Design and implementation of a soft computing-based controller for a complex mechanical system

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Maschinenbau und Verfahrenstechnik
der
Universität Duisburg-Essen
zur Erlangung des akademischen Grades
eines
Doktors der Ingenieurwissenschaften
Dr.-Ing.
genehmigte Dissertation

von

Hammam Tamimi
aus
Hebron, Palästina

Gutachter: Univ.-Prof. Dr.-Ing. Dirk Söffker
Univ.-Prof. Dr. rer. nat. Johannes Gottschling
Tag der mündlichen Prüfung: 22.09.2016

Kurzfassung

Soft-Computing basierende Regler beinhalten Algorithmen, die im Bereich des Maschinellen Lernens einzuordnen sind. Diese Regler sind in der Lage eine geeignete Steuerungsstrategie durch direkte Interaktion mit einer dynamischen Regelstrecke zu entwerfen. Sowohl klassische als auch moderne Reglerentwurfsmethoden hängen von der Genauigkeit des verwendeten dynamischen Systemmodells ab, was insbesondere bei steigender Komplexität des Systems und auftretenden Modellunsicherheiten nicht mehr uneingeschränkt gewährleistet werden kann. Die Ziele von Soft-Computing basierenden Reglern sind die Verbesserung der Güte des Regelverhaltens und eine geeignete Anpassung der Regler ohne eine mathematische Modellbildung auf Grundlage von physikalischen Gesetzen.

Im Rahmen dieser Arbeit werden fünf Algorithmen zur Modellbildung und Regelung dynamischer Systeme untersucht, welche auf dem Mehrschichten-Perzeptron-Netzwerk (Multi-Layer Perceptron network, MLP), auf der Methode der Support Vector Machine (SVM), der Gauß-Prozesse, der radialen Basisfunktionen (Radial Basis Functions, RBF) sowie der Fuzzy-Inferenz-Systeme basieren. Im Anschluss an die Darstellung der zugrunde liegenden mathematischen Zusammenhänge dieser Methoden sowie deren Hauptanwendungsfelder im Bereich der Modellbildung und Regelung dynamischer Systeme wird eine systematische Evaluierung der fünf Methoden diskutiert. Anhand der Verwendung quantitativer Gütekennziffern werden diese Methoden für die Verwendung in der Modellbildung und Regelung dynamischer Systeme vergleichbar gegenübergestellt. Basierend auf den Ergebnissen der Evaluierung wird der SVM-basierte Algorithmus als Kernalgorithmus des Soft-Computing basierenden Reglers verwendet.

Der vorgestellte Regler besteht aus zwei Hauptteilen, wobei der erste Teil aus einer Modellfunktion der dynamischen Regelstrecke und einem SVM-basierten Beobachter besteht, und der zweite Teil basierend auf dem Systemmodell eine geeignete Regelstrategie generiert. Die Verifikation des SVM-basierten Regleralgorithmus erfolgt anhand eines FEM-Modells eines dynamischen elastischen Balken bzw. einseitig eingespannten elastischen Balkens. Dieses Modell kann z. B. als Ersatzmodell für das mechanische Verhalten eines flexiblen Roboterarms oder einer Flugzeugtragfläche verwendet werden. Der Hauptteil der Modellfunktion besteht aus einem automatischen Systemidentifikationsalgorithmus, der auch die Integration eines systematischen Modellbildungsansatzes für dynamische Systeme ermöglicht.Die Ergebnisse des SVM-basierten Beobachter zeigen ähnliches Verhalten zum Kalman-Bucy Beobachter. Auch die Sensitivitätsanalyse der Parameter zeigt eine bessere Güte der SVM-basierten Beobachter im Vergleich mit den Kalman-Bucy Beobachtern.

Im Anschluss wird der SVM-basierte Regler zur Schwingungsregelung des Kragträgers verwendet. Hierbei werden vergleichbare Ergebnisse zum LQR-Regler erzielt. Eine experimentelle Validierung des SVM-basierten Reglers erfolgt an Versuchsständen eines elastischen Biegebalkens sowie eines invertierten Biegebalkens. Die Zustandsbeobachtung führt zu vergleichbaren Ergebnissen verglichen mit einem Kalman-Bucy Beobachter. Auch die Modellbildung des elastischen Balkens führt zu guten Übereinstimmungen. Die Regelgüte des Soft-Computing basierenden Reglers wurde am Versuchsstand des invertierten Biegebalkens experimentell erprobt. Es wird deutlich, dass Ergebnisse im Rahmen der erforderlichen Vorgaben erzielt werden können.

Abstract

The focus of this thesis is to obtain a soft computing-based controller for complex mechanical system. soft computing-based controllers are based on machine learning algorithm that able to develop suitable control strategies by direct interaction with targeted dynamic systems. Classical and modern control design methods depend on the accuracy of the system dynamic model which cannot be achieved due to the dynamic system complexity and modeling uncertainties. A soft computing-based controller aims to improve the performance of the close loop system and to give the controller adaptation ability as well as to reduce the need for mathematical modeling based on physical laws.

In this work five different softcomputing algorithms used in the field of modeling and controlling dynamic systems are investigated. These algorithms are Multi-Layer Perceptron (MLP) network, Support Vector Machine (SVM), Gaussian process, Radial Basis Function (RBF), and Fuzzy Inference System (FIS). The basic mathematical description of each algorithm is given. Additionally, the most recent applications in modeling and controlling of dynamic system are summarized. A systematic evaluation of the five algorithms is proposed. The goal of the evaluation is to provide quantitative measure of the performance of soft computing algorithms when used in modeling and controlling a dynamic system. Based on the evaluation, the SVM algorithm is selected as the core learning algorithm for the soft computing-based controller.

The controller has two main units. The first unit has two functions of modeling dynamic system and obtaining a SVM-based observer. The second unit is in charge of generating suitable control strategy based on the dynamic model obtained. The verification of the controller using SVM algorithm is done using an elastic cantilever beam modeled using Finite Element Method (FEM). An elastic cantilever beam can be considered as a representation of flexible single-link manipulator or aircraft wing. In the core of the modeling unit, an automatic system identification algorithm which allows a systematic modeling approach of dynamic systems is implemented. The results show that the system dynamic model using SVM algorithm is accurate with respect to the FEM model. As for the SVM-based observer the results show that it has good estimation in comparison with to different Kalman-Bucy observers. The sensitivity to parameters variations analysis shows that the SVM-based observer has better performance than Kalman-Bucy observer. The SVM-based controller is used to control the vibration of the cantilever beam; the results show that the model reference controller using SVM has a similar performance to LQR-controller.

The validation of the controller using SVM algorithm is carried out using the elastic cantilever beam test rig and the inverted cantilever beam test rig. The states estimation using SVM-based observer of the elastic cantilever beam test rig is successful and accurate compared to a Kalman-Bucy observer. Modeling of the elastic

cantilever beam using the SVM algorithm shows good accuracy.
The performance of controller is tested on the inverted cantilever beam test rig. The results show that required performance objective can be realized using this control strategy.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

Many of the dynamical processes that have to be controlled are complex, highly nonlinear, non-stationary and stochastic. Classical and modern control techniques have attempted to deal with such systems through the robustness properties [Har94]. Such methods are applicable to processes whose variation from linear time-invariant in relatively small and bounded within known limits; however, many processes are of such complexity that they defy representation by linear time invariant analytical models [Har94]. Typically control engineers make assumptions about the system dynamics in order to be able to establish a suitable mathematical model of the system, the operation conditions, and the effect of the environment on the system model. Based on these assumptions the controller is designed and tuned to have the desired performance within the assumed variations. Unfortunately, for complex dynamic system the variations are not always small, as a result of this the controllers are not always successful in achieving their goals.

An alternative approach to classical model-based control is to design controllers whose structure and consequent outputs in response to external commands are determined by experiential evidence, i.e. the observed input/output behavior of the plant, rather than by reference to a mathematical or model-based description of the controller. The controller is then a so-called intelligent controller [Har94].

Intelligent controllers employ techniques that can sense and be rational their environment and execute commands in a situative, adaptive and robust manner. Also it should be noted that intelligent controllers are not defined in terms of specific algorithms. The desire to increase systems autonomy motivates research to integrate more fields of science into the controller kernel that include artificial intelligence and signal processing, data fusion system, situation assessment.

Motivated to design controller that react properly when the targeted dynamic system suffer from major variations from its original situation, has the ability to associate its knowledge to face new situation, and has the technical awareness of its performance; control engineers integrated the field of machine learning science in order to be able to realize the task at hand.

A soft computing-based controller is an intelligent agent programmed using machine learning algorithm. Intelligent Agent (IA) is a term used in Artificial Intelligence. According to [RN15], IA is defined as an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals. IA should also be able to learn and use knowledge to achieve their goals.

The characteristics of IAs should be extended to soft computing-based controllers, then according to [KK98] a soft computing-based controller exhibits the following characteristics:

- Incrementally provide solutions for new problems,
- Online and in real-time adaptation ability,
- Self-performance evaluation in term of behavior, error and success,
- Continuous learning and improvement via direct interaction with environment,
- Quick learning from large amount of data.
- Abel to store models and patterns and have retrieval capacities.
- Representation of long and short term memory, age and forgetting through parameters.

In literature, two approaches to achieve soft computing-based controllers can be found; the first approach is called Human Simulated Intelligent Controller (HSIC) [LCLI97, YLX$^+$09, CGH08, LSC08], and the second approach is called the Brain Emotional Learning Based Intelligent Controller (BELBIC) [JAL10, JRAL09, DM14, KL09, YM11].

The HSIC theory aims at emulating human behavior based on the human macro control structure, and starts with the lowest level of hierarchical intelligent control, constructs the model of the ”motor sensory preview intelligence” in manual control. The HSIC controller possesses excellent control properties like a human operator as shown in [LCLI97].

The HSIC approach is different from the conventional approaches. It possesses the property of the parallel problem solving, logic and language control, and also includes of the analytical quantitative control of the traditional ways[LCLI97]. According to [LCLI97] the HSIC approach holds the following fundamental structures and functions:

- Information processing and decision making with hierarchical structure,
- Characteristic identification and characteristic memory on line,
- Multi-mode control with combination of open and close loop control, positive and negative feedback control qualitative decision and quantitative control,
- Using heuristic and intuition. these fundamental structures and function are given in [LCLI97], and are repeated here.

The HSIC is implemented on variety of systems such as the inverted pendulum on a cart which is a nonlinear, under-actuated system with the goal of swinging up and stabilizing the pendulum on the upright position [LCLI97], a brushless DC motor which is a nonlinear electrical system [YLX$^+$09], and complex dynamic systems with time delay such application in [CGH08, LSC08]. In these applications the HSIC proof superior performance with respect to classical model-based controllers. The studies have not investigated the effect of parameters variations on the controller performance. Also, there is no information provided about the computational load required with respect to classical controller.

The Brain Emotional Learning Based Intelligent Controller (BELBIC) is introduced in [LSS04] as intelligent controller, the proposed controller is based on the computational model of the Amygdala and the Orbitofrontal cortex, which are the main parts of limbic system in the brain, proposed in [MB00]. The fast learning ability of the BELBIC makes it a powerful model free controller for many tasks. The BELBIC is employed as a model free controller approach; this approach is preferable when the controlled system has high complexity and various uncertainties. In doing so, the expenses of system identification are removed [AS06].

The BELBIC is applied to different systems illustrate its high capability in controlling complex system using the model free approach. The BELCIC has been applied to stabilize the inverted pendulum on a cart in [JRAL09]; where it shows a superior performance in comparison to a well tuned PID-controller design. In [DM14] the BELBIC approach has been implemented on a brushless DC motor to control its speed. The results in [DM14] show both accurate steady state and fast transient speed responses are possible to achieve on a wide range of speed. The BELBIC controller has been applied to control the speed of permanent magnetic stepper motor in [OMTT05], the results show that the BELBIC has a fast response, high accuracy. The BELBIC approach is very reliable in comparison to classical PID controller tuned for the processes. As in the HSIC studies, The effect of system parameters variations on the controller performance is not investigated.

1.2 Soft computing-based controller

Based on [RN15, KK98] a soft computing-based controller can be defined as an intelligent controller using machine learning algorithms, which through the interaction with the targeted dynamic system, is able to create a model of the dynamic system. Using this model the soft computing-based controller is able to generate an appropriate control strategy in order to achieve the desired goals provided. Additionally a soft computing-based controller should be able to detect changes dynamic system and adapt itself online and in real time in order to insure the close loop system stability. Moreover, a soft computing-based controller should be capable of analyzing and maintain an optimal performance.

1.3 Motivation

The modern machine has increase demand in high performance, and efficient power consumption. Due to these requirements, the modern machine is build using light weight material, and smaller actuators. This leads to increase of flexibly of the structure of the machine and the overall machine nonlinearities. Traditionally, a classical or modern controller is used this machine and relay on the robustness of the controller to maintain the machine high performance. Unfortunately this is not always the case. Since the controller design depend on model which is not always accurate due to the high complexity of the machine. Therefore there is increase need to implement new methods that provide suitable performance based on the data.

The main goal of this work is to achieve a high performance controller using the learning ability of the soft computing algorithms which allows the soft computing-based controller to adaptive and improve its performance based on the measured data. The soft computing-based controller is able to make multi-step ahead predictions this allows the controller to calculate the most efficient control signal to achieve the desired reference goal.

1.4 Organization of the thesis

Parts of this thesis are submitted as journal paper [TS16b, TS16c, TS16a] or presented in international conferences [TS14, TS15a, TS15b].The remaining parts of this thesis are consists of the following chapters:

- In chapter 2, a review of previous works in modeling and controlling of dynamic system using soft computing algorithms is presented [TS16c, TS16a].
- In chapter 3, an evaluation of the different soft computing algorithms in modeling and controlling is performed [TS16c, TS16a].
- In chapter 4, the soft computing-based controller concept is introduced [TS14].
- In chapter 5 the validation results are also provided [TS15a, TS16b].
- In chapter 6, the verification results of the soft computing-based controller are presented[TS15b, TS16b].
- In chapter 7, a summary of the thesis and future outlook are given.

2 Literature review

In this chapter the modeling and inverse model control using soft computing algorithms are briefly discussed. The soft computing algorithms in focus of this chapter are Multi-Layer Perceptron (MLP) network, Support Vector Machine (SVM), Gaussian process, Radial Basis Function (RBF) and Fuzzy Inference System (FIS). In the first section a brief description of the mathematical modeling of dynamic system using soft computing algorithms. The most recent applications and improvements applied to model dynamic systems using soft computing algorithms also are described. In the second section the direct inverse model control using soft computing algorithms are introduced. Additionally, the most recent applications realized using soft computing algorithms are presented. Parts from this chapter are based on text and material that are previously published [TS16c, TS16a].

2.1 Modeling of dynamic system using soft computing algorithms

This section concerns with modeling of dynamic systems using five soft computing algorithms. These algorithms are Multi-Layer Perceptron (MLP) net, Support Vector Machine (SVM), Gaussian process, Radial Basis Function (RBF), and Fuzzy Inference System (FIS). Several goals for modeling a dynamic system are known, such as prediction of the system behavior considering different operation conditions, fault detection, and controller design. Modeling of the dynamic system based on measured input-output data, with no or less restrictive knowledge requirements about the physical effects governing the system, is called black-box modeling. It is important and useful to use existing fundamental knowledge about the systems behavior. The fundamental knowledge about the system can include system order, the system dynamics like acceptable sampling frequency, system stability, operating range, time delay presence, and characteristics of the system nonlinearities.

To design suitable controllers, the used dynamic model of the desired system should be able to produce accurate multi-step ahead prediction. The mathematical description of this dynamic model is given as

$$\hat{y}(k) = f\Big(u(k), u(k-1), u(k-2), ..., \\ u(k-p), \hat{y}(k-1), \hat{y}(k-2), ..., \hat{y}(k-p)\Big), \quad (2.1)$$

where f is the desired function, $u \in \mathbb{R}^m$ denotes the system input vector, $\hat{y} \in \mathbb{R}^r$ the estimated system output vector, k the time step, p the time step delay, m the number of inputs, and r the number of outputs. For simplicity, let

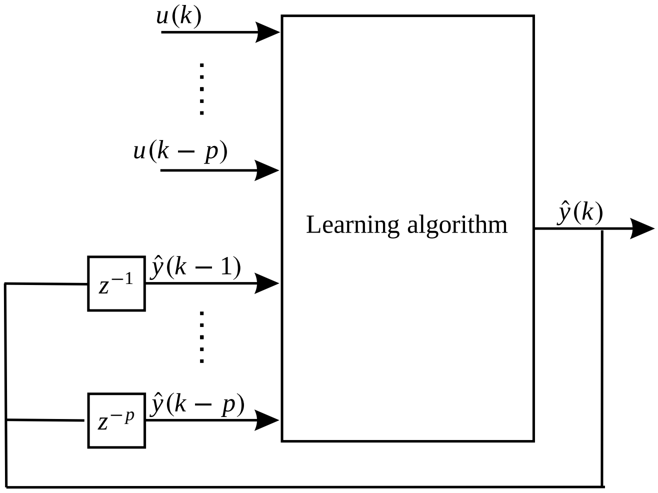

Figure 2.1: Output error model structure [TS16c]

$$q(k) = \Big[u(k), u(k-1), u(k-2), ..., \\ u(k-p), \hat{y}(k-1), \hat{y}(k-2), ..., \hat{y}(k-p)\Big]. \quad (2.2)$$

Depending on the field of study, the mathematical description given in equation 2.1 is known as recurrent input/output model [Hay98, SVdM96, Zur92], nonlinear output error (NOE) model [SZL+95], or parallel modes [NP90]. As noted in [NRPH00], modeling of dynamic system using NOE structure is achievable when the measurement noise is white noise, else the nonlinear autoregressive exogenous (NARX) model is applied. The NARX model is valid for one step ahead prediction [NRPH00].

2.1.1 Soft computing algorithms applied to model dynamic systems

In this section five major soft computing algorithms used in dynamic system identification are presented. In this section mathematical description of each soft computing algorithm is given. Then the most recent developments made in system identification.

2.1.2 Dynamic system modeling using multi-layer perceptron

Recurrent Multi-Layer Percepton (MLP) network is the most popular algorithm of modeling dynamic systems using soft computing algorithm. Recurrent MLP is also known as recurrent neural network or dynamic neural network. In figure 2.2 an example Recurrent MLP that consists of one hidden layer and one output layer is shown. The hidden layer uses J neurons while the output layer uses I neurons. The mathematical description of the recurrent MLP shown in figure 2.2 is given as

$$\hat{y}(k) = \sum_{i=1}^{I} w_i \sum_{j=1}^{J} \Big(\sigma_{ij}(v_{ij}^T q(k) + b_{ij}) \Big) + b_i, \tag{2.3}$$

where
$q(k)$ denotes the input vector,
j is the index of the neuron in the hidden layer,
i is the index of the neuron in the output layer,
J is the total number of neurons in the hidden layer,
I is the total number of neurons in the output layer,
v denotes the hidden layer weights vector,
b denotes the bias terms,
w denotes the output layer weights vector, and
σ denotes the activation function.

Different activation functions are used, the most common one is the sigmoid function

$$\sigma(v_{ij}^T q(k) + b_{ij}) = \frac{1}{1 + e^{-(v_{ij}^T q(k) + b_{ij})}}. \tag{2.4}$$

The use of the MLP network to identify nonlinear dynamic systems has been proposed in [LF88, CBG90, NP90]. The recurrent MLP models are able to capture various plant nonlinearities as demonstrated in [LNG95, SM97]. In [GHJ03] it is shown that MLP is an effective soft computing algorithm for modeling and controlling a broad category of complex nonlinear systems. Also, the recurrent MLP has the advantage of providing long range predictions even in the presence of measurement noise as noted in [SM97].

