will be set to zero. For small matrices it is possible to check all the possible (zero forcing) secondary conditions and compare the results with respect to their simplicity cost. Thus the row with

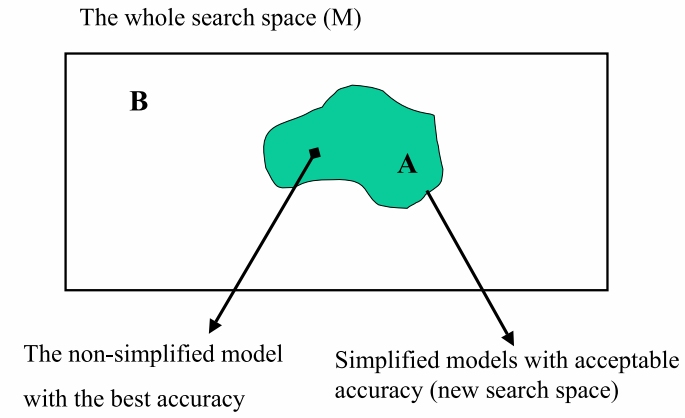

Figure 2: Error bound fitness function's effect on the search space

the higher cost function has the highest simplicity and at the same time it doesn't lose much accuracy. But when the number of different options exceeds a limit, we propose to use GA. Then, the steps and algorithm of this search are as follows:

1. Generate a starting population of genes
2. Evaluate the quality index of genes (Fitness Functions)
   - Construct the secondary conditions ($\mathbf{h_{w,i}}, \mathbf{l_{w,i}}$) corresponding to each gene
   - Calculate the simplified $\mathbf{W_i}$ using (21)
   - Evaluate each gene's fitness function $\mathbf{F}$, using (22,23)
3. Sort the genes regarding their fitness values
4. if (the breaking condition is fulfilled)
   choose the row vector corresponding to the best gene
   else
   Generate a fitter population using genetic operators and go to 2).

In this GA, a gene is a binary string of length n, where each bit indicates, whether the corresponding element of w_j^T should be enforced to zero or not. For more details see [3, 4, 5] and section 6.4.

6.3 Methods for simplifying the reduced order system matrices $\tilde{\mathbf{E}}_{\mathbf{nc}}$

Each row of $\tilde{\mathbf{E}}_{\mathbf{nc}}$ shows the optimal estimation of the corresponding state variable's derivative in the original system. Similar to the previous step, our method tries to replace very small elements of each

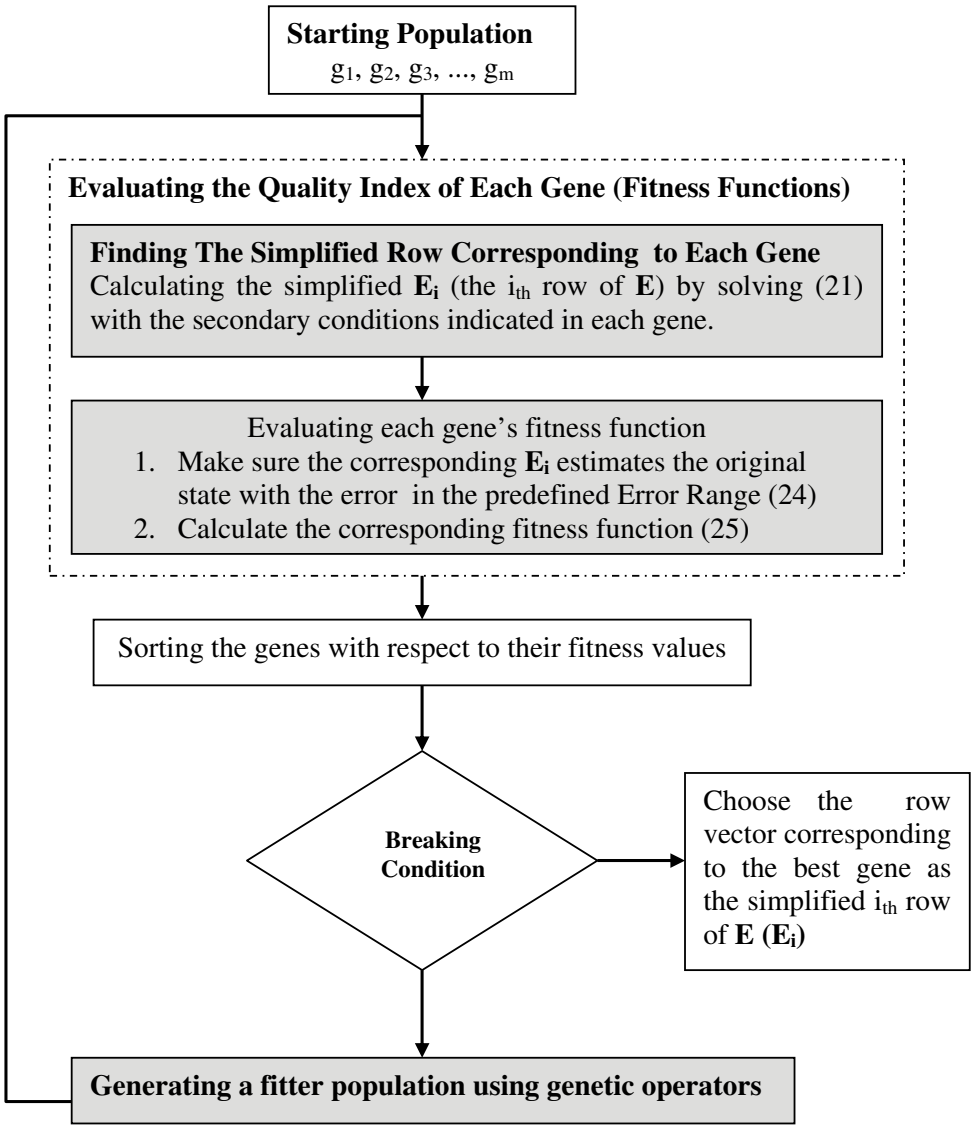

Figure 3: Structure simplification flow chart using GA for each row of $\tilde{\mathbf{E}}$.

row of $\tilde{\mathbf{E}}$ with zeros and simultaneously examine the effect of this replacement on the approximation error. The acceptable error range for each row is defined as follows:

$$[\mathbf{Er_{Ei}}\ ,\ (1+k)\mathbf{Er_{Ei}}] \tag{24}$$

where $\mathbf{Er_{Ei}} = \min_{\mathbf{e_i^T}} \parallel \dot{\mathbf{x}}^{\mathbf{T}}_{\mathbf{do_i}} - \mathbf{E}^{\mathbf{T}}_{\mathbf{nc_i}}\mathbf{M}\parallel$. At first our algorithm replaces some elements with zeros by adding corresponding (zero forcing) secondary conditions to (13) and solving it. Then it checks the accuracy criterion and calculates the following cost function:

$$F = \textit{number of nonzero elements} \tag{25}$$

The steps and algorithm of this search is depicted as a flow chart in Fig. 3, which is an alternative to the above gray shaded scheme.

6.4 How does genetic algorithm help structure simplification?

Genetic algorithms apply the principles of evolution present in nature to find solutions to optimization problems. This is achieved by maintaining a population of best solutions during searches. A string with a fixed bit-length usually represents a solution. In order to evaluate each potential solution, genetic algorithms need a fitness function that assigns a scalar fitness value to any particular solution. Once the representation scheme and fitness function are determined, a GA can start searching. Initially, often at random, GAs create a certain number of strings, called the population size, to form the first population generation. Next, the fitness function is used to evaluate each solution in this first generation. Better solutions obtain higher fitness values. Then, on the basis of these evaluations, genetic operators such as selection, recombination, and mutation are applied to produce the next generation. While recombination operator is very helpful in the local search, mutation helps in expanding the search area, and thus exploring the whole search space. The procedures of evaluation and generation are iteratively performed until the optimal solution is found, or until the time allotted for computation is over.

How Does GA Work: In this part the basic steps involved in any GAs are illustrated. These steps include:

1. Defining the problem in a gene representation: A suitable representation of the solution of the optimization problem should be defined in the form of a binary string, consequently the required gene length can be determined.
2. Defining a fitness function: The fitness function assigns a scalar fitness value for each gene, in such a way that this value reflects the goodness of the solution. The input of the fitness function is a binary gene and the output should be a scalar value.
3. Creating an initial random population: The initial population can be created randomly or based on some intuitional knowledge.
4. Setting the evolution parameters: The evolution parameters allow us to control the evolution process, by controlling the genetic operators. Recombination probability, recombination number, mutation number, mutation probability and elite number are the more useful parameters that can be tuned for improving a GA.
5. Evolving a fitter population: Starting from the initial population, a new population is created by applying the genetic operators such as recombination, mutation, elitism on the parent genes. The genes inside the new population are sorted according to the fitness value in an ascending or descending order depending on the problem that we are looking for the minimum or the maximum.
6. Repeating the evolution process until the fitness criteria are met, or until the pre-specified number of iterations is reached.

These steps are graphically depicted in Fig. 4. With respect to the previous sections, suitable choices of $\mathbf{l}$ and $\mathbf{h}$ are needed as candidates for the optimal simplified reduced order system. GAs can be used to search between different options. In this method each option is presented in form of a bit string (so-called individual) that only consists of ones and zeros and ones show the places that zeros should be inserted in the corresponding row of matrices $\mathbf{E}$ and $\mathbf{W}$. For instance suppose the second row of matrix $\mathbf{W}$ found form (10) is equal to the following row vector:

$$\mathbf{W_2} = [\ 2.1 \qquad 5.10^{-5} \qquad 6.10^{-17}\ ]$$

The constraint for the GA that forces the element w_{23} of of matrix $\mathbf{W}$ to zero, can be presented by the following row vector:

$$\mathbf{g^T_{W_2}} = \underbrace{0}_{1_{st}\text{ col.}} \quad \underbrace{0}_{2_{nd}\text{ col.}} \quad \underbrace{1}_{3_{rd}\text{ col.}} = 001$$

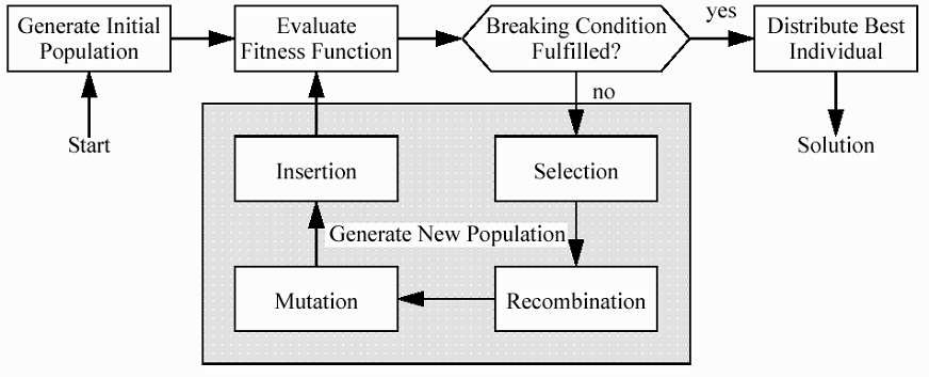

Figure 4: Typical genetic algorithms structure

Consequently every row vector that has the length three and contains only ones and zeros corresponds to a simplified structure of the second row of matrix $\mathbf{W}$. In an implementation of our institute, the starting population is selected randomly[1] and the tournament selection, two point cross over and normal mutation are used as genetic operators [18] and the GA produces new generations with better and better individuals as long as the breaking condition (number of produced generations) is not fulfilled.