The recurrent MLP network architecture design is important due to its impact on the network's information processing [XZZW09]. According to [XZZW09], successful modeling of nonlinear dynamic systems using recurrent MLP depends on three factors, namely the network architecture, input time delays, and the network weights. The majority of the research focuses on tuning weights of MLP to obtain a model of the nonlinear dynamic systems. On the other hand the network architecture is designed with human interventions via a tedious trial-and-error process [XZZW09]. Research on automatic design of MLP network architecture in carried out by many works such as [Che90, Fre90, YKS91, ALS93, Wei94]. According to [XZZW09] obtaining a good architecture systematically and autonomously is a challenging problem.

According to [Pea95, KMD$^+$96, PB03], the network weights are obtained through the training process which is a major challenge. Different suggestions in literature are made to overcome this issue such as backpropagation method [RHW86], conjugate gradient method [LK90], Levenberg-Marquardt optimization [Mar63], or methods which uses genetic algorithm (GA) [Gol89].

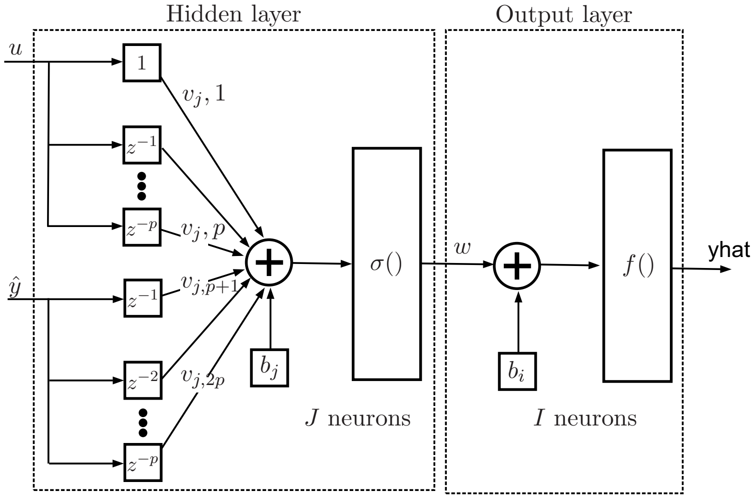

Figure 2.2: Recurrent Multi-Layer Percepton Network [TS16c]

Recent development on MLP

In the following the most recent applications using the Multi-Layer Perceptron network are briefly introduced:

- In [LPCL15] a Recurrent Wavelet Neural Network (RWNN) model is proposed to identify nonlinear dynamic systems. The proposed RWNN model consists of a four-layer structure. The temporal relation is generated in the network by adding feedback connection in the second layer. Additionally an online learning algorithm is proposed consisting of structure and parameter learning. The structure learning is based on input partitions to determine the number of wavelet bases. Parameter learning uses the supervised gradient descent method to adjust the shape of wavelet functions, feedback weights, and the connection weights. The approach is applied to model a chaotic Henon system. The results are compared with memory neural network and Elman recurrent neural network. It was observed that the accuracy of RWNN is higher in comparison to the Elman and memory neural network.

- The Context of Layered Locally Recurrent Neural Networks (CLLRNN) for system identification purpose is proposed in [Cob13]. The CLLRNN consists of one input layer, one or more hidden layers, one output layer, and one context layer. The context layer is introduced to improve the ability of the network to capture linear characteristics of the system. Dynamic memory is provided by means of feedback connections from nodes in the first hidden layer to nodes in the context layer. Self-recurrent connections in all nodes of the context and hidden layers are used. The approach is applied to model linear, nonlinear, and MIMO dynamic systems and compared with Elman type Recurrent Neural Network (ERNN). The results show improvement over the ERNN results.

- In [AF11] a new architecture called Sigmoid Diagonal Recurrent Neural Network (SDRNN) is proposed. This architecture is achieved by adding a sigmoid weight vector in the hidden layer neurons to adapt the shape of sigmoid function. The logarithm is applied successfully to identify a second order nonlinear and rigid non-minimum phase systems. The simulation results showed that the SDRNN is more efficient and accurate than the DRNN.

- In [TR10] the identification of nonlinear dynamic system using recurrent neural network with multi-segment piecewise-linear connection weight is proposed. The traditional recurrent neural networks are based on connection weights which are a single real number, but in [TR10] the weight of each connection is multivalued, depending on the value of the input data involved. This approach is tested using twelve different chaotic systems and compared with traditional recurrent neural network. The simulation results showed improvements in the accuracy of model predictions.

- The Pipelined Functional Link Artificial Recurrent Neural Network (PFLARNN) is proposed for nonlinear dynamic system identification [ZZ09]. The proposed PFLARNN consists of a number of simple small-scale functional link artificial recurrent neural network (FLARNN) modules. The approach is applied to four different nonlinear second order dynamic systems. The results are compared with functional link artificial recurrent neural network and multilayer perceptron multilayer perceptron neural network. The comparison results show that the FLARNN has lower normalized mean squared error so it has higher accuracy over the other two algorithms.

Modeling of dynamic system using MLP algorithm is one of the most popular applications the method is used for. In this section a short description of the modeling using MLP algorithm is provided, additionally the structure and resulting mathematical model are explained. The recent developments of the MLP algorithm are summarized. In section 3.2, the MLP algorithm is compared with other soft computing algorithms used to model dynamic systems.

2.1.3 Dynamic systems modeling using radial basis function

Radial Basis Function (RBF) networks are considered as an alternative to MLP. A radial basis function network is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.

Like neural networks and fuzzy inference systems, radial basis function networks are also proven to be a universal approximator [PS91]. According to [YXPW11], RBF networks are easy to design, show good generalization, strong tolerance to input noise, and online learning abilities.

The dynamic model given using radial basis function as

$$\hat{y}(k) = \sum_{i=1}^{I} w_i \sum_{j=1}^{J} \Big(\rho_{ij}(||v_{ij}^T q(k) + b_{ij} - c_{ij}||) \Big) + b_i, \tag{2.5}$$

where
j is the index of the neuron in the hidden layer,
i is the index of the neuron in the output layer,
J is the total number of neurons in the hidden layer,
I is the total number of neurons in the output layer,
f is the output activation function,
v is the weight of neuron in the hidden layer,
w is the weight of neuron in the output layer,
b is a bias term,
c is the center vector for neuron, and
$\rho(|| \quad ||)$ is a Gaussian function.

The parameters v, w, b, and c are obtained after the training. According to [SKP01] the training of RBF network is carried on in two steps. In the first step the parameters (v, w, b, and c) are initialized in the next step adaption process is carried on. The center vector c is determined using K-means clustering. A fine tunning of the network can be obtained using a third step by applying backpropagation method.

Recent implementation using RBF

In the following the most recent applications are briefly introduced . The RBF networks are used in modeling dynamic systems:

- In [DLHJ$^+$13] A-optimality design criterion is integrated into the two stage selection approach presented by [LPB06]. The computational efficiency of the approach presented in [LPB06] is kept by introducing a residual matrix which is used to recursively update the variance of estimated parameters. The

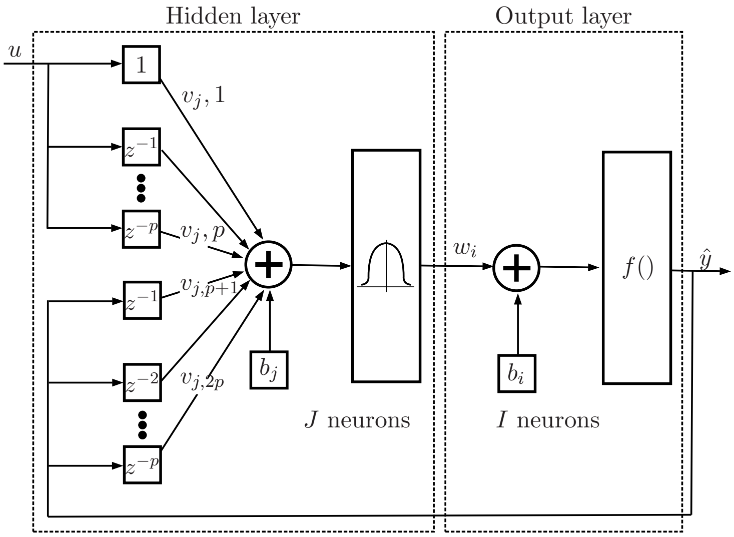

Figure 2.3: Recurrent Radial Basis Function Network [TS16c]

approach presented in [DLHJ$^+$13] is applied to model the Van der Pol system. The simulation results are compared with original two-stage selection proposed in [LPB06] and the orthogonal least squares method. The comparison results illustrate the effectiveness of this approach.

- In [XLX12], a system identification approach that combines a modified differential evolution algorithm and RBF network is proposed. In [XLX12], it is suggested to modify the values of differential evolution algorithm during the run of training to improve the differential evolution algorithm and avoid being trapped in local optima. This approach configures the structure of the RBF network automatically and obtains the model parameters simultaneously based on the input-output data without a priori knowledge. The approach is applied to model the Hermite polynomial function. The approach demonstrated good effectiveness.

- In [HCQ10], Online Self-Organizing Radial Basis Function (SO-RBF) network is proposed. The goal of this algorithm is to reduce the redundant RBF nodes in the hidden layer. The structure of the RBF network is designed using growing and pruning method. The growing and pruning method is based on the radius of receptive field of the RBF nodes and stable error of the requirements. The gradient-descent algorithm is used to adjust the position and width of the RBF nodes in the hidden layer. The algorithm was compared to two algorithms which are dynamic fuzzy neural network and growing-and-pruning-RBF. The experimental results of modeling wastewater treatment system show that the proposed method is efficient for network structure optimization. It achieves better performance than the models obtained using dynamic fuzzy neural network and growing-and-pruning-RBF algorithms.

- In [CHLH09], the dynamic model tuned using Particle Swarm Optimization (PSO) algorithm is proposed. The PSO algorithm is adopted to tune one RBF unit's center vector and diagonal covariance matrix by minimizing the leave-one-out mean square error. The PSO algorithm assisted the orthogonal forward regression in determining the sufficient number of tunable RBF nodes required for modeling. The approach is applied to model three different systems based on experimental data. The results are compared with local regularization assisted orthogonal least squares algorithm. The compression results demonstrate the effectiveness of this approach. It is noted that models obtained using the PSO algorithm are simpler than models generated using the local regularization assisted orthogonal least squares algorithm.
- In [LLSŻ08] a self-organizing Radial Basis Function network that enables faster training than traditional RBF, is presented. The self-organizing radial basis function network is proposed as real-time approximator for continuous-time dynamical systems with many inputs. The self-organizing radial basis function network adapts its structure dynamically to maintain the accuracy of the prediction. Performance of the self-organizing radial basis function network is evaluated by modeling four different dynamic systems based on simulation results, the dynamics systems are a third order linear unstable dynamic system, Van der Pol Oscillator, Lorentz attractor, and a three-phase singly fed induction machine. The self-organizing radial basis function network showed very high approximation accuracy.

The use of RBF network to model dynamic system can be considered as the second most popular algorithm after MLP algorithm. A brief explanation of the RBF algorithm is given in this section, followed by the structure and mathematical function of the model. A summary of the most recent development of the algorithm in order to increase the accuracy of the modeling is given. A comparison of the RBF algorithm with other soft computing is carried out in section 3.2.

2.1.4 Dynamic system modeling using support vector machine

The Support Vector Machine (SVM) algorithm is proposed in [VGS96, Vap98a, Vap98b, Vap99]. In the field of classification, the SVM algorithm is widely used. On the other hand the use of the SVM algorithm in the field of system identification is not fully explored. In [SV02] the SVM algorithm is adapted to model an unknown dynamic system measured by input-output relations. The algorithm is known as Least-Squares Support Vector Machine (LS-SVM). As an example given in [SV02], the double scroll system is successfully modeled using LS-SVM.

Support Vector Machine is a supervised learning method, which requires training data set. This data set consists of N samples of the system inputs u and outputs y. The data set is given as $D = \{u(k), y(k)\}_{k=1}^{N}$. The modeling process of an unknown dynamic system with the input vector input vector $u \in \mathbb{R}^m$ and output vector $y \in \mathbb{R}^r$ is done by estimating the function $f()$ given as

$$\hat{y}(k) = f\Big(\hat{y}(k-1), \hat{y}(k-2), ..., \hat{y}(k-p), \\ u(k), u(k-1), ..., u(k-p)\Big). \quad (2.6)$$

As noted in [SV99], the nonlinear dynamic model is given as

$$\hat{y}(k) = \sum_{j=1}^{M} \alpha_j K(q(k), q_j) + b, \quad (2.7)$$

where
α_k is the Lagrange multiplier,
$K(,)$ is the Kernel function,
q_j is a support vector,
b is a bias term, and
M is the number of support vectors.

It is noted in [HCL10] that if the number of features is large, there is no need to map the data into higher dimensional space. So, the nonlinear mapping does not improve the performance. Therefore in this study a linear kernel function is presented. The dynamic model is given as

$$\hat{y}(k) = \sum_{j=1}^{M} \alpha_j < q(k), q_j > +b, \quad (2.8)$$

where $<,>$ represent the dot product.

As given in [SV02], the Lagrange multiplier α_k, the bias term b are calculated by solving the following equation

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & Q^T Q + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad (2.9)$$

where
$Q = [q_1, ..., q_{N-p}]$,
$y = [y_1, ..., y_{N-p}]$,
$\vec{1} = [1_1, ..., 1_{N-p}]$,
$\alpha = [\alpha_1, ..., \alpha_{N-p}]$ and
γ is a positive real constant.

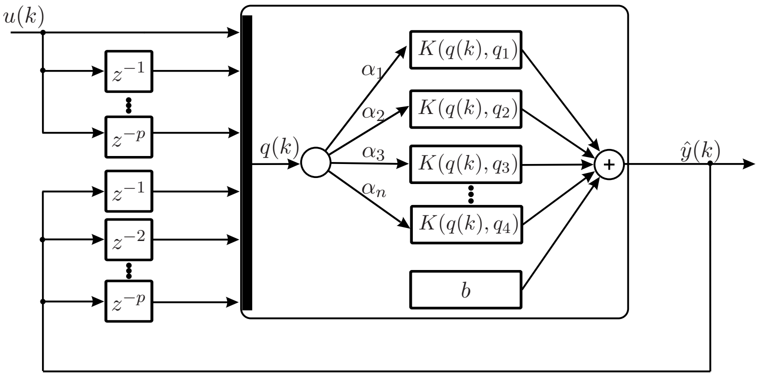

Figure 2.4: Recurrent LS-SVM model [TS15a]

The graphical representation of the recurrent LS-SVM model is shown in Fig. 2.4, [TS15a]. In the recurrent LS-SVM model to achieve multi-step ahead prediction the output of model is fed back to the model input. In [YLJH03], it is concluded that the SVM needs less tuning effort in comparison to MLP network. Also, it is noted that the SVM algorithm requires less computational time than MLP. This is due to the fact that the SVM algorithm uses linear equations to obtain the results while MLP network uses backpropagation algorithm. According to [YLJH03] the models obtained using the SVM algorithm have better generalization ability than models obtained using MLP network. Additionally, for the SVM no risk of getting stuck into local minima exists, thus the SVM always results in a unique and global solution [YLJH03].

Recent development on SVM

In the following the most recent applications using the Support Vector machine are briefly introduced:

- In [HW14] Weighted Support Vector Echo State Machine (WSVESM) is proposed for the modeling of multivariate dynamic systems. In this approach the observed data are treated as multivariate time series. Different weights are assigned to the time series data, then the weights are optimized using a suitable solution path. The approach is used to model the Lorenz system, the results are compared with echo state network, support vector echo-state machine, and Elman network. As a conclusion, it was reported that the WSVESM appears more robust with respect to modeling capability than the echo state network, the extreme learning machine, and the support vector echo state machine algorithms.

- In [LX12] the integration of Support Vector Regression (SVR) with fuzzy rules is proposed. This integration improves the performance of SVM. A new class of kernels which are based on fuzzy theory is designed. This allows the incorporation of prior information in the form of fuzzy rules into the training process of support vector regressions. The approach is applied to identify three systems. The first one is a simulation example presented in [Chu07]. The second and third examples are based on experimental data. The second model represents a gas oven system and the third is a prediction model of forest-fire. It has been concluded that the proposed approach achieves the desired accuracy.

- In [WM11] the use of wavelet function as kernel for the SVM is proposed. Marr wavelet is found to satisfy the admissible condition. The approach is applied to model rotation speed of a helicopter based on experimental data. The results are compared with MLP model, the comparison results showed that the wavelet-based SVM model has a simpler structure, the training converges faster, the wavelet-based SVM model has higher generalization abilities than MLP.

- In [LSB09] a combination of Wavelet Kernel Function with the Linear Programming Support Regression (LP-SVR) is done. The linear programming support regression with wavelet kernel is applied to model a hydraulic robot arm based on experimental data. The results are compared with Quadratic Programing-SVR (QP-SVR), although the accuracy is less than the QP-SVR, the LP-SVR with wavelet kernel model has used less support vectors than QP-SVR. From the results it can be concluded that the LP-SVR model is more compact than the QP-SVR model.

- In [ZLZ08] the Particle Swarm Optimization (PSO) algorithm is combined with Sequential Minimal Optimization (SMO) and LS-SVM to optimize the parameters of SMO and LS-SVM. The approach is applied to model Mackey-Glass Series, which is a chaotic system. The results are compared with self-organizing fuzzy neural network, SMO and LS-SVM algorithms. It has been concluded that the model based on LS-SVM with PSO has more accuracy than the other models obtained using self-organizing fuzzy neural network, SMO and LS-SVM algorithms.

To summarize this section, the modeling of dynamic system using SVM algorithms is briefly explained. First the mathematical function is given and then the structure of dynamic system models using SVM algorithm is provided. A short summary of the most recent development introduced to the SVM algorithm is discussed. A comparison of the SVM algorithm with other soft computing algorithms used in modeling dynamic system is given in section 3.2.

2.1.5 Dynamic systems modeling using Gaussian process

Modeling using Gaussian Process (GP) is used as an example for non-parametric modeling based on information about the prediction underestimates [HK78], [KGBMS05]. In [Ras96] the relationship between MLP network and GP is shown. The Gaussian process is a MLP network with an infinite number of neurons in hidden layers.

Modeling examples using GP are given in [MSJS99, LMSL00, GL02, SMSL$^+$03, GL02]. In [MSJS99] the GP technique is applied to model vehicle dynamics. In [LMSL00] the GP is used to identify nonlinear dynamic structure. Modeling of nonlinear hydraulic system is presented in [GL02]. In [SMSL$^+$03] it is used to model nonlinear dynamic systems.

Using GP, a dynamic system is described as

$$\hat{y}(k)_{|q(k)} \sim \mathcal{N}\Big(\mu\big(q(k)\big), var(q(k))\Big), \tag{2.10}$$

where the mean $\mu\big(q(k)\big)$ is

$$\mu\big(k(q)\big) = \boldsymbol{k}(q(k))^T \boldsymbol{K}^{-1} q(k),$$

the variance $var(q(k))$ is

$$var\big(q(k)\big) = cov(q(k), q(k)) - \boldsymbol{k}(q(k))^T \boldsymbol{K}^{-1} \boldsymbol{k}(q(k)),$$

the covariance vector $\boldsymbol{k}(q(k))$ is given as

$$\boldsymbol{k}(q(k))^T = [cov(q(k), q_1), ..., cov(q(k), q_{N-p})], \text{ and}$$

the covariance matrix $\boldsymbol{K}$ is given as

$$\boldsymbol{K} = \begin{bmatrix} cov(q_1, q_1) & cov(q_1, q_2) & \dots & cov(q_1, q_{N-p}) \\ cov(q_2, q_1) & cov(q_2, q_2) & \dots & cov(q_2, q_{N-p}) \\ \vdots & \vdots & \vdots & \vdots \\ cov(q_{N-p}, q_1) & cov(q_{N-p}, q_2) & \dots & cov(q_{N-n}, q_{N-p}) \end{bmatrix},$$

with $cov()$ as covariance function. A graphical representation of the modeling dynamic system using GP is shown in figure 2.5.

Several covariance functions are available for the use such as linear [Ras04], Gaussian noise, squared exponential, and others. According to [Wil98] in practice it is unlikely to know which covariance function to use. As a solution a parametric family of covariance functions with parameter vector θ is chosen, so the parameter vector θ is

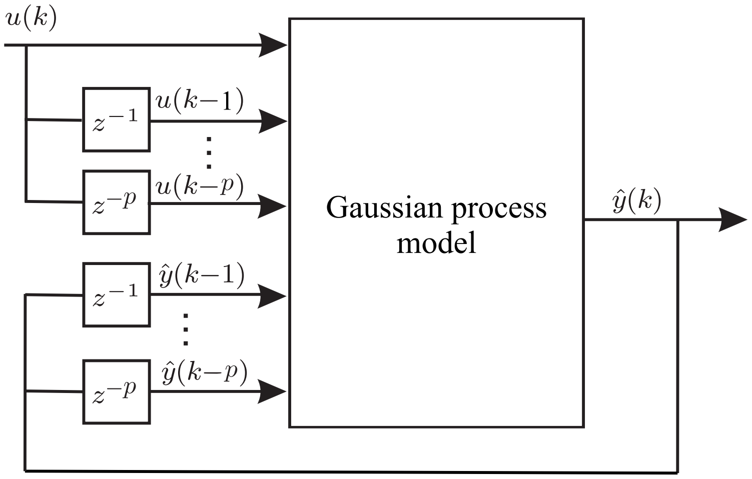

Figure 2.5: Recurrent Gaussian Process model [TS16c]

estimated using the method of maximum likelihood or using a Bayesian approach. In this case a posterior distribution over the parameters is obtained.

As it can be seen in equation 2.10 the outputs of a GP model are the mean and the variance. Using mean and variance the confidence interval of the estimation can be calculated. Based on the confidence interval the quality of prediction is evaluated. The measure is available only for GP models. To illustrate this measure two GP models are trained. The first is a model with good estimation ability while the second is a poor model. The results of the estimation of the two models are shown in figure 2.6. In figure 2.6 the output of the real system (blue) and the estimation using the GP model (green) are given. The confidence interval of the estimation (gray) made by the GP model are to be observed. In figure 2.6.a, it can be observed that the two curves are overlapping each other and also that the confidence interval is a very tight interval. In figure 2.6.b it can be observed that estimation is not suitable and the curves are deviating from each other. Also, it can be noted that the confidence interval is wider than the one shown in figure 2.6.a.

Recent Implementation using GP

In the following the most recent applications using GP are briefly introduced:

- In [AK11] the Gaussian process modeling in combination with a local model-based network approach is implemented. A methodology of the GP model identification including the prior knowledge based on linear local models is introduced. The approach from [AK11] is implemented using two examples; the first one is reduced to modeling of nonlinear second-order system based on simulation data. The second example is related to the identification of a two

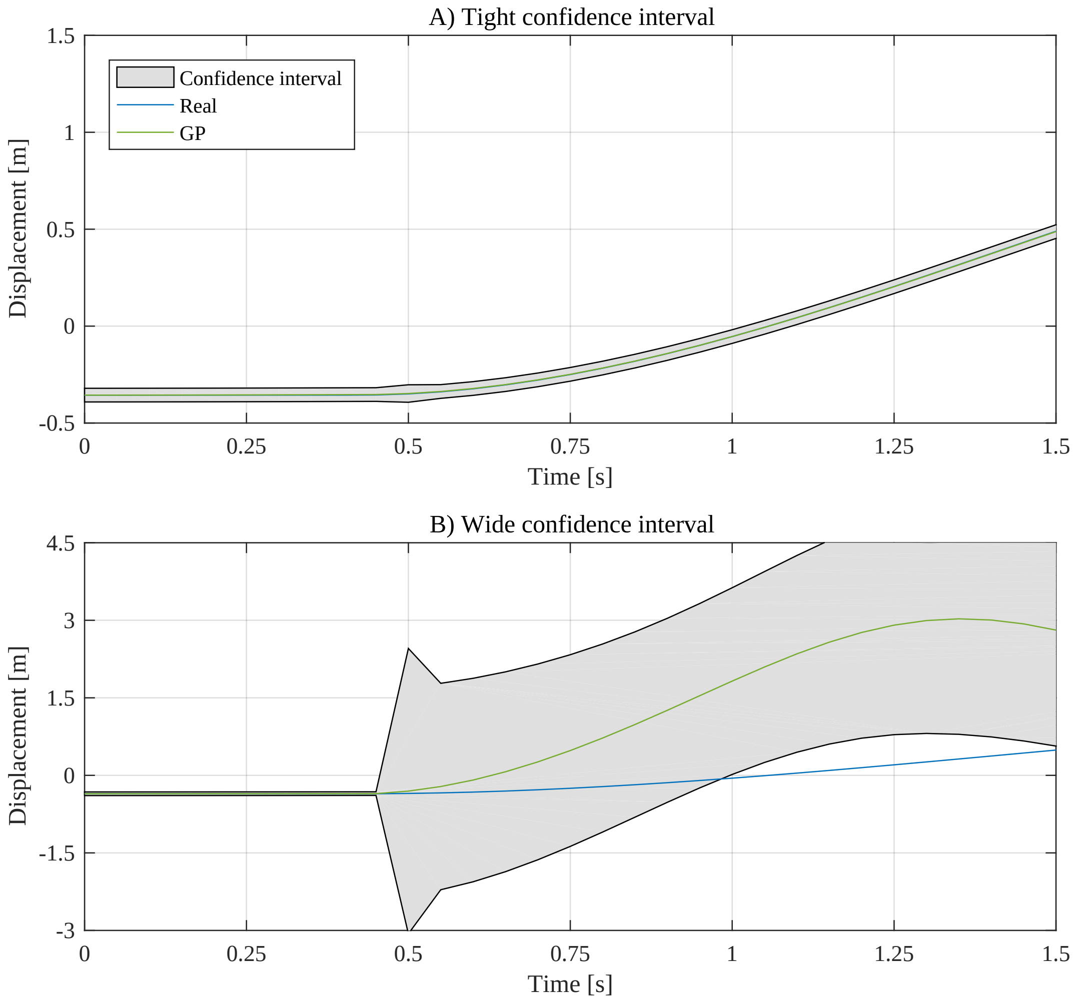

Figure 2.6: Estimation using GP-based model with confidence intervals [TS16c]

tank system based on experimental data. For both cases the approach shows acceptable accuracy.

- In [PŠ14] the recursive system identification using GP modeling is implemented. The goal is to reduce the computational load of the original GP model calculation referred to as Full Gaussian Process (FGP). The Recursive Gaussian Process regression algorithm (RGP) is proposed in [Hub13]. It is used to model a nonlinear dynamic system based on simulation data compared to FGP. It is demonstrated that RGP achieves a comparable accuracy to FGP but using less computational demands. It has been concluded that the RGP has less generalization performance than FGP.

- In [HS14], a Multi-model Predictive Control (MMPC) based on GP is proposed. Many dynamic systems operate in several different working points, which means that the identification process results in different models. A switching strategy that identifies the change in the working point accurately and

quickly is proposed. Based on the deviation between the actual system output and the predictive output the switching strategy is realized in [HS14]. The approach is implemented using a second order nonlinear system with four different working points. The simulation results are compared with Support Vector Machine model results. The comparison shows that the GP models are more accurate than those trained using SVM algorithm.

- In [CLC13] the identification of nonlinear dynamic system using Selective Recursive-GP (SR-GP) is proposed. To overcome the computational load of the traditional GP, a method that recursively updating the covariance matrix is implemented. The update scheme is realized by selectively admitting data to regions requiring enhancement. Additionally a learning process is improved by removing redundant data from the model. The SR-GP is applied to an industrial boiler based on experimental data. The prediction results show the effectiveness of the approach. It has been concluded that the approach is online feasible.

Modeling of dynamic system using GP algorithm is the most recent method of soft computing algorithms. In this section a brief explanation of the modeling of dynamic system using GP algorithms is given. First the dynamic system model based on GP is provided then structure of dynamic system is illustrated. A comparison of the GP algorithm with other soft computing algorithms used in modeling dynamic system is provided in section 3.2.

2.1.6 Dynamic systems modeling using fuzzy inference system

A Fuzzy Inference System (FIS) is a system that uses Fuzzy Set Theory to map inputs to outputs. In [TS85, SY93] the effectiveness of modeling of nonlinear systems using such a method is demonstrated. Fuzzy Inference System based only on expert knowledge lacks the required accuracy. To overcome this the neuro-fuzzy modeling approach is introduced in [Jan93, CFCT98, JS01]. The neuro-fuzzy inference system acquires knowledge from a set of input-output data. In figure 2.7 the general structure of fuzzy inference system is shown.

The dynamic model's ouput $\hat{y}(k)$ is the output of the overall output layer (O_1^5) of the FIS as

$$O_1^5 = \sum_i \bar{w} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, \tag{2.11}$$

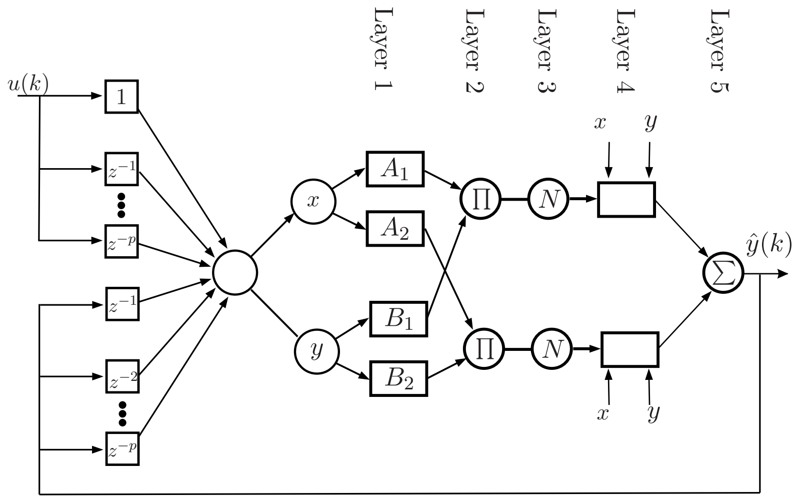

Figure 2.7: Recurrent ANFIS model[TS16c]

with f_i denotes a fuzzy if-then rule, $\bar{w}_i$ i^{th} rules firing strength and w_i the second layer's output. The forth layer output (O_i^4) is given as

$$O_i^4 = \bar{w} f_i = \bar{w}(p_i x + q_i y + r_i). \tag{2.12}$$

The third layer output(O_i^3) is the ratio of ith rule's firing strength to the sum of all rules' firing strengths given as

$$O_i^3 = \bar{w} = \frac{w_i}{w_1 + w_2}. \tag{2.13}$$

The second layer output (O_i^2), is the product of the signals coming from the first layer, is given as

$$O_i^2 = w_i = \mu_{A_i(x)} \times \mu_{B_i(x)}, \tag{2.14}$$

where $i = 1, 2$.
The first layer output (O_i^1) is give as following

$$O_i^1 = \mu_{A_i}(x), \tag{2.15}$$

where x is the input vector to node i, A_i and B_i are the linguistic label associated with this node function. The membership function is typical bell-shaped function. This function is given as

$$\mu_{A_i(x)} = \frac{1}{1 + \left(\frac{x-c_i}{a_i}\right)^2 b_i}. \tag{2.16}$$

or

$$\mu_{A_i(x)} = e^{-\left(\dfrac{x - c_i}{a_i}\right)^2} \tag{2.17}$$

where a_i, b_i and c_i are training parameters.

The training of the ANFIS is achieved using a hybrid learning algorithm as presented in [Jan93]. In the ANFIS algorithm the least square method is used in the forward path to identify the consequent parameters in the fourth layer. In the backward pass the errors are back propagated and the parameters are updated using gradient descent algorithm.

Recent implementation using ANFIS

In the following the most recent applications using the Fuzzy Inference System are briefly introduced:

- In [SS15] the Probabilistic Fuzzy Inference System is proposed. The goal of this approach is to minimize the effects of uncertainties modeling the dynamic system. This is achieved by combining probability theory and fuzzy theory. The probabilistic fuzzy inference system is divided into three steps which are fuzzification, inference, and output processing. The output processing step consists of order reduction and defuzzification. To demonstrate the effectiveness of the proposed approach in [SS15], the approach is applied to model a continuous stirred tack reactor based on simulation data. After that, the obtained results are compared to conventional fuzzy inference system. It has been found that the probabilistic fuzzy inference system has improved performance over the conventional fuzzy inference system.

- In [YO14] the Dynamic Adaptive Neuro-FuzzyInference system (DANFIS) for nonlinear dynamical system modeling is introduced. The if-parts of the rules in the DANFIS consist of Gaussian type membership function. The then-parts of the rules consists of differential equations of linear functions. The model parameters are found using a gradient-descent-based algorithm Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. The approach is implemented to model the Van der Pol oscillator and the tunnel diode circuit. The results showed very accurate estimation of the target systems.

- In [YP10] the unsupervised ANFIS for solving differential equations is proposed. The proposed approach consists of two parts; the first part satisfies the initial boundary conditions and the second part is an ANFIS. The first part has no adjustable parameter. The second part has no influence on the initial or boundary conditions. The algorithm is applied to model five dynamic systems.

> The dynamic systems are of first and second order. The simulation results are compared with conventional ANFIS system. The unsupervised ANFIS models showed a small deviation from the ANFIS models.

In this section modeling of dynamic system using ANFIS algorithm is introduced. In addition to the structure of the model, the mathematical function is also introduced. Also, a short survey of the most recent developments to the ANFIS algorithm is provided. An evaluation of ANFIS algorithm with respect to other soft computing algorithms used in modeling dynamic system is provided in section 3.2.

2.2 Control of dynamic system using soft computing algorithms

Soft computing algorithms provide a very powerful tool in learning dynamic systems behavior. Recently, soft computing algorithms are been used as direct inverse controller for complex nonlinear dynamic system. A direct inverse model controller is a controller with inverse characteristic of the dynamic system to be controlled. It is provided in [LJ02], that the direct inverse control has large potential in controlling nonlinear system.

Assuming that the unknown dynamic system can be described as

$$\begin{aligned} y(k) = f\Big(&u(k), u(k-1), u(k-2), ..., \\ &u(k-p), y(k-1), y(k-2), ..., y(k-p)\Big), \end{aligned} \tag{2.18}$$

where f is the system dynamic function, $u \in \mathbb{R}^m$ is the system input vector, $y \in \mathbb{R}^r$ is the system output vector, k is the time step, p is time step delay, m is number of inputs, and r is the number of outputs.

The desired inverse controller is obtained by training an approximation of the inverse model of the unknown dynamic system given in equation 2.18 as follows

$$\begin{aligned} \hat{u}(k) = f^{-1}\Big(&\hat{u}(k-1), \hat{u}(k-2), ..., \\ &\hat{u}(k-p), y(k), y(k-1), ..., y(k-p)\Big). \end{aligned} \tag{2.19}$$

For simplicity, let

$$\begin{aligned} q(k) = \Big[&\hat{u}(k-1), \hat{u}(k-2), ..., \\ &\hat{u}(k-p), y(k), y(k-1), ..., y(k-p)\Big]. \end{aligned} \tag{2.20}$$

According to [NRPH00], once the soft computing algorithm finishes the training an inverse model is build. The resulting model can be used to control the dynamic system by substituting the output at time t by the reference signal r if the model has an exact representation of the inverse characteristics of the dynamic system, the inverse model output can drive the dynamic system to the desired r. The direct inverse model controller concept is illustrated in figure 2.8.

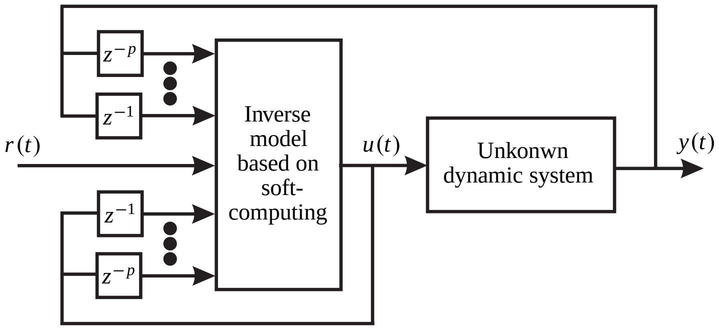

Figure 2.8: Inverse model control structure [TS16a]

2.2.1 Direct inverse model control based on soft computing algorithms

In this section four major soft computing algorithms used in obtaining direct inverse model controller are presented. In this section the mathematical description of each soft computing algorithm is given. Then some of the recent applications.

2.2.2 Multi-layer perceptron-based direct inverse model control

Recurrent Multi-Layer Percepton (MLP) network is the most popular algorithm of modeling and control dynamic systems. Recurrent MLP is also known as recurrent neural network or dynamic neural network. In figure 2.9 the direct inverse model controller based on MLP used to control unknown dynamic system is given. The inverse model controller consists of one hidden layer and one output layer. The hidden layer uses J neurons while the output layer uses I neurons. The mathematical description of the inverse model control shown in figure 2.9 is given as

$$\hat{u}(k) = \sum_{i=1}^{I} w_i \sum_{j=1}^{J} \Big(\sigma_{ij}(v_{ij}^T q(k) + b_{ij})\Big) + b_i, \tag{2.21}$$

where
$q(k)$ denotes the input vector given in equation 2.20,

j the index of the neuron of hidden layer,
i the index of the neuron of output layer,
J the total number of neurons of hidden layer,
I the total number of neurons of output layer,
v the hidden layer weights vector,
b the bias terms,
w the output layer weights vector, and
σ denotes the activation function.