6.5 Improving the results by Tabu search

Tabu search is a kind of iterative search and it is able to eliminate local minima and to search areas beyond a local minimum [22]. Therefore, it increases the possibility of finding the global minimum of a multi-modal search space. This search method can be used, as a complement for GA in structure simplification, by calculating every solution in the neighborhood of the current solution and selecting the best one for the next iteration step. This means that the Tabu search algorithm selects the way that produces the most improvement or the least deterioration. The advantage of Tabu search is the ability to find better solutions in local regions nearby the current solution. Therefore we recommend to solve the introduced problem of structure simplification in two steps by exploiting the strong points of both search algorithms. First by finding the region containing the global optimum or at least a very good suboptimal in the large search space by GA then finding the best solution in this local region by Tabu search algorithm.

6.6 Condensing the search space

In this part two ideas for accelerating the search procedure of GA for structure simplification based on the fitness function in (24,25) (or similarly in (22,23)) are proposed. These ideas reduce the size of search space during the search process by omitting improper (out of range) genes. The first idea is based on the fact that if an individual results in an approximation error greater than predefined value in (22,24), then any individual that has the similar format of gene (has at least the same number of ones in the same places) results in a greater approximation error which is surely out of acceptable range. For example if

$$g_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

results in an approximation error E_1, then any individual which has a gene of the following format:

$$g_i = \begin{bmatrix} * & 1 & * & * & 1 & * \end{bmatrix}$$

[1] It might be useful to add specific genes to the starting population that reflect structures close to $\mathbf{W}$ from (10). For instance in the example in 6.4, $\mathbf{g}_{\mathbf{W_2}}^{\mathbf{T}}$ is a good candidate to be added to the random starting population. However such ideas have not been deeply investigated.

where "*" could be either one or zero, causes an approximation error greater than E_1. For instance in the example, g_1 means forcing the second and the fifth element of a row in matrix $\mathbf{E}$ to zero, while g_i in addition forces some other elements to zero.

Similar to the first idea it can be shown that if an individual has an approximation error less than the predefined error bound, any individual that has the identical format of gene (has at least the same number of zeros in the same places), either is not in the range or has greater or equal fitness value. For example

$$g_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

results in the fitness value equal to 3, then we claim that any individual which has a gene of the following format:

$$g_i = \begin{bmatrix} * & * & 0 & 0 & * & 0 \end{bmatrix}$$

where "*" could be either one or zero, has a fitness values greater than 3 or it is not in the range. The proof is so simple, because if the the approximation error of an identical individual is not in the predefined range (error bound) then anyway it is not acceptable. Otherwise apparently its fitness value is higher because the related genes has less ones that results in more nonzero elements in the corresponding row and subsequently higher value of fitness according to (23). Using the above two ideas after each iteration lots of possible candidates will be omitted from the search list and that results in more effective search and it extremely shrinks the search space after a few iterations.

6.7 Enhancing the fitness function

So far, the fitness function was calculated by (22)-(25) without considering the state trajectories of the reduced model, i.e. without performing any additional simulation. For a more precise evaluation of the quality of approximation, one can simulate the original and the reduced order model and calculate the mean square root of the difference between the trajectories of the original state variables and their approximation from the reduced order model as follows:

$$F = \sum_{j=1}^{P} \sum_{i=1}^{\tilde{n}} \frac{\int_0^{t_{N_j}} [\dot{x}_{do_i}(t) - \mathbf{w_i}^T \tilde{\mathbf{x}}(t)]^2 dt}{\int_0^{t_{N_j}} \dot{x}_{do_i}^2(t)\, dt} \tag{26}$$

This fitness function is equal to the norm of approximation error and is superior to the one proposed in (22,24). Although using this fitness function sounds quite reasonable for measuring the performance of the reduced order system, it encounters several computation difficulties that tremendously increase the convergence time of the GA. In the GA each individual corresponds to a simplified system and in order to calculate (26), the corresponding simplified system ought to be simulated. In fact the simulation of the whole simplified systems (correspond to a new population) takes heavy computation effort and is a very time consuming task. With respect to this inconvenience, (26) can not be used as the main fitness function from the first beginning. Since $\mathbf{Er_{Wi}}$ and $\mathbf{Er_{Ei}}$ are just an approximation of the reduction quality, for achieving better results the structure simplification search process can be carried out in two step as follows:

1. Using (22,24) as the fitness function for the first iterations, till acceptable results are achieved.
2. By fixing the number of zero elements run the GA, using (26) as the fitness function and the results of the first search as starting population, for a few iterations.

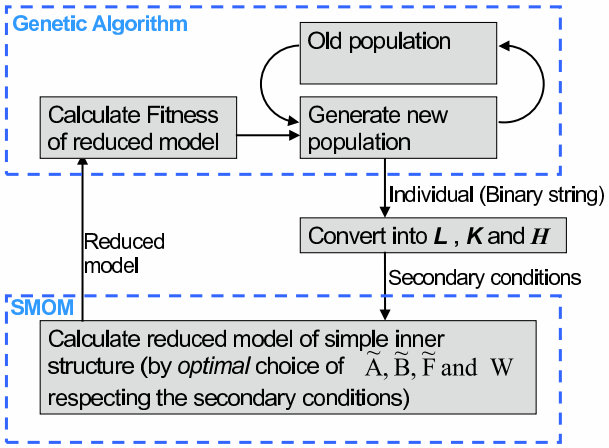

Figure 5: Structure simplification process

7 Illustrative examples

In order to implement our initial ideas, it was necessary to develop the required software and tools. To make clear the required tools, the whole process of structure simplification (which was described in the previous sections) is depicted as a simple flow chart in Fig. 5. The structure simplification process consists of (1) Modeling and simulation of typical nonlinear systems, (2) Order reduction of nonlinear systems with required constraints by system matrices optimization method (*SMOM*) and (3) genetic algorithm to carry out the search process. For each part a package in MATLAB environment is developed such that they are compatible with any nonlinear system in our scope of interest. For further information refer to the user manuals [11, 27] and [1]. In the following two examples illustrate the applicability of the proposed method and its effectiveness.