Different activation functions are used, the most common one is the sigmoid function given as

$$\sigma(v_{ij}^T q(k) + b_{ij}) = \frac{1}{1 + e^{-(v_{ij}^T q(k) + b_{ij})}}. \tag{2.22}$$

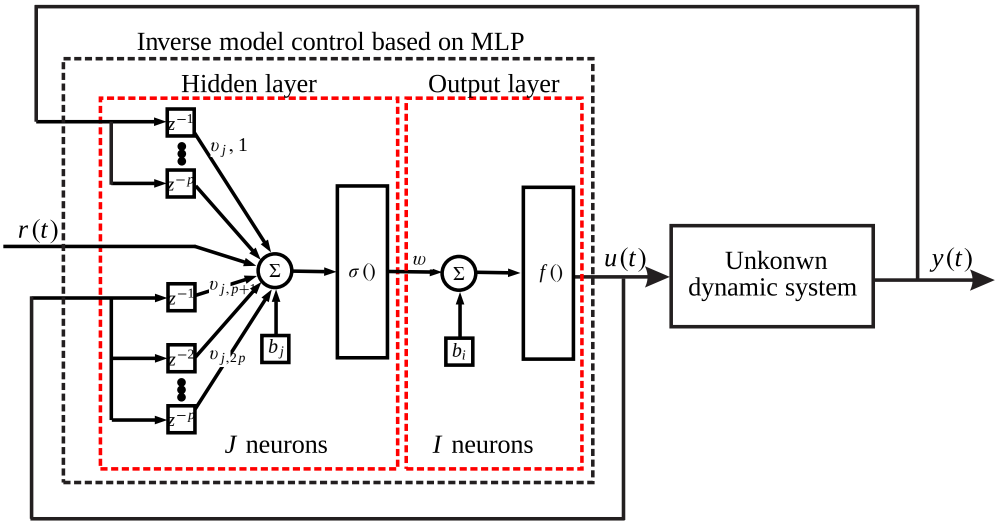

Figure 2.9: Inverse model control structure using MLP [TS16a]

The recurrent MLP models are able to capture various nonlinearities as demonstrated in [LNG95, SM97]. In [GHJ03] it is shown that MLP is an effective soft computing algorithm for modeling and control a broad category of complex nonlinear systems. Here the MLP is used to model discrete and contentious systems. most of the example are limited to first order system.

According to [XZZW09], successful modeling using MLP depends on three factors. These factors are network architecture, input time delays, and network weights. The MLP network architecture design is important due to its impact on the network's information processing [XZZW09]. The majority of the research focuses on tuning weights of MLP. On the other hand the network architecture is via a tedious trial-and-error process [XZZW09]. Some studies on the automatic design of MLP network architecture are carried on such as [Che90, Fre90, YKS91, ALS93, Wei94]. According to [XZZW09] systematically and autonomously obtaining a good architecture is a challenging problem.

According to [Pea95, KMD$^+$96, PB03] The network weights are obtained through the training process which is a major challenge. Different suggestions in literature are made to overcome this issue such as backpropagation method [RHW86], conjugate gradient method [LK90], Levenberg—Marquardt optimization [Mar63], or methods which are based on genetic algorithm (GA) [Gol89].

Recent applications of MLP-based inverse model control

In the following the most recent applications, using inverse model controlling nonlinear systems applying MLP, are briefly introduced:

A 3-DOF tandem helicopter is controlled using direct inverse model control based on wavelet MLP [MSP15]. Wavelet-MLP network has an architecture similar to that of traditional MLP network with the exception that node activation function is wavelet function (i.e. mother wavelet). The experimental results show that direct inverse model controller based on wavelet has better performance than inverse model controller based on traditional MLP.

An optimization of a direct inverse controller based on MLP for a small helicopter is proposed in [SWK15]. To improve the performance of the direct inverse controller an optimization scheme is proposed. The optimization is done by retraining the MLP using new data. Experimental results show that the optimized direct inverse model based on MLP has a better performance than traditional direct inverse model controller based on MLP.

In [RHJ16] a top and bottom temperature of debutanizer column (a type of fractional distillation column) control is performed using a combination of direct inverse model controller based on MLP and internal model control approaches. The debutanizer column system is hard to control due to its high nonlinear dynamical behavior and the existence of constraints in inputs and state variables. In [RHJ16] mixture of closed-loop and open-loop data are used to train the inverse model controller. The simulation results show that inverse model controller has higher accuracy and performance that a well-tuned PID controller.

In [RPMP15], the direct inverse model control is used to control the level of a conical shaped tank system. A conical tank system is widely used in industrial application due to the easy flow of liquid across its cross section area. The training of the controller is realized using Levenberg-Marquardt back-propagation algorithm. The simulation results show that the direct inverse model controller has a good dynamic behavior and accurate response, no comparison with other controller is done.

A direct inverse model controller based on improved Cerebellar Model Articulation Controller (CMAC) for Continuous Stirred-Rank Reactor CSTR is proposed in [GML10]. This approach introduces the Improved Credit Assigned CMAC ICA-CMAC with fast learning speed, which is helpful in real time control of CSTR. The simulation results show that ICA-CMAC is five times faster when compared with the conventional CMAC. Also, it is demonstrated that the ICA-CMAC has better control precision.

In [ARM09] the direct inverse control strategy based on MLP is proposed to control a Continuous Stirred Tank Reactor (CSTR) with the Van de Vusse reaction. For evaluation the direct inverse model controller is compared with PID controller. The simulation results dominated that the direct inverse model controller has superior performance to the PID controller. To improve the performance of the direct inverse model control, internal model control (IMC) structure is introduced. The internal model structure increases the robustness of the controller in the presence of perturbations in process parameters.

In [ML06] the expected state trajectory and corresponding control sequence are obtained using complex particle swarm optimization algorithm. Then MLP network is applied to learn the mapping between the expected state trajectory and control sequence. The approach aims for generalizing design method of neuro-controllers. A synchronous machine model is used to demonstrate the efficiency and simplify of the approach. The results show very good optimization and the solution is obtained in a short time.

In this section a short description in obtaining the inverse model controller based on MLP algorithm is provided, additionally the structure and resulting mathematical model are explained. The recent applications the inverse model controller based on MLP algorithm are summarized. it can be noted that the inverse model controller based on MLP is applied to several dynamic systems, unfortunately there are no information on the training time required to obtain such a controller or the computational load required to calculated the controller output. In section 3.3, the the inverse model controller based on MLP algorithm is compared with other the inverse model controller based on other soft computing algorithms.

2.2.3 Radial basis function-based direct inverse model control

Radial Basis Function (RBF) networks are considered as an alternative to MLP. A radial basis function network is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.

According to [PS91], it was proven that the RBF networks are universal approximator, just like neural networks and fuzzy inference systems. Additionally a RBF network is easy to design, show good generalization, strong tolerance to input noise, and online learning ability [YXPW11],

The structure of the inverse model controller based on RBF network is shown in figure 2.10. The mathematical representation of the inverse model controller based on RBF is given as

$$\hat{u}(k) = \sum_{i=1}^{I} w_i \sum_{j=1}^{J} \Big(\rho_{ij}(||v_{ij}^T q(k) + b_{ij} - c_{ij}||)\Big) + b_i, \tag{2.23}$$

where
j is the index of the neuron of hidden layer,
i the index of the neuron of output layer,
J the total number of neurons of hidden layer,
I the total number of neurons of output layer,
f the output activation function,
v the hidden layer weights vector,
w the output layer weights vector,
b bias term,
c the center vector for neuron and
$\rho(|| \quad ||)$ a Gaussian function. The parameters v, w, b, and c are obtained after training. According to [SKP01] the training of a RBF network is carried on in two steps. In the first step the parameters v, w, b and c are initialized, in the second step the center vector c is determined using K-means clustering. A fine tunning of the network can be obtained using a third step by applying backpropagation method.

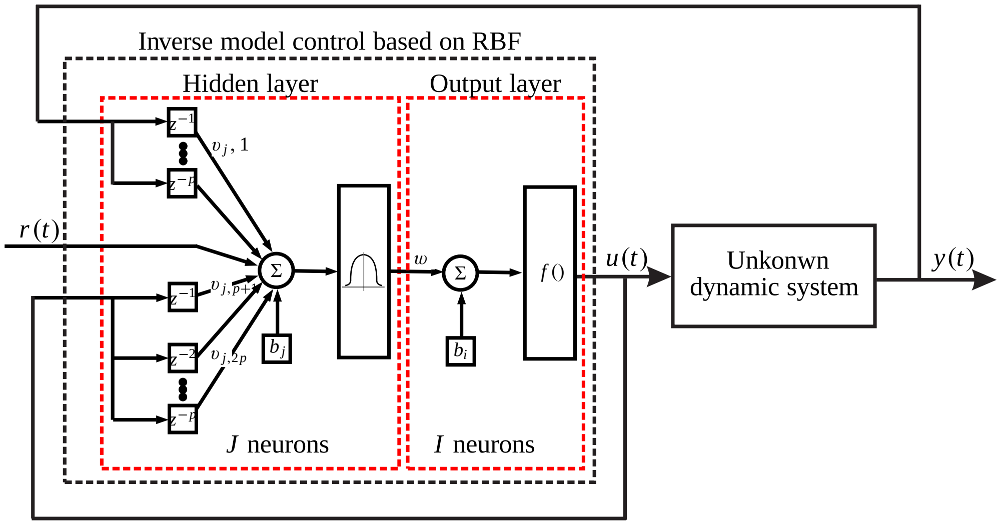

Figure 2.10: Inverse model control struacture using RBF [TS16a]

Recent applications of RBF-based inverse model control

In the following the most recent applications using RBF-based inverse model controller-based, are introduced:

In [JSX15] a direct adaptive controller based RBF to an omni-directional mobile robot (OMR) is proposed. The direct adaptive controller based on RBF uses improved double model structured RBF which allows the control of the nonlinear and um-modeled OMR system. The approach presented in [JSX15] guarantees the stability, this is proved using Lyapunov stability theory. Also, the controller provides robust control strategy for the OMR system. According to the simulation results in [JSX15], the controller demonstrates that relatively high accuracy.

In [AEGAA15] a DC servo motor is controlled using a self-tuning model inverse controller based on RBF. The self-tuning model inverse controller adapt the weight of the RBF network online using equations derived using the least mean squares principle. To increase the operating range, the DC servo motor system is exposed to different type of disturbances. The simulation results show that the self-tuning model self-tuning model inverse controller has very high performance in comparison to a self-tuning PI-controller.

A continuous stirred tank reactor (CSTR) is used in [ASL$^+$14] as a test rig to compare between a direct inverse model controller based on RBF and model predictive control (MPC). The inverse model controller is based on the work done in [ASKS13] where the RBF network is trained with the fuzzy means algorithm. To avoid extrapolation in the RBF the applicability domain concept is incorporated into the framework of the controller. Additionally, an error correction term is added, this allows the controller to account for modeling errors and process uncertainty. The simulation results in [ASL$^+$14] show that the inverse model controller base on RBF provides satisfactory performance in controlling the CSTR, while the MPC base on RBF outperforms the inverse controller terms of speed of responses. The work in [ASL$^+$14] provides no information about the computational load required by the controller.

A 3-DOF helicopter system is controlled using an inverse model controller based on RBF [DZR11]. The inverse model controller is obtained online. The controller is coupled to the 3-DOF helicopter system in series, which forms a pseudo linear system. This allows decoupling the 3-DOF helicopter system. The simulation results show that this control approach has high accuracy in tracking the desired reference.

In [XYWL10], an adaptive inverse model controller based on RBF is applied to control an electronic throttle. An electronic throttle is a DC-motor-driven valve that controls the air inflow an internal-combustion engine. In [XYWL10] the controller is realized using two RBF networks. The first RBF network identifies the plant provides the required the sensitivity information of the plant to the control input.

The second RBF network is trained to obtain the inverse model of the electronic throttle. To ensure fast learning and guarantee convergence adaptive learning rate is used. The simulation and experimental results show that the effectiveness of this approach over a PID controller with feedback compensator and recurrent neuro-controller.

A brief explanation of the inverse model controller based on RBF algorithm is given, followed by the structure, and mathematical function of the controller. A summary of the most recent applications of the inverse model controller based on RBF algorithm are given. A comparison of the inverse model controller based on RBF algorithm with other controllers soft computing is carried out in section 3.3.

2.2.4 Support vector machine-based direct inverse model control

The Support Vector Machine (SVM) algorithm is proposed in [VGS96, Vap98a, Vap98b, Vap99]. In the field of classification, the SVM algorithm is widely used. On the other hand the use of the SVM algorithm in the field of system modeling and control is not fully explored. In [SV02] the SVM algorithm is adapted to model an unknown dynamic system measured by input-output relations. The algorithm is known as least squares support vector machine. As an example given in [SV02], the double scroll system is successfully modeled using LS-SVM.

Support vector machine is a supervised learning method, which requires training data set. This data set consists of N samples of the system inputs u and outputs y. The data set is given as $D = \{u(k), y(k)\}_{k=1}^{N}$. The modeling process of the inverse model of the unknown dynamic system with the input vector input vector $u \in \mathbb{R}^m$ and output vector $y \in \mathbb{R}^r$ is done by estimating the function $f()$ given as

$$\hat{u}(k) = \sum_{j=1}^{M} \alpha_j K(q(k), q_j) + b, \tag{2.24}$$

where
α_k denotes the Lagrange multiplier,
$K(,)$ the Kernel function,
q_j a support vector,
b a bias term, and
M the number of support vectors.

The structure of the inverse model controller based on LS-SVM is shown in figure 2.11. It is noted in [HCL10] that if the number of features is large mapping of data into higher dimensional space is not needed. The inverse model control based on SVM using linear kernel is given as

$$\hat{u}(k) = \sum_{j=1}^{M} \alpha_j < q(k), q_j > +b, \tag{2.25}$$

where $<,>$ represent the dot product.

As given in [SV02], the Lagrange multiplier α_k and the bias term b are calculated by solving the following equation

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & Q^TQ + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ u \end{bmatrix}, \tag{2.26}$$

where
$Q = [q_1, ..., q_{N-p}]$,
$y = [u_1, ..., u_{N-p}]$,
$\vec{1} = [1_1, ..., 1_{N-p}]$,
$\alpha = [\alpha_1, ..., \alpha_{N-p}]$ and
γ is a positive real constant. In [YLJH03], it is concluded that the SVM has less tuning effort in comparison to MLP network. Also, it is noted that the SVM algorithm requires less computational time than MLP. This is due to the fact that the SVM algorithm uses linear equations to obtain the results while MLP network uses backpropagation algorithm. According to [YLJH03] the models obtained using the SVM algorithm has better generalization ability than models obtained using MLP network. Additionally, for the SVM no risk of getting stuck into local minima exists, thus the SVM always results in a unique and global solution [YLJH03].

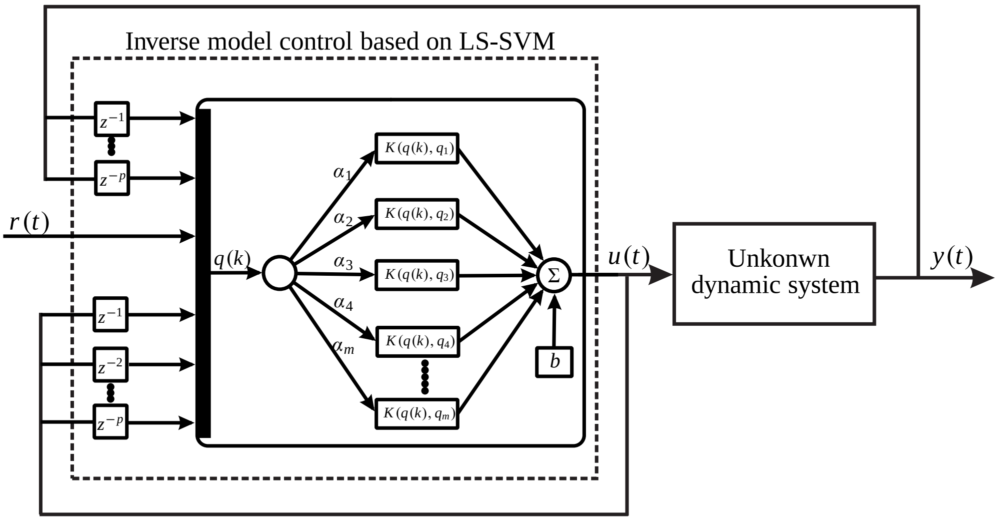

Figure 2.11: Inverse model control struacture using SVM [TS16a]

Recent applications of SVM-based inverse model control

In the following most recent applications using the support vector machine are briefly introduced:

In [ME14], an inverse model control scheme based on SVM for unknown nonlinear systems is proposed. The inverse model control based on SVM uses the Fuzzy C-Means clustering technique to build the controller. The controller is adapted online to obtain a suitable control action. Additionally, the inverse model controller is combined with linear controller to ensure the closed-loop stability. The simulation results show that the inverse model controller based on SVM has efficient performance and accurate response. The controller test is limited only to track 3 different steps reference signals.

In [Ged11], the velocity of a DC-motor is controlled using a direct inverse model controller based on SVM. The DC-motor parameters are changing, therefore it is challenging to find a suitable control strategy. The performance of the direct inverse model controller based on SVM is compared with a traditional PI-controller. The simulation results show that the SVM controller has high performance and good robustness in comparison with the PI-controller. The controller is evaluated using step response with two different amplitude, no further analyses are provided.

A direct inverse model control based on SVM is combined with an internal model control strategy to control a seam valve control of a synchronous generator [YNXF08]. The proposed controller includes two main parts, an SVM model that approximates the inverse characteristics of the system, and dynamic model of the system to compensate for uncertainties. The controller is compared with a PID-controller. The simulation results show that the inverse model controller combined with internal model control has high accuracy and effectiveness in tacking step wise reference signals. other than the step wise step signal there are no further analyses are preformed.

A direct inverse model controller is applied to obtain a suitable control strategy for a supercritical boiler is presented in [SWC07]. The inverse model is identified online. In order to overcome the inverse model controller static error and increase the robustness a PID-controller is introduced to work in parallel to the inverse model controller. The simulation results show that the controller achieve the desired step reference signal.

In [WPS07] an adaptive inverse model controller based on SVM is proposed. This controller combines fast online support vector regression (SVR) algorithm with straight inverse control algorithm. The fast online SVR algorithm is realized using a kernel cache-based method. The adaptive inverse model controller based on SVM is tested on two simulation examples, the first is a second order linear dynamic system introduced in [Xia97] and the second example is a second order nonlinear dynamic system introduced in [NP90]. The simulation results for both examples show that the controller is able accurately to follow a sequence of steps signal.Other test signals signals are not appled to test evaluate the performance of the controller.

In this section a inverse model control-based on SVM is introduced. At first the mathematical principle is repeated. Additionally the structure of the controller is provided. A short summary of the most recent applications is given. it can be noted that there are no information about the training time or computational load. The reference tracking capability of the SVM-based controller are not fully analyzed. In section 3.3 a comparison of inverse model controller based on SVM algorithm with other inverse model controllers based on soft computing algorithms carried on.

2.2.5 Fuzzy inference system-based direct inverse model control

A Fuzzy Inference System (FIS) is a system that uses Fuzzy Set Theory to map inputs to outputs. In [TS85, SY93] the effectiveness of modeling of nonlinear systems is demonstrated. The main disadvantage of FIS based on expert knowledge is that it lacks the desired accuracy. To overcome this problem the neuro-fuzzy modeling approach is introduced in [Jan93, CFCT98, JS01]. The neuro-fuzzy inference system acquires knowledge from a set of input-output data. In figure 2.12 the general structure of inverse model controller based on ANFIS is shown.

The overall output layer ($u = O_1^5$) of the inverse model control based on ANFIS is given as

$$\hat{u}(k) = O_1^5 = \sum_i \bar{w} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, \tag{2.27}$$

where $\bar{w}$ is given in equation 2.29. The forth layer output (O_i^4) is given as

$$\bar{w} f_i = O_i^4 = \bar{w}(p_i x + q_i y + r_i) \tag{2.28}$$

The third layer output(O_i^3) is the ratio of i^{th} rule's firing strength to the sum of all rules' firing strengths ($\bar{w}$) given as

$$\bar{w} = O_i^3 = \frac{w_i}{w_1 + w_2}. \tag{2.29}$$

The second layer output (O_i^2), is the product of the signals coming from the first layer, this is given as

$$O_i^2 = w_i = \mu_{A_i(x)} \times \mu_{B_i(x)}, \tag{2.30}$$

where $i = 1, 2$. The first layer output (O_i^1) is give as following

$$O_i^1 = \mu_{A_i}(x), \tag{2.31}$$

where x is the input vector to node i, A_i is the linguistic label associated with this node function. The membership function is typical bell-shaped function given as

$$\mu_{A_i(x)} = \frac{1}{1 + \left(\frac{x - c_i}{a_i}\right)^2 b_i}. \tag{2.32}$$

or

$$\mu_{A_i(x)} = e^{-\left(\frac{x - c_i}{a_i}\right)^2} \tag{2.33}$$

where a_i, b_i and c_i are parameters to be trained.