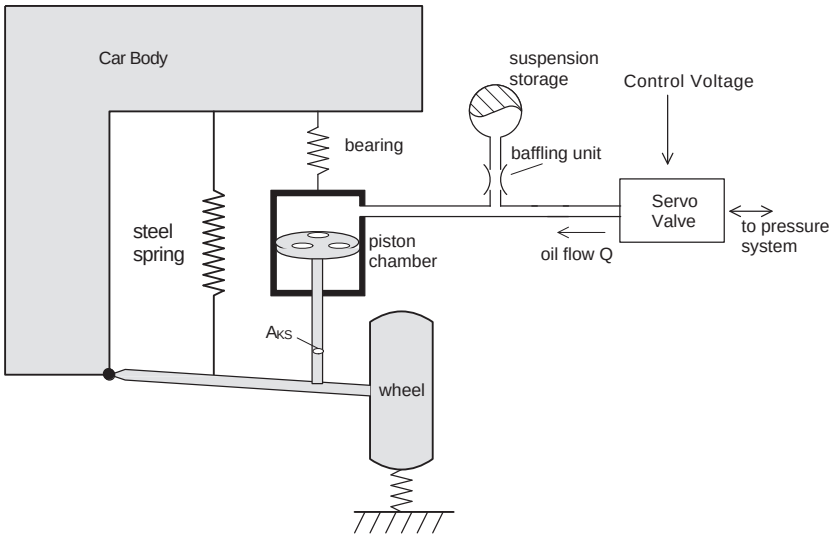

Figure 6: Construction of hydropneumatic vehicle suspension

7.1 Hydropneumatic vehicle suspension

Our method was tested on an active hydropneumatic shown in Fig. 6, which is described in more detail in [6, 15]. This device increases comfortableness and safety by significantly reducing the incongruous movements of the car body compared to a traditional passive spring shock-absorber system. The inputs of this system are the in and outflow of oil in the hydropneumatic system which should be regulated by the controller using measurement data from the sensors. In order to design and simulate such a controlled system a modelling of the system is necessary. The simulation can be accelerated through order reduction and furthermore it simplifies all control algorithms which directly use the model, such as state feedback observer combinations or model based feed forward controllers. Fig. 7 shows the block diagonal representation of this system including its nonlinear parts (more details in [16]). The system has a 10th order model with the system matrices shown below:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4127 & 0 & 127.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 12.75 & 0 & -1613 & 0 & 1600 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 533333 & 0 & -533333 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.244 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -162 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -8100 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 4000 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 8100 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -0.014 & -0.014 & -0.014 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.0005 & -0.0005 & -0.0005 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.66 & 0.66 & 0.66 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 17.5 & -71.43 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

There are seven dominant state variables, so the reduced order model of order 7 can be calculated. The

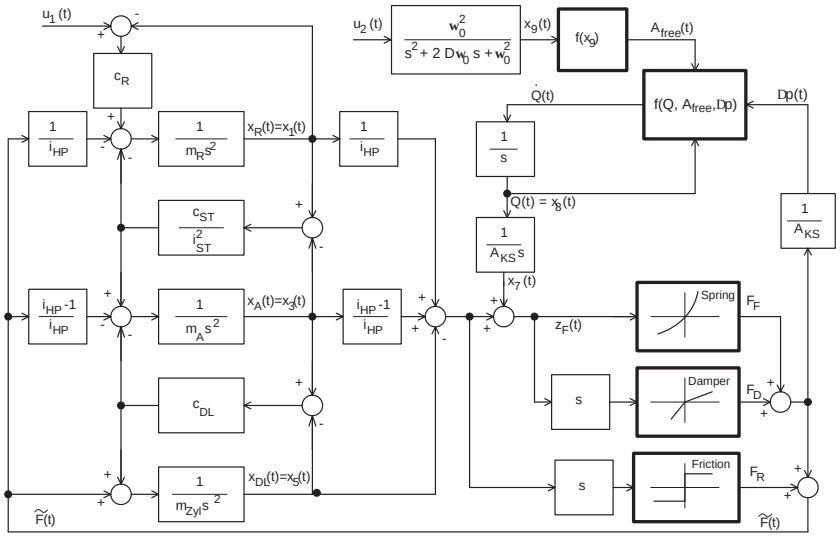

Figure 7: block diagram of hydropneumatic vehicle suspension model

order reduction method without structure simplification calculates the three systems matrices as follows:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 5.9e^{-13} & 1.0 & -3.8e^{-13} & 5.2e^{-14} & 6.7e^{-13} & 7.2e^{-14} & 2.5e^{-14} \\ -4.1e^{3} & -2.4e^{-11} & 1.3e^{2} & 1.2e^{-11} & -1.4e^{-9} & -7.9e^{-11} & 1.0e^{-11} \\ 3.1e^{-13} & -1.5e^{-13} & -8.5e^{-14} & 1.0 & 3.2e^{-13} & 6.9e^{-14} & 1.9e^{-14} \\ 1.3e^{1} & 7.6e^{-3} & -1.1e^{1} & -3.6e^{-2} & -1.3 & 8.8e^{-2} & -8.6e^{-2} \\ -4.9e^{-14} & 7.1e^{-15} & 2.9e^{-14} & 5.0e^{-15} & 4.9e^{-14} & 1.2 & -2.3e^{-15} \\ -2.3e^{-12} & -1.1e^{-14} & 1.9e^{-12} & 7.3e^{-14} & -2.5e^{-12} & 3.5e^{-13} & -7.8e^{-14} \\ -1.4e^{1} & 1.2 & 1.5e^{1} & -3.0 & -2.1e^{1} & -1.4e^{1} & -1.9e^{1} \end{bmatrix}$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} -2.3e^{-14} & 4.7e^{-15} \\ 4.0e^{3} & -4.7e^{-13} \\ 6.1e^{-15} & 4.7e^{-15} \\ -1.3 & 1.8e^{-2} \\ 9.7e^{-15} & 9.4e^{-15} \\ 3.2e^{-13} & -3.5e^{-4} \\ 8.0e^{-1} & 2.8e^{1} \end{bmatrix}, \quad \tilde{\mathbf{F}} = \begin{bmatrix} 5.0e^{-17} & 4.1e^{-16} & 4.2e^{-17} & -8.4e^{-14} & -3.5e^{-14} \\ -1.5e^{-2} & -1.5e^{-2} & -1.5e^{-2} & -1.3e^{-11} & -8.4e^{-12} \\ 4.5e^{-17} & 3.6e^{-16} & 4.3e^{-17} & -7.3e^{-14} & -5.1e^{-14} \\ 1.6e^{-3} & 1.4e^{-3} & 1.5e^{-3} & 3.0e^{-2} & 2.0e^{-1} \\ 2.6e^{-18} & 5.3e^{-18} & -2.8e^{-18} & 6.7e{-}15 & 2.0e^{-14} \\ 8.8e^{-17} & -2.5e^{-16} & 1.8e^{-17} & 1.8e^{1} & -7.1e^{1} \\ 2.3e^{-4} & -2.1e^{-3} & -3.3e^{-4} & 3.2 & -6.9e^{1} \end{bmatrix}$$

The approximation of the original model is good, but the complexity of the reduced model is very high because all of the 98 elements of these matrices are non-zero. Using the simplification method presented through this report a reduced and simplified model of order 7 was achieved, where $\mathbf{R}$ was chosen as in [16]. To find the simple structure system, the GA needed to calculate approximately 120 solutions for the global search and the Tabu search algorithm calculated 15 solutions for the local search within a search space containing $7 \times 2^{14} \approx 114.6 \times 10^3$ possible solutions[2]. The system matrices of the resulted model are as follows:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ -4127.5 & 0 & 127.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -16.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -31.6 \end{bmatrix}$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} 0 & 0 \\ 4000 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 33.5 \end{bmatrix}, \quad \tilde{\mathbf{F}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -0.015 & -0.015 & -0.015 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.0024 & 0.0013 & 0.0015 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 17.5 & -71.4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0.021 & 0.003 & 0.980 & -0.002 & 0.026 & -0.010 & 0.003 \\ -1.734 & 0.146 & 1.189 & 1.022 & -1.671 & 2.570 & -0.5386 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -56.291 & -1.657 & 54.279 & 1.582 & -72.199 & -54.848 & 178.019 \end{bmatrix}$$

where 81 elements are forced to zero and the interaction between the states are extremely decreased. To illustrate the quality of approximation the original system, the reduced system of order 7 and its simplified form are excited with the input $\mathbf{u}(t) = [0.1\sigma(t) \quad 0]^T$, where $\sigma(t)$ denotes the step signal. The time curves of dominant states $\mathbf{x_2}, \mathbf{x_3}, \mathbf{x_5}$ and $\mathbf{x_6}$ are depicted in Fig. 8. The results show that the reduced simplified system, found by the two search algorithms, is less accurate than original reduced system, but the error lies in an acceptable range. It is evident that this error is the price paid for much higher simplicity.

[2] In [5] and [6] the search space is $2^{7(7+2+5)} = 2^{98}$, because the row by row simplification was not used!

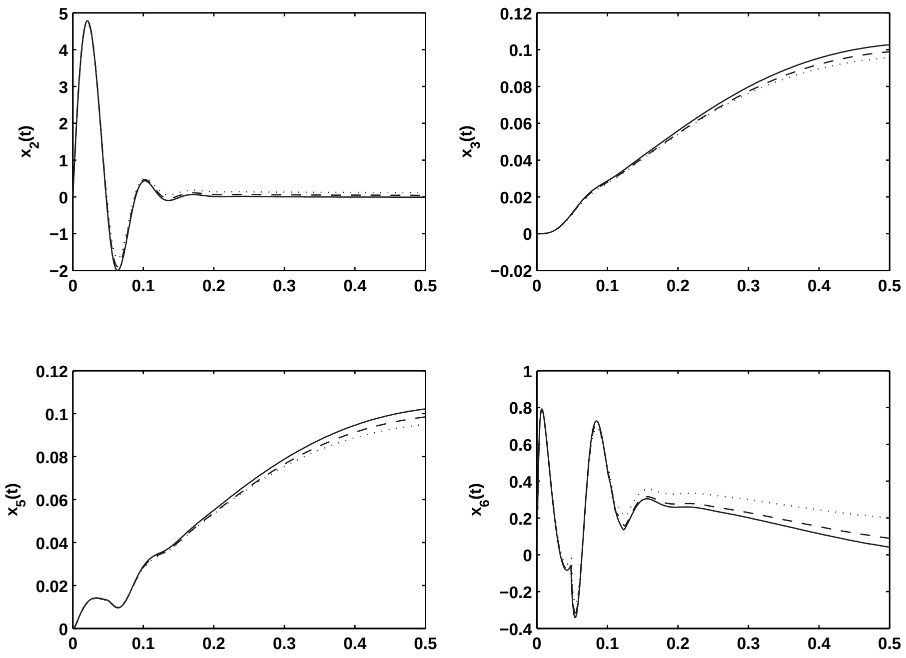

Figure 8: Time curves of four dominant state variables of the original system (solid), reduced order system of 7_{th} order derived by SMO without simplification (dash) and its simplified form (dotted) excited by $\mathbf{u}(t) = [0.1\sigma(t) \quad 0]^T$.