The training of the ANFIS is achieved using a hybrid learning algorithm as presented in [Jan93]. In the ANFIS algorithm the least square method is used in the forward path to identify the consequent parameters in the fourth layer. In the backward pass the errors are back propagated and the parameters are updated using gradient descent algorithm. The structure of The inverse model controller based on ANFIS network is shown in figure 2.12 .

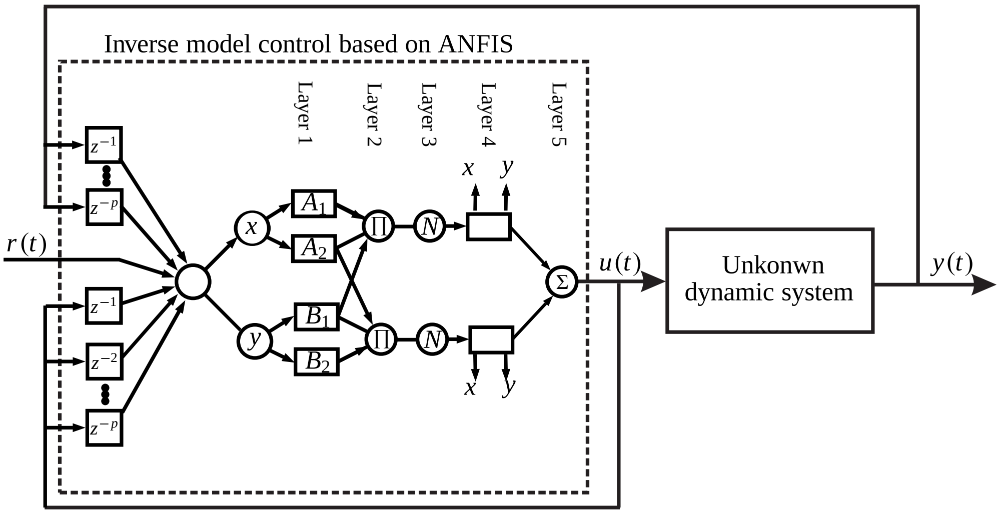

Figure 2.12: Inverse model control struacture using ANFIS [TS16a]

Recent applications of ANFIS-based inverse model control

In the following the most recent applications using the Fuzzy Inference System are briefly introduced: In [JRP+15], an extreme-ANFIS is used to learn the inverse characteristics of CSTR. The extreme-ANFIS is a simple and fast learning algorithm that can be used to tune the parameters of Takagi-Sugeno Fuzzy Inference System. Also, the extreme-ANFIS provides better generalization than conventional hybrid learning approaches. The inverse model controller baes on ANFIS is compared with conventional PID-controller. The simulation results show that the inverse model controller based on ANFIS is effective and accurate controller.

A cable-driven parallel system is controller using a direct inverse model controller based on ANFIS in [WT15]. The inverse model controller uses type-2 fuzzy logic systems to learn the inverse dynamic model of the cable-driven parallel system. Type-2 fuzzy logic system has better handling of the different sources of uncertainties in comparison with traditional fuzzy logic system. In [WT15], the cable-driven parallel system is divided into six subsystems, and a direct inverse controller is realized for each subsystem.

The experimental results show that the direct inverse model controller based on type-2 fuzzy logic systems can achieve successfully its goals. The controller is applied only to level the top surface of a payload, and balance the tensions of the four cables.

In [LRH13] a direct inverse model controller based on ANFIS is proposed to control an Electro-Hydraulic Actuator (EHA). The experimental results show that the direct inverse model controller based on has good reference tracking for step and sine wave signals. Also the direct inverse model controller based on ANFIS is able to handle external disturbances to the EHA system.

A control scheme that combines an inverse model controller based on ANFIS with Iterative learning control is proposed in [Has11]. The goal of introducing the iterative learning control is to improve the tracking performance of the controlled system iteratively by using past experiences. The proposed scheme in [Has11] is tested to control the fluid level in two coupled tank system. The simulation results show that the controller has no overshot and zero steady state error additionally it has very accurate reference tracking for a square wave signal.

In [CDJQYDB10], a MLP network is combined with type-2 fuzzy network to generate a model inverse controller based on a type-2 fuzzy neural network (T2FNN). The inverse model controller is used to control the level of a cable-driven parallel mechanism. The iterative least squares estimation method is used for parameters learning. The approach in [CDJQYDB10] is compared with model inverse control based on type-1 fuzzy neural network (T1FNN) using experimental results. The experimental results show that the performance of T2FNN is more consistently than T1FNN.

In this section the inverse model controller based on ANFIS algorithm is introduced. In addition to the structure and the mathematical function are presented. A short survey of the most recent applications of the inverse model controller based on ANFIS algorithm is provided. An evaluation of inverse model controller based on ANFIS with respect to other inverse model controller based on soft computing algorithms is provided in section 3.3.

2.3 Summary

In section 2.1 a brief discussion of the five main soft computing algorithms used for modeling dynamic systems was provided. Here Multi-Layer Perceptron (MLP) network, Support Vector Machine (SVM), Gaussian Process, Radial Basis Function and Fuzzy Inference System (FIS) were introduced. The training methods of each algorithm were described, and the structure of the model for dynamic systems using the algorithm was shown. Additionally the most recent developments made in the recent years were provided.

In section 2.2 a discussion and comparison of the four main soft computing algorithms used to build inverse model controller was summarized . Here Multi-Layer Perceptron (MLP) network, Support Vector Machine (SVM), Radial Basis Function and Fuzzy Inference System (FIS) were considered. The training methods of obtaining the inverse model controller using each soft computing algorithms were provided, and the different structures of the inverse model controller using soft computing algorithms were shown. Additionally the most recent applications in the recent years were provided.

3 Evaluation of soft computing algorithms in modeling and control of dynamic systems

In this chapter soft computing algorithms used for modeling dynamic systems or obtaining direct inverse model controller are evaluated. The evaluation is done using a simple nonlinear benchmark example. The evaluation results can be used as a measure for the suitability of the soft computing algorithms in modeling and control dynamic systems. Parts from this chapter are based on text and material that are previously published [TS16c, TS16a].

3.1 Benchmark and test signals

In this section the benchmark example used for evaluation of the different soft computing algorithms is descried. Also the training signal used to training/learning the dynamic model of the benchmark and the direct inverse model controller is given. Finally the test signals used to compare and evaluate the soft computing are provided.

3.1.1 Benchmark

In [KS14] several benchmark examples are provided which has been used in different soft computing techniques (methods). It is noted that insufficient information about benchmark examples prevent evaluation, reproduction, and comparison. It is also noted that information about the prediction error if its refers to one step ahead prediction or multi-step ahead predictions are missing. To overcome this inherent problem in this contribution simple and easy to operate dynamic systems are chosen as examples. In this work, the dynamic system given in [NRPH00] is selected to be the benchmark example given as

$$\ddot{y} + \dot{y} + y + y^3 = u \tag{3.1}$$

with u as input/force and y as output/displacement.

3.1.2 Training and test signals

A good training signal should cover the whole operating range of the system of interest. Here the Multilevel Pseudo-Random Signal (MLPRS) (figure 3.1) is chosen. The Multilevel Pseudo-Random Signal is uniformly distributed signal with the range ±4.8 and with the duration of 200 seconds. The smallest segment in the training signal has a duration of 0.5 second while the longest segment has a duration of

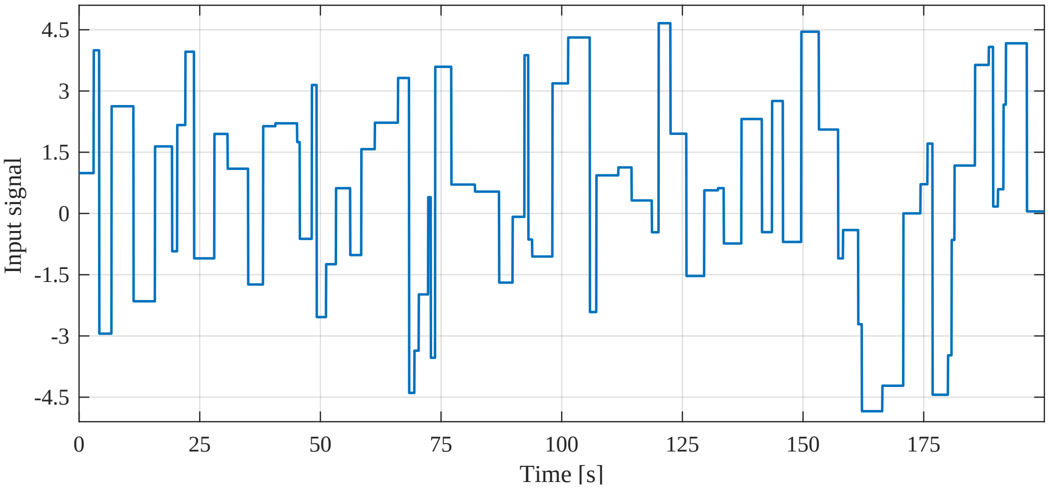

Figure 3.1: Test signal 1: Multi-level Pseudo-Random Signal [TS16c]

5 seconds. For testing four test signals are used: steps sequence, sawtooth wave, triangular wave, and chirp. In figure 3.2 a sequence of steps is shown. The duration of the entire test signal is 30 seconds with the amplitude of 2. Each segment has a duration of 5 seconds. Furthermore a sawtooth wave is used as test signal. The

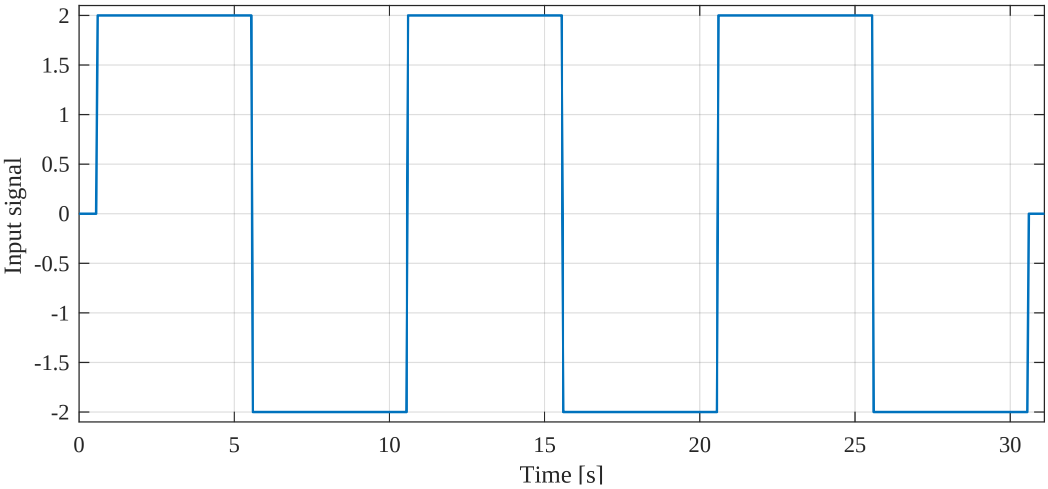

Figure 3.2: Test signal 2: Step sequence [TS16c]

signal resembles the teeth of a saw as shown in figure 3.3. The duration of the signal is around 15 seconds, the amplitude of the signal is 2, and slop of the ramp is 0.02. The triangular wave test signal is shown in figure 3.4. The signal's duration is around 42 seconds. The signal amplitude is 1 and the slope is 0.01. The last test signal to be used is the chirp signal which is a sine wave with gradually increasing

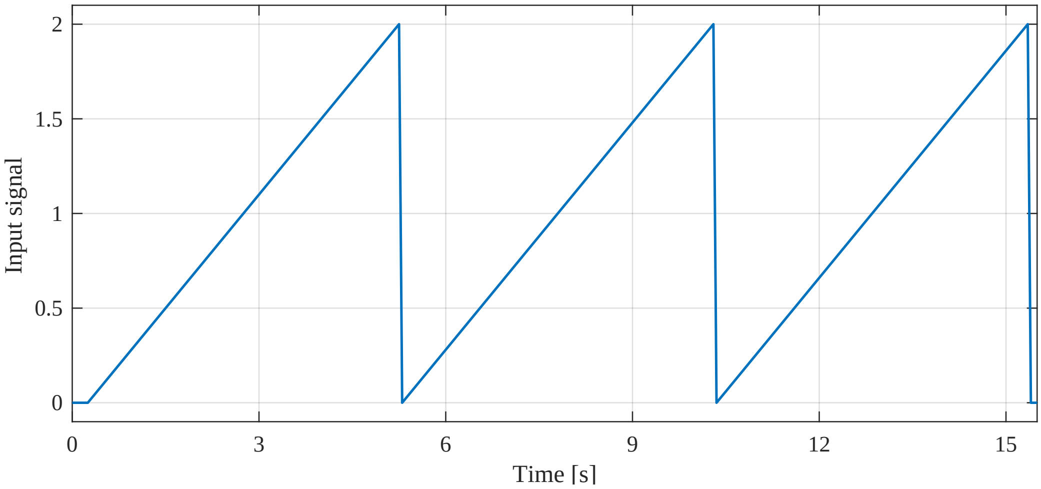

Figure 3.3: Test signal 3: Sawtooth wave signal [TS16c]

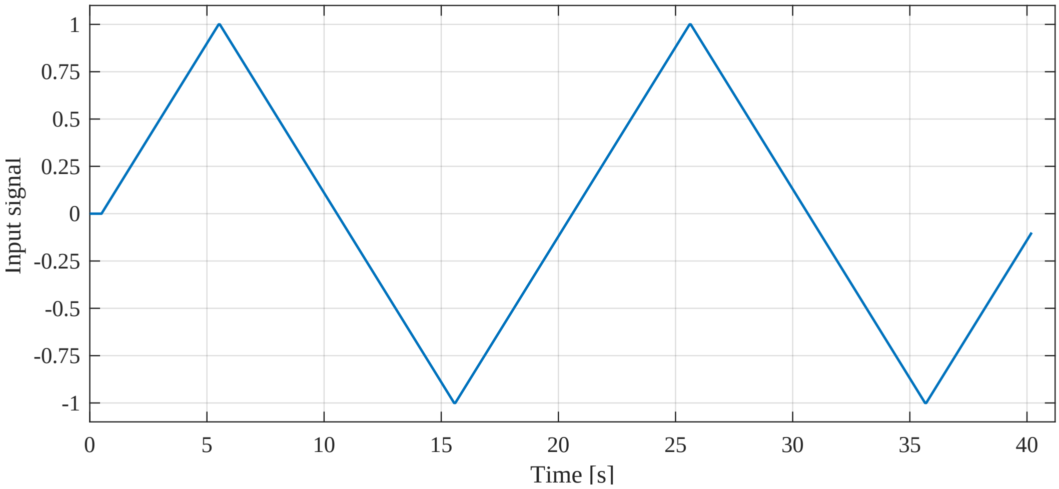

Figure 3.4: Test signal 4: Triangular wave signal [TS16c]

frequency as shown in figure 3.5. The signal duration is around 20 seconds with amplitude of 2 and initial frequency of 0.1 rad/s and final frequency of 3 rad/s. The four test signals are used to evaluate the generalization ability of the trained models using the different soft computing algorithms.

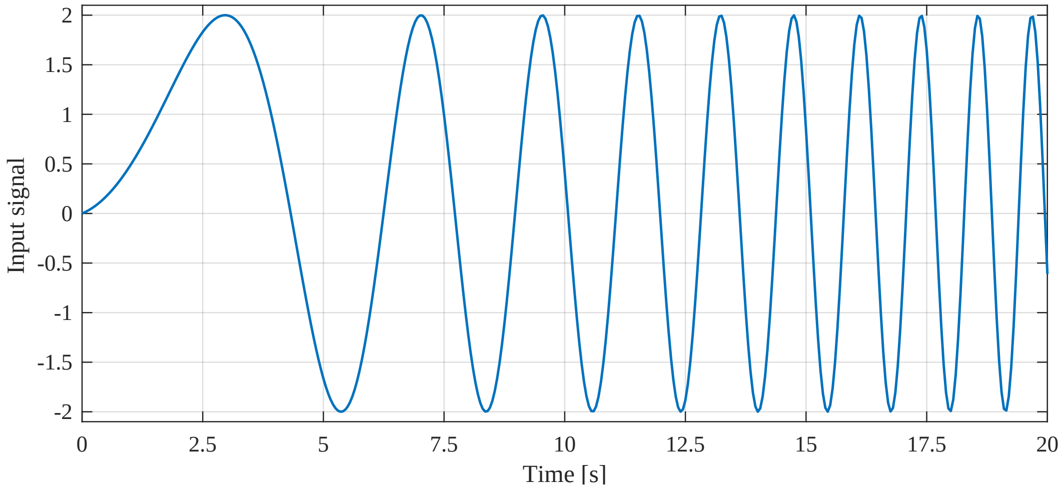

Figure 3.5: Test signal 5: Chirp signal [TS16c]

3.2 Evaluation of the soft computing algorithms in modeling dynamic system

The main goal of the chap is to evaluate of the performance the above soft computing algorithms for the purpose of modeling of dynamic systems.

3.2.1 Learning/training time

The time required by the algorithm to converge to an acceptable model is referred to as learning time/ training time. To compare the approaches (eq. 3.1), test runs are realized. The results with respect to learning time are shown in figure 3.6. From the results, it can be obtained that the ANFIS algorithm needs less time than the other four algorithms. It also can be observed that the GP algorithm requires the most learning time. The results shown in figure 3.6 provide an indication on the suitability of the algorithms for online applications. It should be assumed that the dynamic system behavior is slower than the training process else the algorithms are not suitable for online training. It should be noted that all the results are obtained using same desktop PC which has 3.4GHz processor.

3.2.2 Training parameters

Each of the above described algorithms has several parameters to tune the learning process to obtain an accurate model. In this work the main tuning parameters are considered. In table 3.1 a summary of the parameters needed to be selected (tuned)

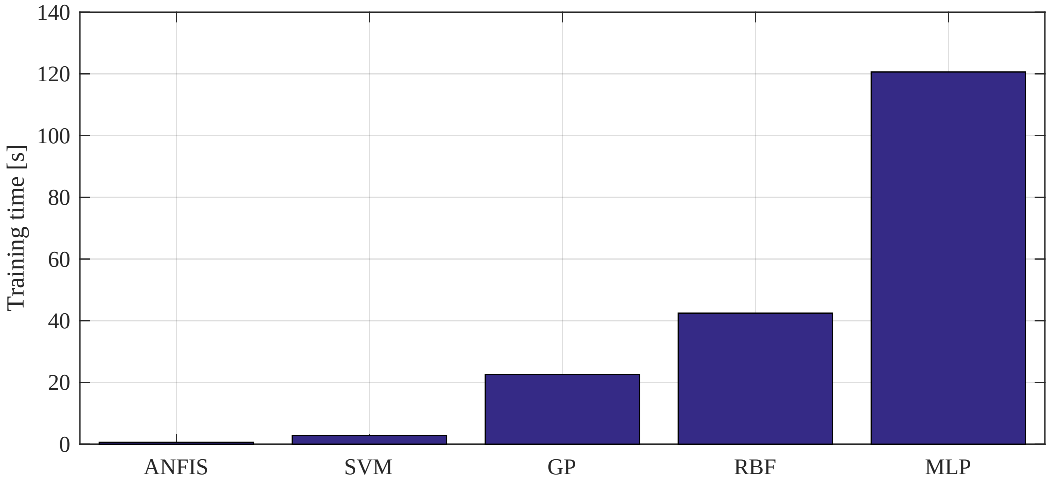

Figure 3.6: Training Time [TS16c]

to obtain the desired model, is given. The most popular choice for the number of layers for MLP and RBF networks is three; one input layer, one hidden layer and one output layer. However, there is no proof that this structure will provide a good model.