7.2 Mechatronic Protection Valve

Another example that was used for further implementation of the method is a mechatronic protection valve. The model is of order 11 and contains nine nonlinear equations. The mechanical structure of the model is shown in Fig. 9. The valve can be activated for pressure reduction by stimulating a pneumatic valve. The activation is realized by a rectangular pulse-function with an amplitude change from 0 to 25 Volts. For illustration of the reduction process, the main system matrix ($\mathbf{A}$) of the protection valve is shown below:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0088 & 0 & 0.0039 & 0 & -0.013 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

By applying *SMO* method in conjunction with *DSFA*, the model was reduced to order 8. The snapshots were generated by simulation of the original system with a sequence of pulse-functions representing a repeated switching of the valve. Then the snapshot matrices are evaluated with help of *DSFA*. The quality criterion in (17), (18) and (19) was set to $\theta_{max} = \frac{\pi}{100}$ and *DSFA* assigned the vector of dominant states

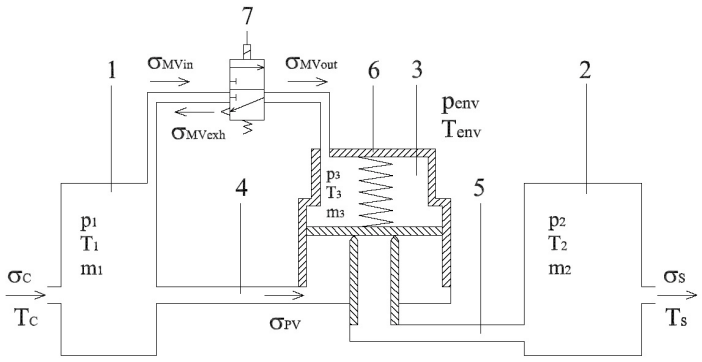

Figure 9: Mechatronic protection valve, curtesy of Huba Nemeth [19].

X_{do} of order $n' = 8$. The reduced main matrix ($\mathbf{A}'$) is as follows:

$$\mathbf{A}' = \begin{bmatrix} 4.5 \cdot 10^{-13} & 1 & 0 & -4.8 \cdot 10^{-14} & -9.6 \cdot 10^{-15} & 0 & 1.6 \cdot 10^{-20} & -1.6 \cdot 10^{-16} \\ -5 \cdot 10^{5} & -500 & 0 & -0.001 & -0.028 & 0 & 0.0038 & -8.3 \cdot 10^{-8} \\ -1.3 \cdot 10^{-10} & -1.4 \cdot 10^{-13} & 0 & 4.8 \cdot 10^{-12} & 1.7 \cdot 10^{-12} & 0 & -1.1 \cdot 10^{-18} & 7.2 \cdot 10^{-15} \\ -1.4 \cdot 10^{-12} & -1.8 \cdot 10^{-15} & 0 & 9 \cdot 10^{-14} & 2.4 \cdot 10^{-14} & 0 & -4.7 \cdot 10^{-21} & 1 \\ 1.7 \cdot 10^{-8} & 7.5 \cdot 10^{-11} & 0 & -6.5 \cdot 10^{-9} & -1.2 \cdot 10^{-9} & 0 & 7.5 \cdot 10^{-16} & -4 \cdot 10^{-12} \\ -0.069 & 4.2 \cdot 10^{-6} & 0 & 0.0020 & -1.4 \cdot 10^{-4} & 0 & 1.5 \cdot 10^{-9} & -2.5 \cdot 10^{-8} \\ 1.7 \cdot 10^{-5} & 2.1 \cdot 10^{-8} & 0 & -1.1 \cdot 10^{-6} & -2.9 \cdot 10^{-7} & 0 & 1.4 \cdot 10^{-13} & -1.6 \cdot 10^{-9} \\ 3.5 \cdot 10^{-9} & -1.4 \cdot 10^{-11} & 0 & -5.7 \cdot 10^{6} & 2.1 \cdot 10^{-10} & 0 & -2.3 \cdot 10^{-16} & -1000 \end{bmatrix}$$

In contrast to matrix $\mathbf{A}$ of the original system, matrix $\mathbf{A}'$ is filled with many nonzero elements. It has therefore a higher inner structure complexity. Usually not all of these elements share the same impact on the affiliated differential equation. To find a system with a less complex structure genetic algorithms in combination with an error bound fitness function as described in previous sections can be used to simplify the system. The result of this procedure is shown below.

$$\mathbf{A}'' = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -5 \cdot 10^{5} & -500 & 0 & -0.001 & -0.028 & 0 & 0.0038 & -8.3 \cdot 10^{-8} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.4 \cdot 10^{-14} & 0 & 0 & 1 \\ 0 & 7.5 \cdot 10^{-11} & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.069 & 0 & 0 & 0 & -1.4 \cdot 10^{-4} & 0 & 1.5 \cdot 10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5.7 \cdot 10^{6} & 0 & 0 & 0 & -1000 \end{bmatrix}$$

Fig. 10 shows the simulation results of the output (valve pressure) for the original model of order 11, the reduced model of order 8 and the reduced and simplified model of order 8 for a pulse function with duration of $20ms$. The resulting trajectories show that the original system has been well approximated by the reduced system. Furthermore, applying structure simplification to the reduced system does not much aggravate the approximation quality. The error of both reduced systems, referring to the angle between the subspace of $\mathbf{y}(t)$ and $\tilde{\mathbf{y}}(t)$, is $0.4 \cdot \pi/100$. Also, by order reduction from 11 to 8, the required storage capacity was decreased to 63% of the original system and the simulation effort was decreased by 30%.