Table 3.1: Soft computing primary training parameters [TS16c]

Algorithm	Primary tuning parameter	Total number
MLP	Number of layers, number of neurons, activation function of the neurons,training algorithm, learning rate number of epoch	5
RBF	Number of layers, number of neurons, training algorithm, learning rate number of epoch	4
ANFIS	Specifies the number of membership functions, training method, training epoch number, learning rate, number of epoch	5
GP	Covariance function, inference methods, likelihood functions	3
SVM	Kernel function, cost parameter	2

3.2.3 Generalization

The ability of the model to realize accurate results on others than training dataset is called generalization. In control system application, it is important to work with models with high generalization capability. In this section, the generalization capability of the discussed algorithms are presented.

In [KS14] several assessment criteria for evaluating the performance of dynamic models or controllers are summarized. For modeling one of the following criteria can be used to provide quantitative assessment of the model performance:

- Maximum absolute error

$$J_{MAE} = J_{max} = \max_{1 \leq k \leq N} |y(k) - \hat{y}(k)| \tag{3.2}$$

- Sum of squared errors (SSE)

$$J_{SSE} = \sum_{k=1}^{N} |y(k) - \hat{y}(k)| \tag{3.3}$$

- Mean squared error (MSE)

$$J_{MSE} = \frac{1}{N} \sum_{k=1}^{N} |y(k) - \hat{y}(k)| \tag{3.4}$$

- Root mean squared error (RMSE)

$$J_{RMSE} = \sqrt{J_{MSE}} \tag{3.5}$$

Other model evaluation criteria count the number of features used in the model. These criteria are not considered in this work since the number of the features for each model is the same. Here for evaluation of the models the sum of square SSE is used.

Different test signals are used to test generalization ability. In figure 3.7 the estimation results of a sequence of steps (figure 3.2) are shown. All models show good estimation of the dynamic system output given in equation 3.1. The estimation results of the dynamic response of the system for triangular wave signal presented (figure 3.4) are shown in figure 3.8. The best accuracy is achieved by the SVM model. The error is less than ±0.05m for all models except for RBF model. The estimation results related to the response of the dynamic system to a sawtooth wave (figure 3.3), are shown in figure 3.9. All models have good estimation. It can be observed that the estimation error is between ±0.05m. The SVM-based model has the best estimation. The results for estimating of the system response to a sweep signal demonstrated (figure 3.5), are shown in figure 3.10. The estimation errors of the model are between ±0.08. The best performance is given GP-base model.

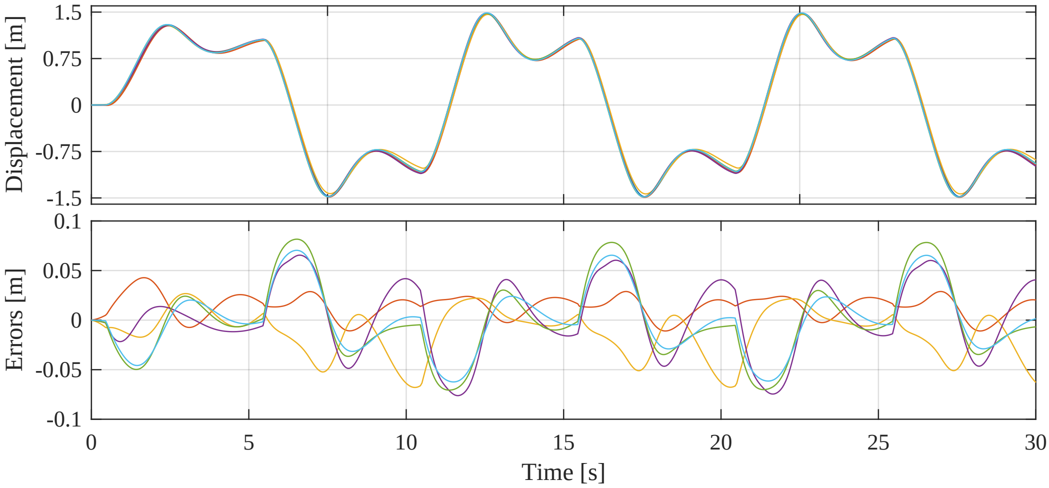

Figure 3.7: Models output to steps sequence

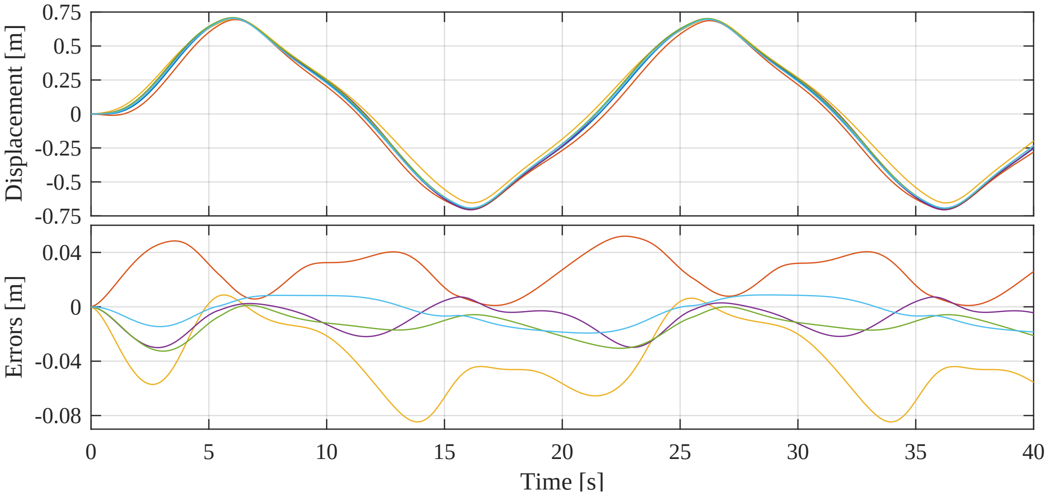

Figure 3.8: Models output to triangle input

The resultant sum of squared errors for all test signals (SSE) is shown in figure 3.12. It can be observed that the SVM-based model has the best performance since it has the least resultant SSE values. The GP-based model has the second best perforce and it can be noted that the RBF-based model has the highest resultant SSE.

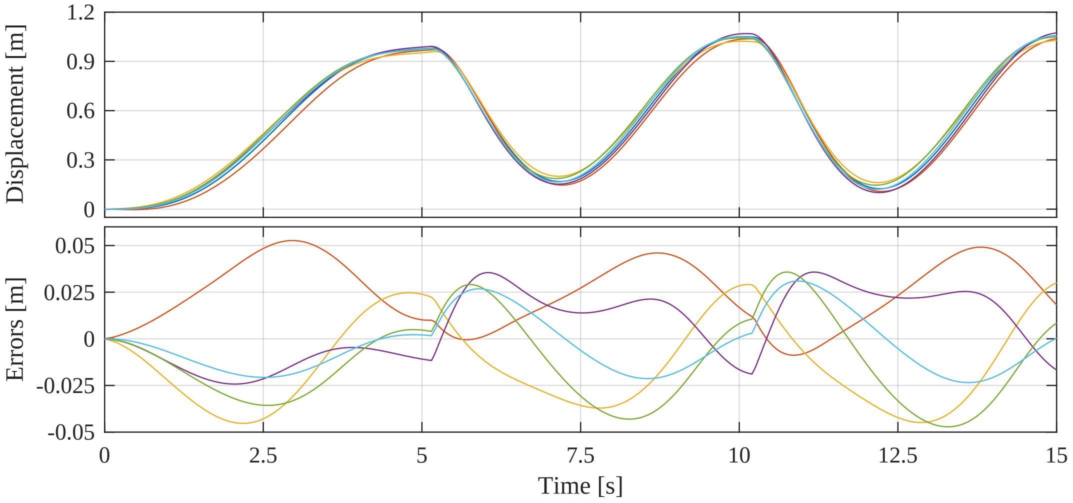

Figure 3.9: Models output to sawtooth wave input

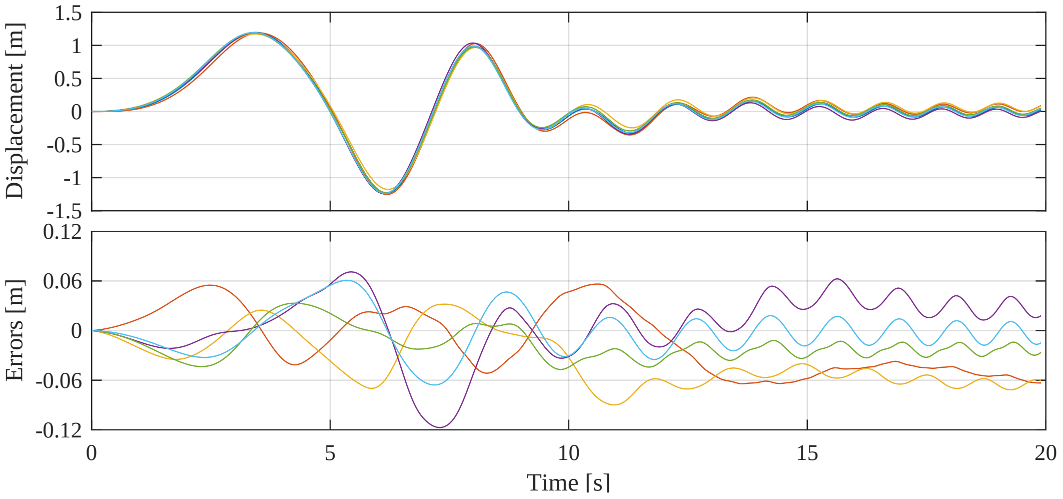

Figure 3.10: Models output to sweep input

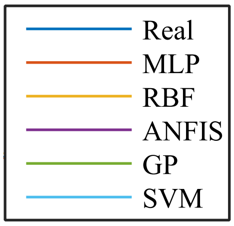

Figure 3.11: Color assignments [TS16c]

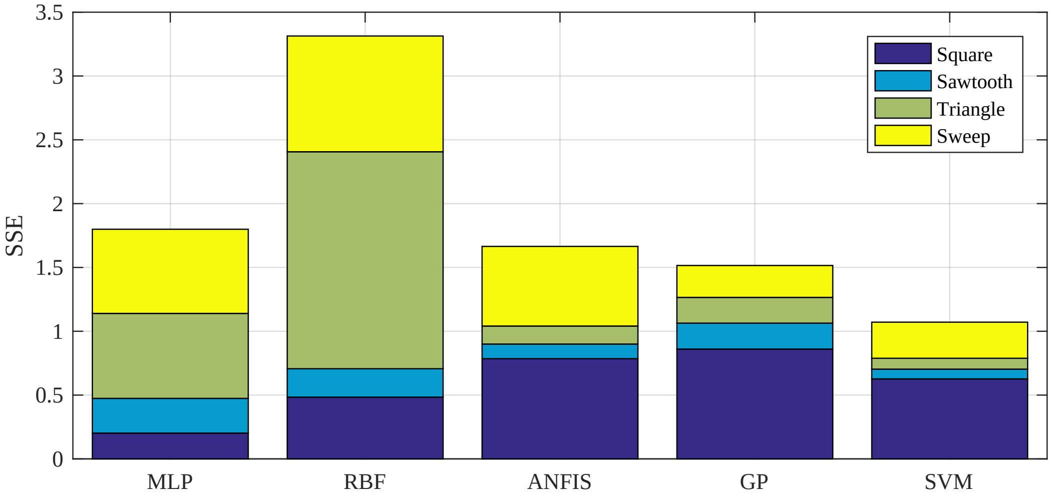

Figure 3.12: Resultant Sum of squared errors [TS16c]

3.2.4 Prediction time

The time required to predict n steps is referred to as prediction time. For real time applications, it is important to use models that can achieve n steps within a time period less than the sampling time. The total prediction time for the five models is illustrated in figure 3.13. The GP-based model requires the most computation time followed by SVM-based model. The larger computational load of GP-based model is a result of the results from computing of the (large) covariance matrix which is computed for each prediction step. The RBF-based model has the shortest computation time, followed by MLP-model. The MLP-based model and RBF-model are considered relatively simple in comparison with GP-model therefore require less computation process. It should be noted that all the results are obtained using same desktop PC which has 3.4GHz processor.

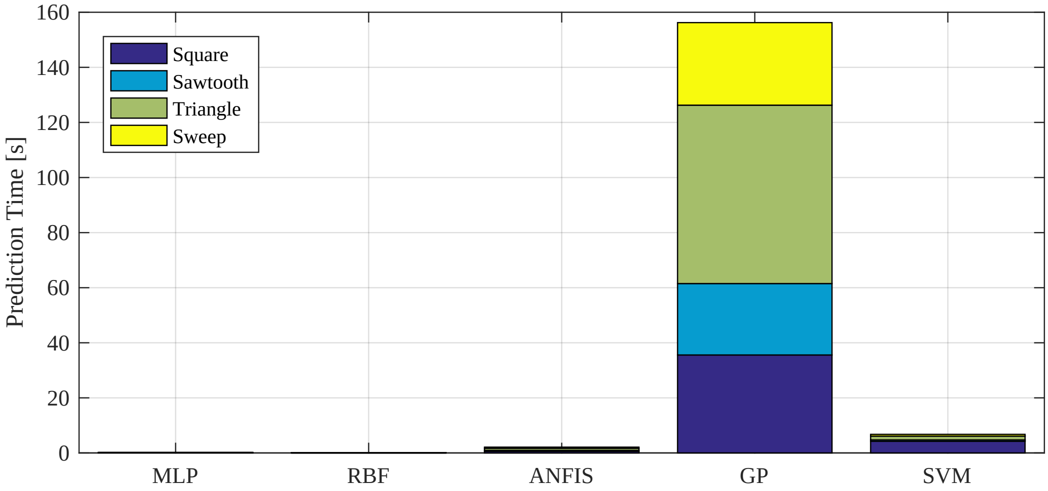

Figure 3.13: Total prediction Time [TS16c]

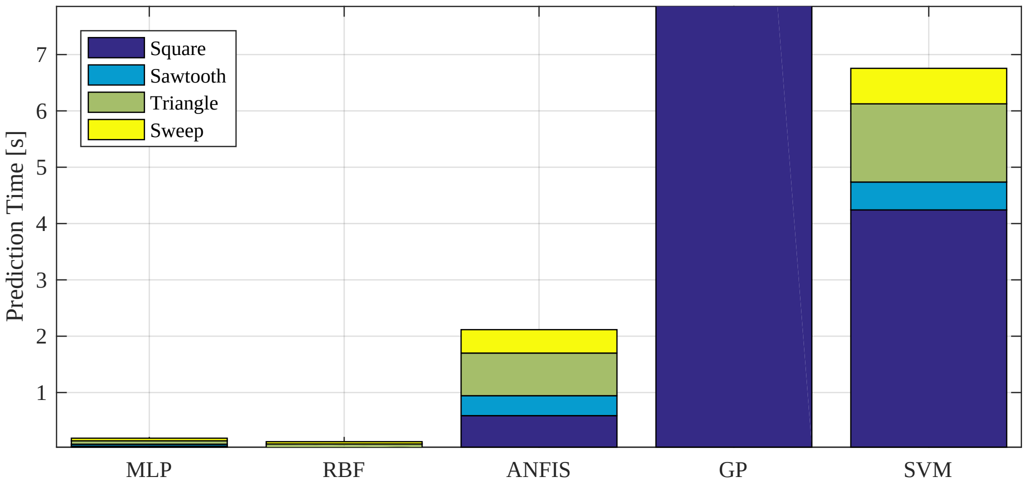

Figure 3.14: Total prediction time: Zoom in [TS16c]

3.3 Evaluation of the direct inverse model control based on soft computing algorithms

The main goal of the section is to evaluate of the performance the above soft computing algorithms for the purpose of dynamic systems control.

3.3.1 Learning time

The time required by the algorithm to converge to an acceptable inverse model of the targeted dynamic system is referred to as learning time/ training time. In figure 3.15 the time required for each algorithm to produce an inverse model for the system given in equation 3.1 using four aforementioned algorithms is illustrated. From the results, it can be obtained that the ANFIS algorithm needs less time than the other three algorithms.

The suitability of the algorithms for online training applications can be concluded from figure 3.15. To be able obtain a suitable inverse model controller, the training process should be faster than the behavior of the targeted dynamic system else the algorithms are not suitable for online training. It should be noted that the speed of the training depends on the size of the training dataset so a small training dataset denotes faster training. It should be noted that all the results are obtained using same desktop PC which has 3.4GHz processor. Running an inverse model controller based on soft computing on real time application requires that the average computational time is much less than the sample time provided by the hardware. In figure 3.16 the average computational time of 100 steps required for the different inverse model controllers based on soft computing algorithms is given. It can be observed that the inverse model controller based on SVM requires less computational load than the other inverse model controllers.

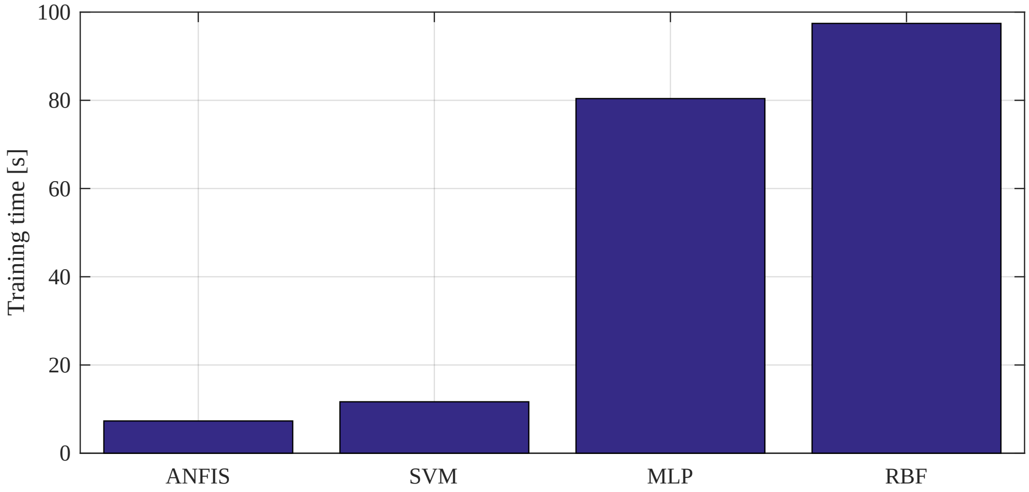

Figure 3.15: Controllers training time [TS16a]

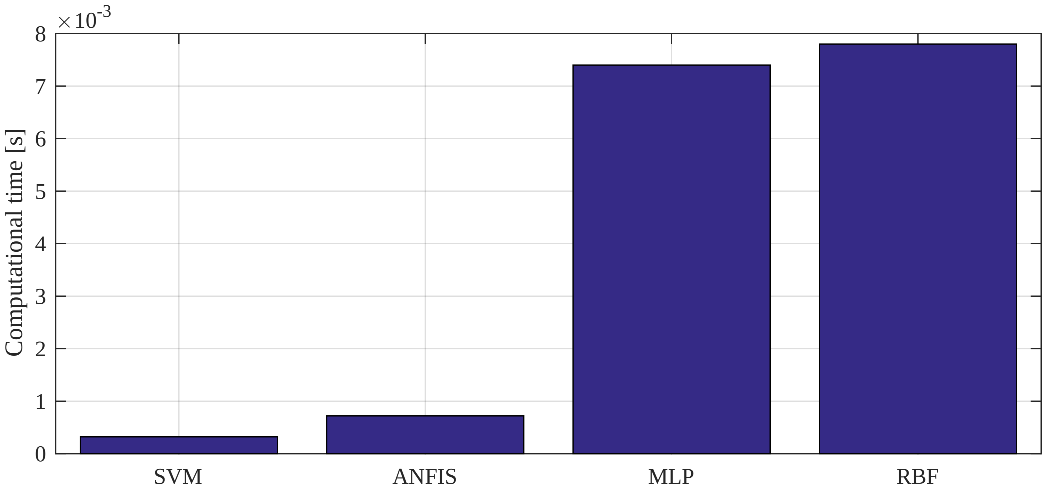

Figure 3.16: Controller mean computational time required for one step [TS16a]

3.3.2 Training parameters

Soft computing algorithms have tuning parameters that determine the speed of the training process and the quality of the resulting model. The same training parameters considered for obtaining the dynamic model are considered for obtaining the inverse model controller as given in section 3.2.2 table 3.1.