Experiments in reducing to orders less than 8 without significant loss of accuracy were not successful and in some cases resulted in unstable models. We believe that it roots in the domination of nonlinear parts which has also reflected in the almost *empty* matrix $\mathbf{A}$.

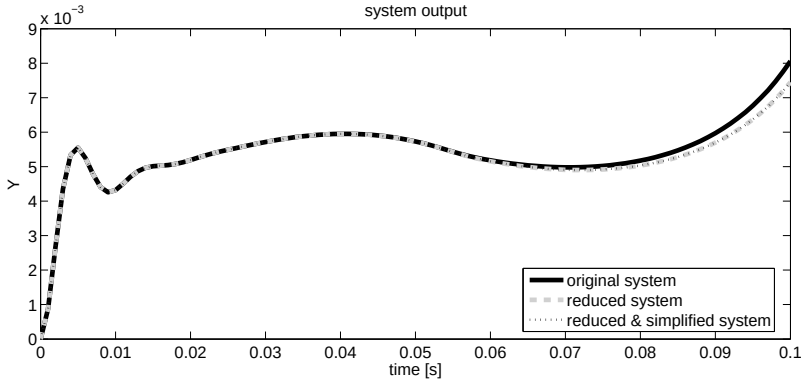

Figure 10: The graph shows the time-response of the original and the reduced systems for a rectangular pulse-function of $20[ms]$ duration.

8 Conclusion

The primary method of system matrices optimization for order reduction of nonlinear systems may not be efficient for big nonlinear systems. In this report we introduced the row by row method which contracts the optimization problem and opens new fields to improve the results. Then an add-on simplification scheme was presented that delivers models of reduced order and simple inner structure at the same time. Model structures were coded in binary strings and optimized using Genetic Algorithms. A special fitness functions was suggested that reduces the computational effort drastically while still delivering good approximation results in practice. In additions some routines for sorting out unacceptable selections during the search procedure were proposed that increase the pace of convergence considerably. It should be also noted that one of the advantages of the new method in comparison to the old ideas is *simulation free* concept reflected in the fitness functions (22) and (24). In this algorithm only the snapshots of the original system are required and no further simulation of the original system or the reduced order system is necessary. Future work shall focus on the suitable choices of starting populations of the genetic algorithm, gained from the original model for instance, and on the application of the method to even larger and more challenging practical engineering problems.

A Appendix

Proof of Theorem 1:
The proof based on the following lemma

Lemma 1. the matrix $\mathbf{Xopt} = \begin{bmatrix} \mathbf{x_{opt_1}} \\ \vdots \\ \mathbf{x_{opt_n}} \end{bmatrix}$ is the optimal answer of least square problem

$$\min \| \mathbf{A} - \mathbf{XB} \|$$

if and only if $\mathbf{x_{opt_i}}$ is the optimal answer of the least square problem

$$\min_{\mathbf{x}} \| \mathbf{A_i} - \mathbf{xB} \|, \qquad i = 1, 2, \ldots, n$$

where $\mathbf{A_i}$ and $\mathbf{x_{opt_i}}$ are the i_{th}row of matrices $\mathbf{A}$ and $\mathbf{X_{opt}}$.

Proof: Lemma1 permits us to write:

$$\begin{aligned}\min_{\mathbf{X}} \|\mathbf{A}-\mathbf{XB}\| = \|\mathbf{A}-\mathbf{X_{opt}}\,\mathbf{B}\| &= \left\|\mathbf{A}-\begin{bmatrix}\mathbf{x_{opt_1}}\\ \vdots \\ \mathbf{x_{opt_n}}\end{bmatrix}\mathbf{B}\right\| \\ &= \left\|\begin{bmatrix}\mathbf{A_1}-\mathbf{x_{opt_1}B}\\ \vdots \\ \mathbf{A_n}-\mathbf{x_{opt_n}B}\end{bmatrix}\right\| \\ &= \sum_{i=1}^{n}\|\ \mathbf{A_i}-\mathbf{x_{opt_i}B}\| = \sum_{i=1}^{n} J_i\end{aligned} \tag{27}$$

and

$$\min_{\mathbf{x_i}} \|\mathbf{A_i}-\mathbf{x_iB}\| = \|\mathbf{A_i}-\mathbf{\tilde{x}_{opt_i}}\,\mathbf{B}\| = \tilde{J}_i,\ i=1,\cdots,n \tag{28}$$