3.3.3 Performance of the direct inverse model control based on soft computing algorithms

Different criteria to measure controller performance are summarized in [KS14]. Here the soft computing algorithms-based controllers are evaluated using the step response which include rise time, settling time, percent overshoot (PO) and steady-state error. In addition to the over all controller effort (J_u) given as

$$J_u = \int_{t=0}^{T} u^2(t)dt. \tag{3.6}$$

The performance of controllers are summarized in tables 3.2 and 3.3. From table 3.2 it can be noted that all four controller have acceptable performance in term of the steady state error, fast rise time, and settling time. Also, it can be noted that ANFIS-based inverse controller shows strong overshoot.

Table 3.2: Performance of inverse model controller based on soft computing algorithms [TS16a]

Algorithm	Rise time [s]	Settling time[s]	PO (%)	Steady-state error
MLP	0.3114	1.9279	8.94	1.20×10^{-3}
RBF	0.3097	1.9355	11.93	5.5×10^{-3}
ANFIS	0.2355	2.809	34.46	0.1×10^{-3}
SVM	0.2270	2.150	12.71	0.1×10^{-3}

The control effort for each controller is shown in table 3.3 where it can be noted that the ANFIS-based inverse model controller consumes more energy with respect other three controllers. A conservative conclusion can be made from tables 3.2 and 3.3, that SVM-based inverse model controller has higher accuracy, fast response time, produces acceptable percent overshot and consumes moderate amount of energy. Therefore it can be concluded that the SVM-based inverse model controller has a slightly better performance than the other three controllers.

Table 3.3: Control effort of the inverse model controller based on soft computing algorithms [TS16a]

Algorithm	Control effort
MLP	127.61
RBF	112.78
ANFIS	940.90
SVM	283.46

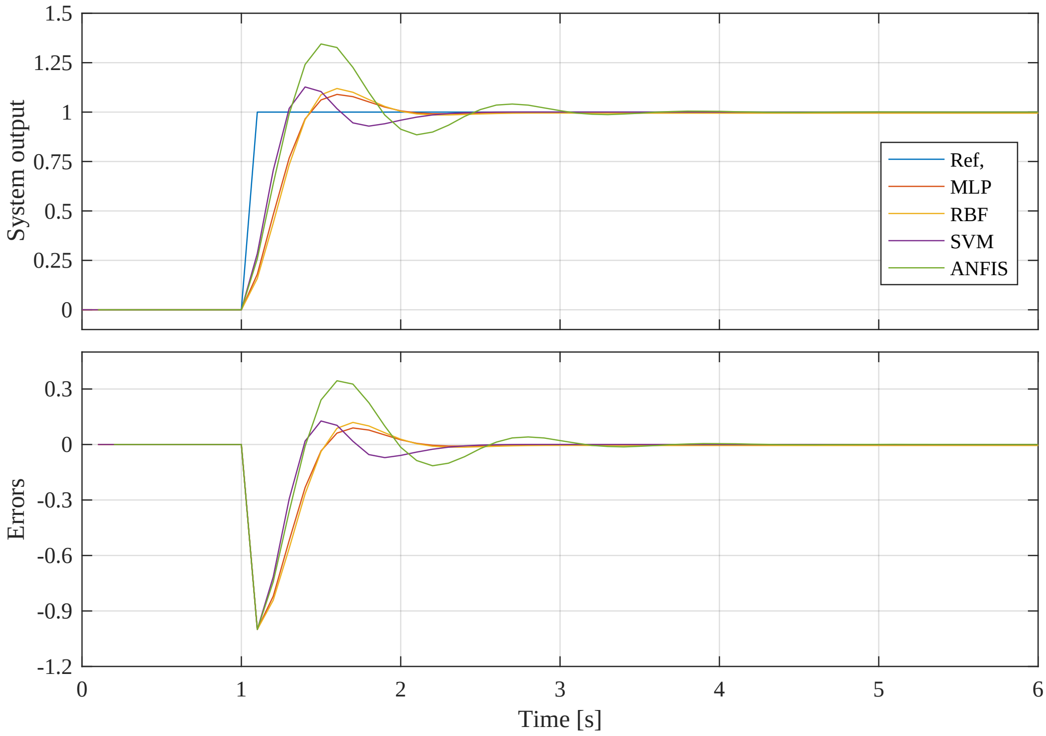

Figure 3.17: Closed-loop step response [TS16a]

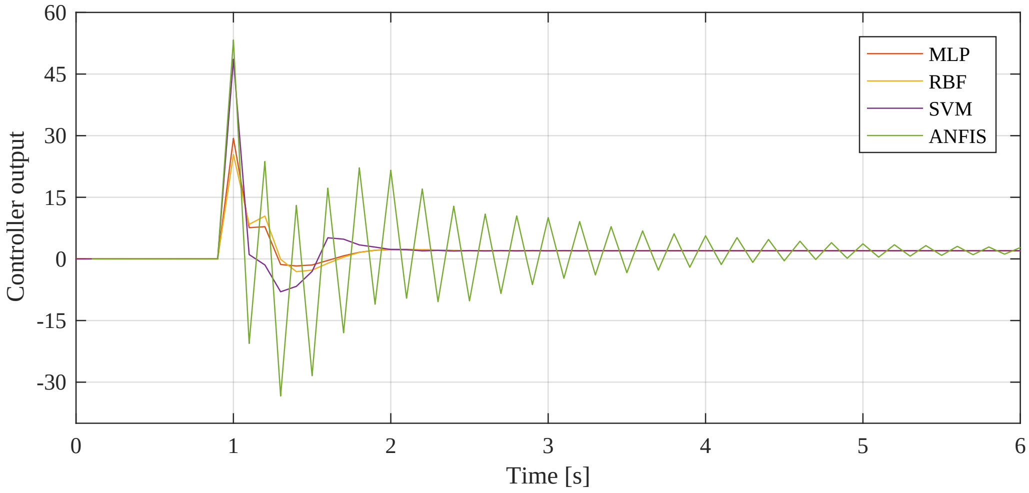

Figure 3.18: Controller output [TS16a]

3.3.4 Reference tracking

A second performance measurement of controllers is the reference tracking ability. In this section 4 standard reference tracking are provided. The reference signals are triangular wave, sawtooth, square sequence, and sweep. In figure 3.19 the triangular wave signal is shown. The signal's duration is around 42 seconds. The signal amplitude is 1. The sawtooth signal is shown in figure 3.20. The signal resembles the teeth of a saw. The duration of the signal is around 15 seconds, the amplitude of the signal is 2. The square signal is shown in figure 3.21. The duration of the entire test signal is 30 seconds with the amplitude of 2. Each segment has a duration of 5 seconds. The sweep signal is shown in figure 3.22. The signal duration is around 50 seconds with amplitude of 1 and initial frequency of 0.01 rad/s and final frequency of 1 rad/s.

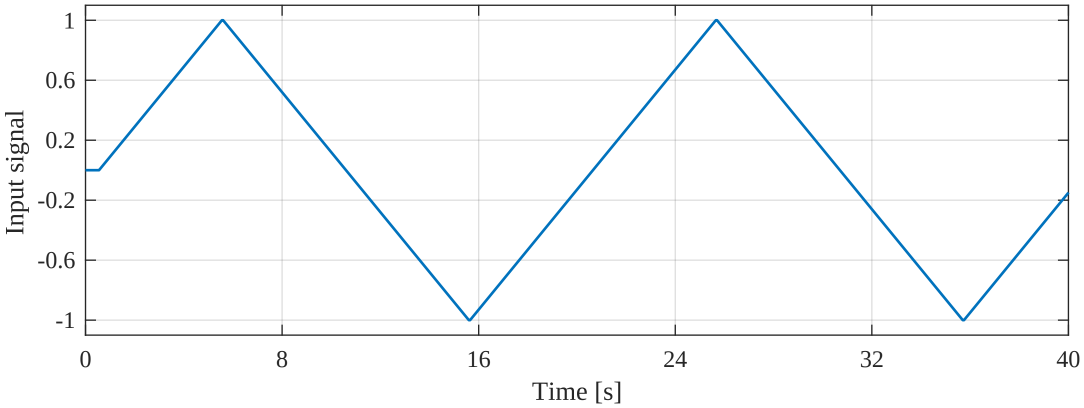

Figure 3.19: Reference signal: triangular wave [TS16a]

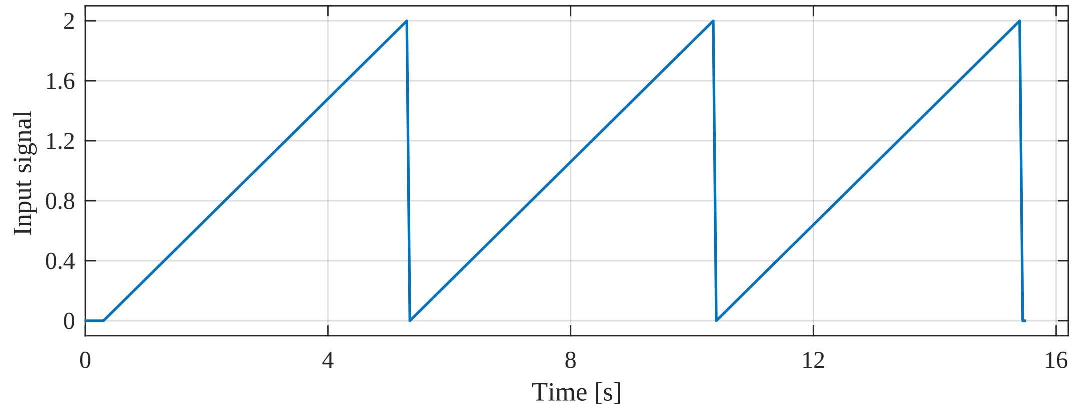

Figure 3.20: Reference signal: Sawtooth [TS16a]

The results of reference tracking test are shown in figures 3.24, 3.25, 3.26, and 3.27. In general, all four inverse model controller are able to accurately track the

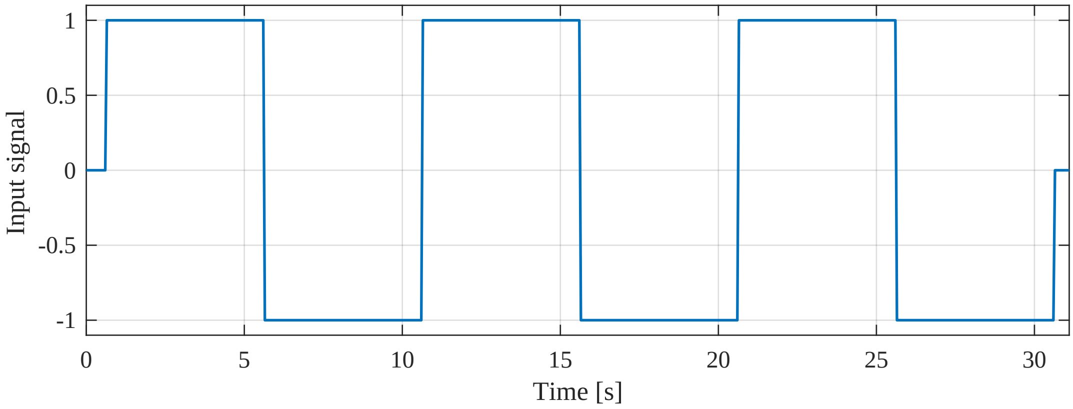

Figure 3.21: Reference signal: Square signal [TS16a]

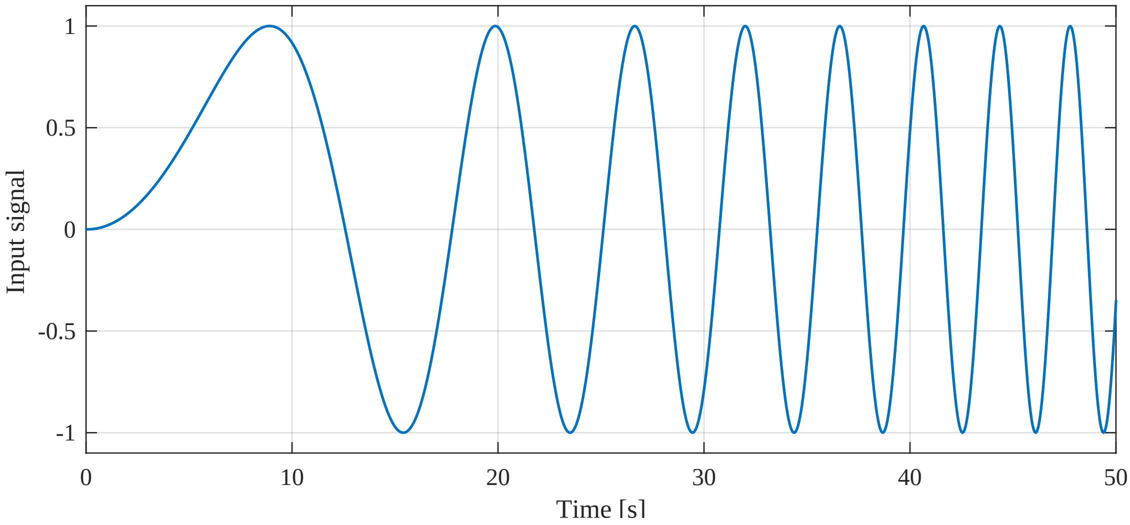

Figure 3.22: Reference signal: Sweep signal [TS16a]

desired reference. In Figure 3.24 the results of triangle wave are shown, it can be observed that all inverse model controller based on soft computing are able to track the desired reference signal. The same accurate tracking results as results tracking of sweep signal as shown in figure 3.26. It can be observed that with increase of the frequency the error increases. In figures 3.25 and 3.27, the results of inverse model controllers based on soft computing are not as high due the sharp change in the reference signals.

In figure 3.28 the overall tracking performance is shown, it can be observed that the RBF-based inverse model controller has most accuracy. The other three controllers have similar tracking performance. It can be observed that the major contribution of the tracking error is as results of the square wave signal tracking and the tracking error contribution of the triangle wave signal is so small and it could not be observed

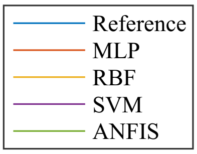

Figure 3.23: Color assignments [TS16a]

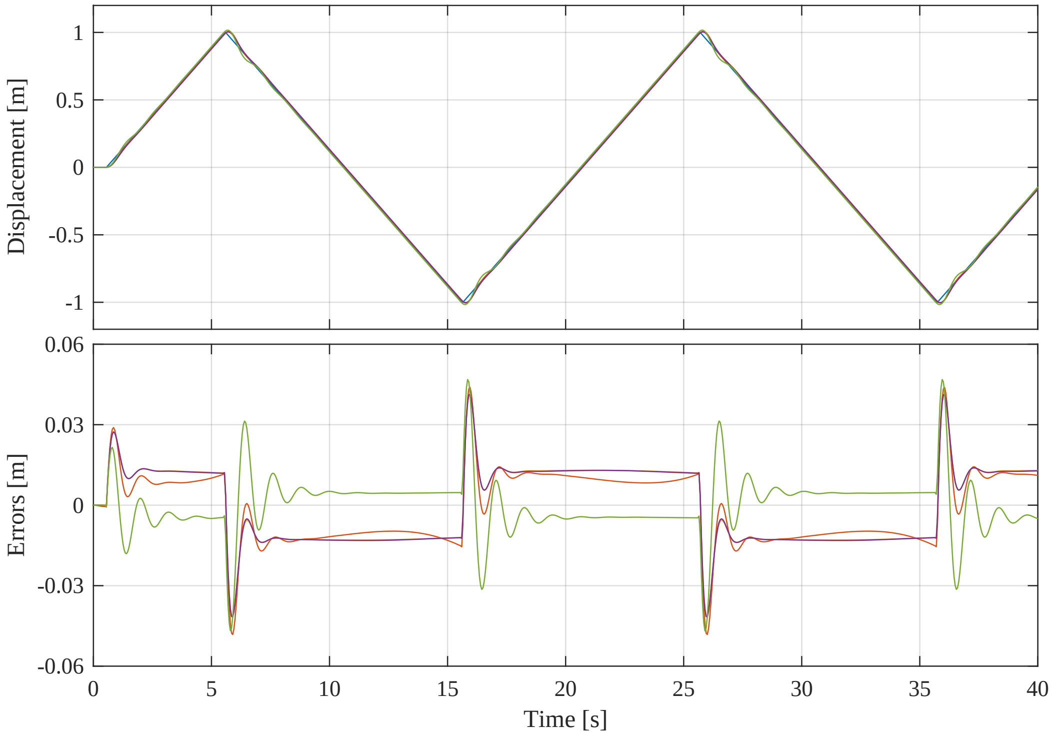

Figure 3.24: Reference tracking: Triangle wave[TS16a]

with respect to other reference signals tracking error. The overall control effort of all inverse model controllers based on soft computing is shown in figure 3.29. It can be observed that the MLP-based inverse model controller consumes the least energy with respect to the other controllers. it is also can be observed that the tracking of the square wave requires the most energy with respect to tracking the other test signals. Tracking of the triangle wave signal requires the least effort by all controller and therefor it can be observed.

The finial evaluation of the inverse model controllers based on soft computing if performed using the tracking error control effort proposed on [LS09]. The results of this approach are shown in figure 3.30, it can be observed that the MLP-based inverse model controller has the best performance followed by the the SVM-based inverse model control. It can be noted the ANFIS-based inverse model controller has the least performance effort.