We would like to prove that:

$$\mathbf{\tilde{x}_{opt_i}} = \mathbf{x_{opt_i}} \quad \text{and} \quad J_i = \tilde{J}_i,\ i=1,\cdots,n$$

so if we suppose $J_i > \tilde{J}_i$ then we build the matrix $\mathbf{\hat{x}_{opt}} = \begin{bmatrix}\mathbf{x_{opt_1}}\\ \vdots \\ \mathbf{\tilde{x}_{opt_i}}\\ \vdots \\ \mathbf{x_{opt_n}}\end{bmatrix}$, then

$$\|\mathbf{A\hat{x}_{opt}}-\mathbf{B}\| = J_1+\cdots+\tilde{J}_i+\cdots+J_n < \sum_{i=1}^{n} J_i$$

that it contradicts with (27). Similarly if we suppose that $J_i < \tilde{J}_i$ then

$$\|\mathbf{A_i x_{opt}}-\mathbf{B}\| = J_i < \tilde{J}_i$$

that it contradicts with (28), therefore $J_i = \tilde{J}_i$.
According to the fact that the answer of $\min \|\mathbf{A_i}-\mathbf{x_iB}\|$ is unique and we have proved that:

$$\|\mathbf{A_i}-\mathbf{x_{opt_i}B}\| = \|\mathbf{A_i}-\mathbf{\tilde{x}_{opt_i}B}\| = J_i = \tilde{J}_i$$

thus $\mathbf{\tilde{x}_{opt_i}} = \mathbf{x_{opt_i}}$ which completes the proof of the theorem.■

References

[1] Albunni, M. A. and Yousefi, A. *Genetic Algorithms Toolbox for MATLAB.* University of Bremen, 2003.

[2] Bjorck, A. and Golub, G. Numerical methods for computing angles between linear subspaces. *Math. Comp. 27*, pages 579–594, 1973.

[3] Buttelmann, M. and Lohmann, B. Model simplification and order reduction of nonlinear systems with genetic algorithms. In *Proceedings of the IMACS Symposium on Mathematical Modelling,3rd MATHMOD*, pages 777–781, Vienna, 2000.

[4] Buttelmann, M. and Lohmann, B. Non-linear model reduction by genetic algorithms with using a system structure related fitness function. In *European Control Conference (ECC), Porto, Portugal*, pages 1870–1875, 2001.

[5] Buttelmann, M. and Lohmann, B. Two optimization methods for solving the problem of structure simplification of nonlinear systems. In *8th IEEE International conference on Methods and Models in Automation and Robotics*, pages 1257–1262, Poland, 2002.

[6] Buttelmann, M. and Lohmann, B. Optimierung mit Genetischen Algorithmen und eine Anwendung zur Modellreduktion. *Automatisierungstechnik (at)*, 52(4):151–163, 2004.

[7] Chu, M.T., Funderlic, R.W., and Golub, G.H. A rank-one reduction formula and its applications to matrix factorizations. *SIAM Review*, 37:512–530, 1995.

[8] Fletcher, R. *Practical Methods of Optimization, Vol. 1, Unconstrained Optimization, and Vol. 2, Constrained Optimization*. John Wiley and Sons., 1980.

[9] Golub, G. H. and Van Loan, C. F. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[10] Hubert, L., Meuleman,J., and Heiser, W. Two purposes for matrix factorizations: A historical appraisal. *SIAM Review*, 37:68–82, 2000.

[11] Jung, D. and Yousefi, A. *A MATLAB-Based Package for System Matrices Optimization method*. Technical University of Munich, 2005.

[12] Kagstrom, B and Ruhe, A. On angles between subspaces of a finite dimensional inner product space. *Lecture Notes in Mathematics 973, Springer*, pages 263–285, 1983.

[13] Kokotovic, P.V, O'Malley, R.E., and Sannuti, P. Singlar perturbation and order reduction in cnontrol theory - an overview. *Automatica*, 12:123–132, 1976.

[14] Lawson, C.L. and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

[15] Lohmann, B. *Ordnungsreduktion und Dominanzanalyse nichtlinearer Systeme*, volume 8 of *VDI-Verlag*. VDI-Fortschrittsberichte, Dusseldorf, 1994.

[16] Lohmann, B. Application of model order reduction to a hydrpneumatic vehicle suspension. *IEEE Transactions on Control Systems Technology*, 3(2):102–109, March 1995.

[17] Lohmann, B. Order reduction and determination of dominant state varialbes for nonlinear systems. *Mathematical Modeling of Systems*, 1(2):77–90, 1995.

[18] Mitchell, M. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, 1996.

[19] Nemeth, H. *Nonlinear Modelling and Control of a Mechatronic Protection Valve*. PhD thesis, Budapest University of Technology and Economics, Vehicles and Mobile Machines Ph.D. School, 2004.

[20] Newman, A. J. Model reduction via the karhunen-loeve expansion, part i: An exposition. Technical report, University of Maryland, 1996.

[21] Newman, A. J. Model reduction via the karhunen-loeve expansion, part ii: Some elementary examples. Technical report, University of Maryland, 1996.

[22] Pham, D. T. and Karaboga, D. *Inteligent Optimisation Techniques - Genetic Algorithms, Tabu Search, Simulated Anealing and Neural Networks*. Springer, 2000.

[23] Saksena, V.R., O'Reilly, J., and Kokotovic, P.V. Singlar perturbation and time-scale methods in cnontrol theory - survey. *Automatica*, 20:273–293, 1984.

[24] Scherpen, J. M. A. Balancing for nonlinear systems. *System & Control Letters*, 21:143–153, 1993.

[25] Scherpen, J. M. A. *Balancing for Nonlinear Systems*. PhD thesis, University of Twente, 1994.

[26] Volkwein, S. Proper orthogonal decomposition and singular value decomposition. Technical report, Graz University, 1999.

[27] Yousefi, A. *Nolinear Systems Toolbox for MATLAB*. University of Bremen, 2003.

[28] Yousefi, A. and Lohmann, B. Structure simplification of reduced order nonlinear systems using efficient fitness functions and search spaces. In *The 10th IFAC Symposium on Large Scale Systems:Theory and Applications*, Osaka, Japan, 2004.

[29] Yousefi, A., Lohmann, B., and Buttelmann, M. Row by row structure simplification. In *The American Control Conference*, Boston, USA, 2004.