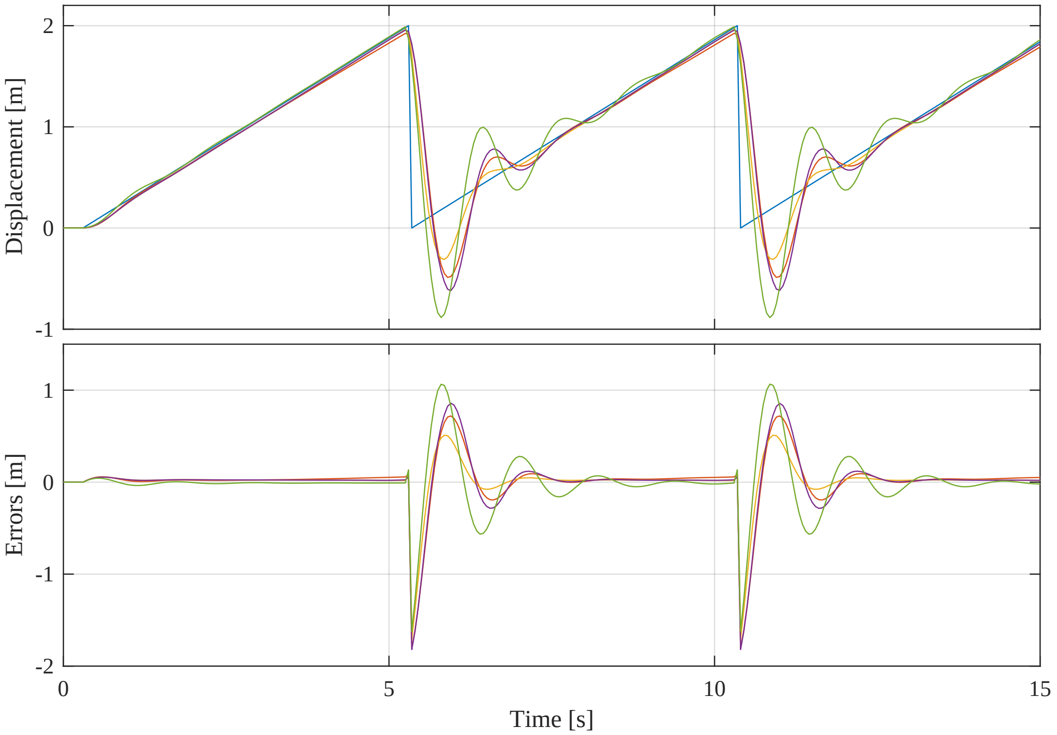

Figure 3.25: Reference tracking: Sawtooth wave [TS16a]

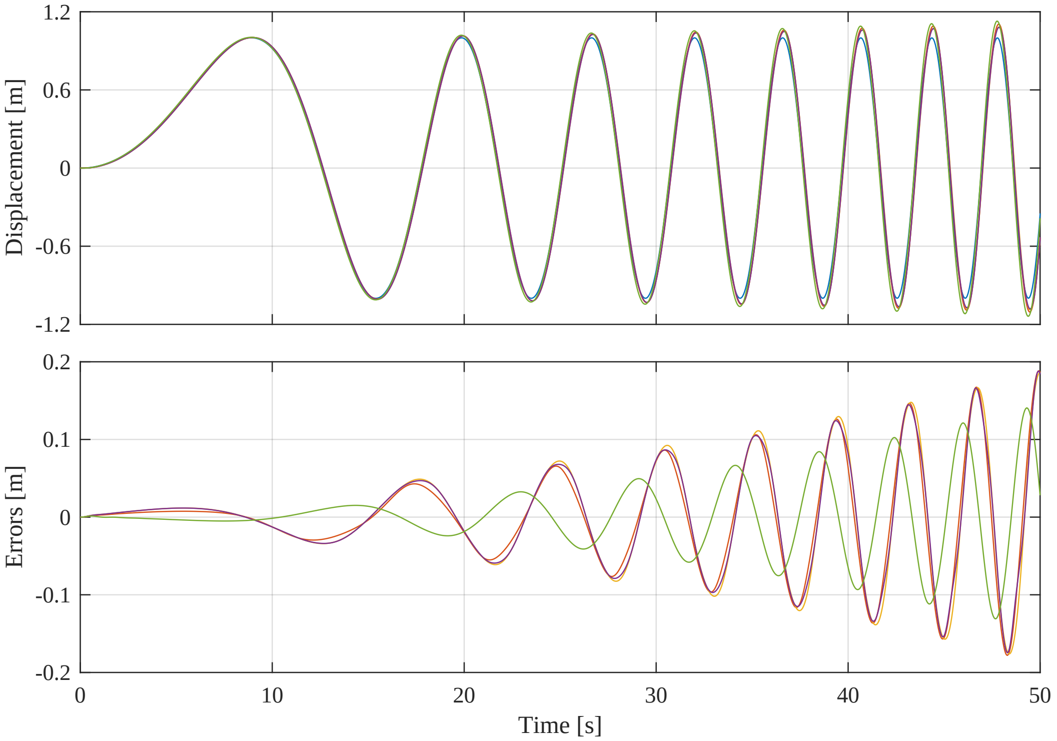

Figure 3.26: Reference tracking: Sweep signal [TS16a]

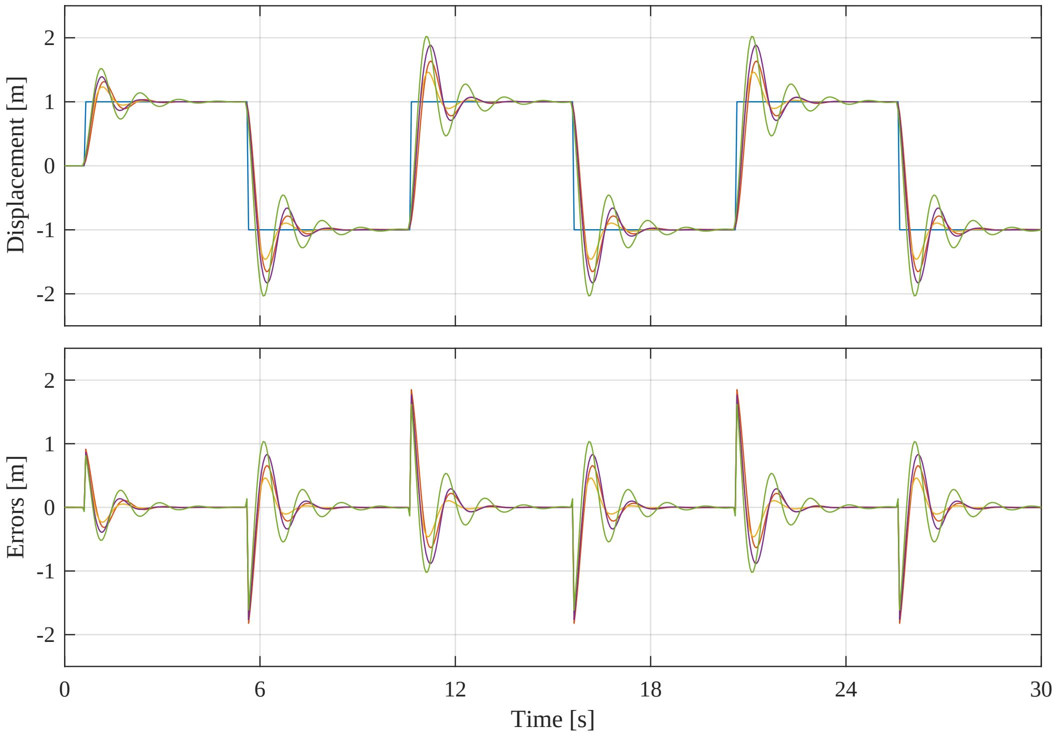

Figure 3.27: Reference tracking: Square wave [TS16a]

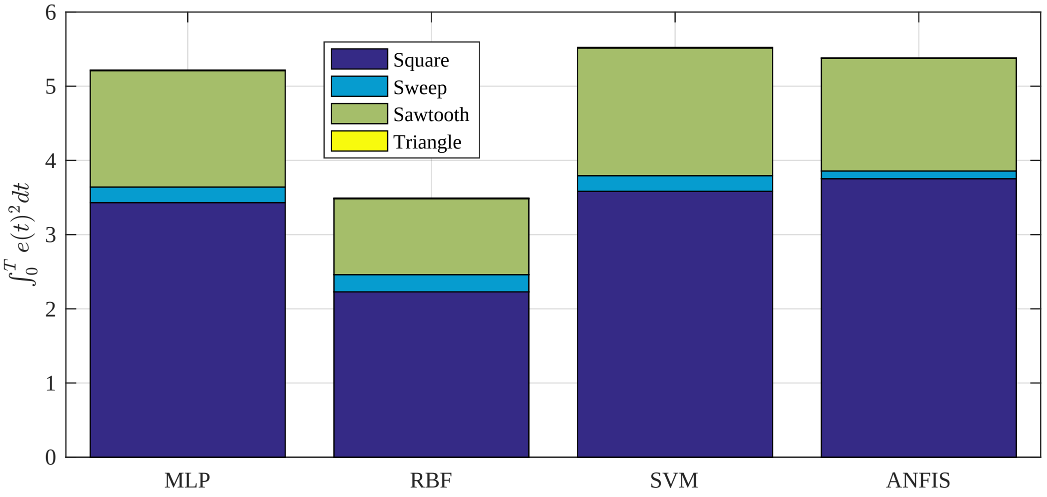

Figure 3.28: Performance: Tracking error for reference tracking [TS16a]

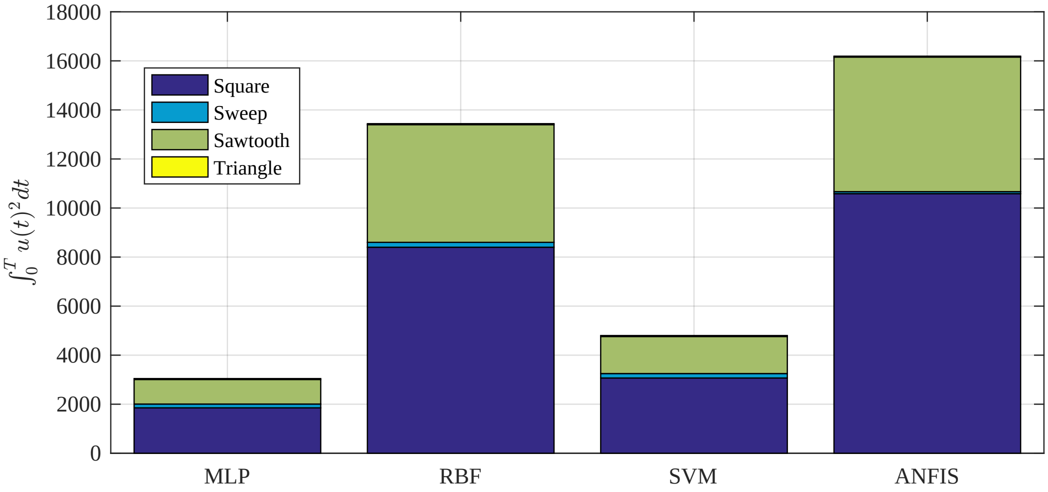

Figure 3.29: Performance: Control effort for reference tracking [TS16a]

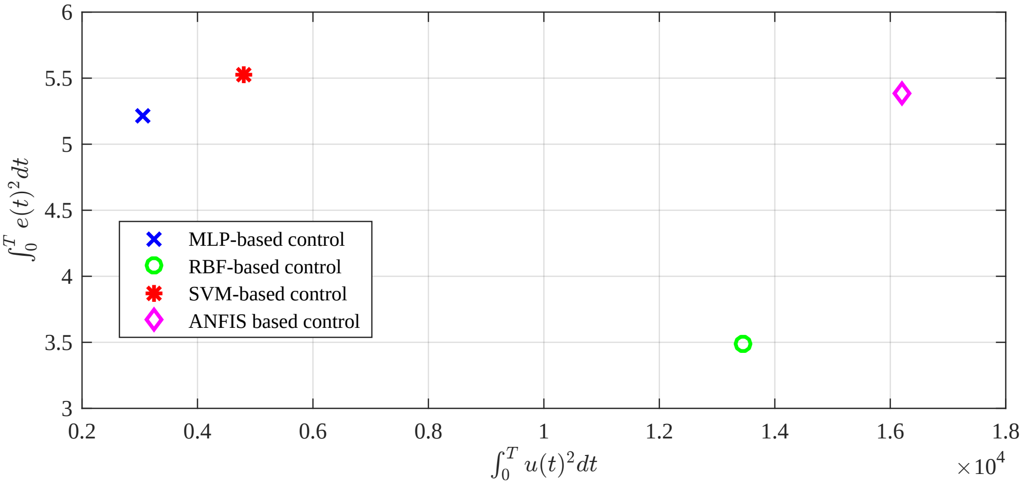

Figure 3.30: Performace [TS16a]

3.4 Summary

This chapter was divided into three sections, In section 3.1 the benchmark used to evaluating the performance of soft computing algoirthms was given. Additionally the training and test signals were provided.

In section 3.2 the training results for modeling the benchmark were provided. From the training results it can be observed that the computational requirements show strong deviations due to the different training methods. It is observed that the MLP requires the most training time; on the other hand ANFIS has the shortest training time. From the comparison of the results it can be stated the the algorithms have good generalization results. From the results obtained it can be noted that the SVM shows the best generalization results. On the other hand, the computational load required for each algorithm to estimate the system output, have wide variation. It is noted that the MLP and RBF have the least computation load requirement while the GP has the most computational load requirements. The GP model has a second output which provides indication about the quality of the estimation; this is a unique characteristic of the GP. Based on the results obtained decisions about the choice of modeling/prediction methods can be realized on a qualified base.

In section 3.3 the results for obtaining a direct inverse model controller based on soft computing for the benchmark were provided. From the training results it can concluded that the SVM has fast learning algorithm. Based on the performance results all the algorithms are able a relatively good performance controllers. The SVM-based inverse model controller has relatively the best performance, where the controller has fast response time, acceptable percent overshoot, near zeros steady-state error and has agreeable control effort.

4 Soft computing-based controller for complex mechanical system

In this chapter the soft computing-based controller for complex mechanical system is presented. The effect of data-set size on the SVM-based observer is briefly described.Parts from this chapter are based on text and material that are previously published [TS14].

4.1 Soft computing-based controller for complex mechanical system

In this work, a novel soft computing-based controller concept is developed. The main aim of the concept is to enhance the performance of multi-input and multi-output (MIMO) system control. The structure of the introduced soft computing-based control of complex mechanical system is shown in figure 4.1. The structure consists of two modules; a modeling module and a control module. The modeling module has two functions; the first function is to capture the unknown dynamics of the targeted system $\dot{x} = f(x, u)$. The resulting model is then used in the control module. The second function is to build an observer that estimates the internal states of the unknown system using its outputs. This observer is applied directly in the control loop. The control module is responsible of generating suitable control law using the model acquired by the modeling module so that the system follows the desired trajectory. Once the control law is acquired it can be applied to the control loop thus closing the control loop. It is assumed, that the system is stable, fully controllable, and fully observable. Additionally, it is assumed that the internal states are measured for the time period T allowing the modeling module to build both the observer and dynamic model.

The modeling module is based on SVM algorithm which is based on convex optimization so the resulting solution is always global solution [WY04]. Additionally, the algorithm has less tuning parameters in comparison with multilayer perceptron network. The adaptation of support vector regression for dynamic system is given in [WY04]. The training of the dynamic model $\dot{x} = f(x, u)$ is achieved according to the Nonlinear Output Error (NOE) model described in [NRPH00]. The model uses the system input and state signals to produce multi-step ahead prediction. For the training of the observer it is achieved using autoregresive model with external input (ARX) model which as also was described in [NRPH00]. A classical observer uses input and output signals to estimate the internal state of the system $\dot{\hat{x}} = g(u, y)$. The soft computing-based observer is based on empirical data, a model that uses only the output of the system to estimate the internal states of the system $\dot{\hat{x}} = g(y)$ is of interest thus the complexity of the model is reduced.

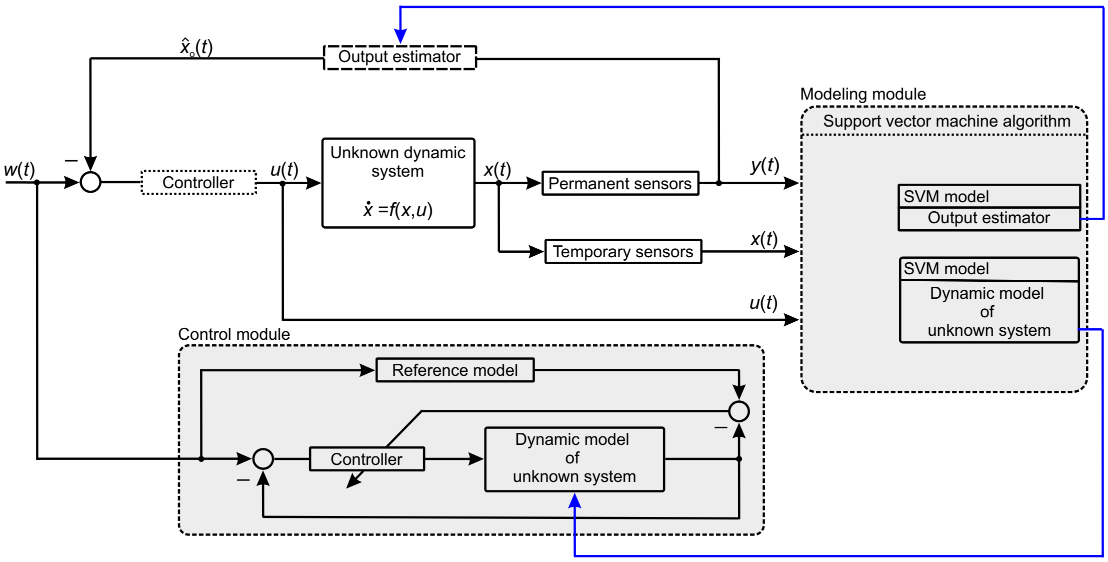

Figure 4.1: Structure of the soft computing-based control of complex system [TS14]

The control module is also based on SVM algorithm. The initial control law is acquired using the direct inverse model method. In this method the controller is trained to have the inverse dynamic of the dynamic system $u = f^{-1}(x, u)$. Later on the performance of the controller is modified according to the desired behavior described by the reference model. The performance of the controller is verified using the dynamic model obtained using the modeling module. Once the desired performance is satisfied the controller is applied to the real dynamic system.

4.2 The effect of data-set size on dynamic systems model based on SVM algorithm

A major problem in machine learning field is the size of the data-set used for training and its effect on the quality of the resulting model. This problem is refereed to as the bias-variance dilemma [Kod14, SW11, TK08, Kec01, Nel01]. This problem also affect the SVM-based dynamic model. In this section the relation between the data-set size and the accuracy of the SVM-based modeling/control is explained.

The training is done on a finite training dataset D. The performance of the model is evaluated using the Sum of Squared Errors (SSE), which is used as loss function.

As noted in [Nel01] the estimation error loss function is

$$\begin{aligned} E\{e^2\} &= E\{(y-\hat{y})^2\} \\ &= E\{((y_u+\varepsilon)-\hat{y})^2\} \\ &= E\{(y_u-\hat{y})^2\} + E\{(y_u-\hat{y})\varepsilon\} + E\{\varepsilon^2\}\}, \end{aligned} \tag{4.1}$$

where ε denotes the noise and y_u the measured system output corresponding to the system input u.

The term $E(y_u - \hat{y})^2$ represents the error between the measured system output $\hat{y}$ and the targeted the measured system output y_u. The term $E\{(y_u - \hat{y})\varepsilon\}$ vanishes if the noise is uncorrelated with the process and the measured system output [Nel01]. The term $E\{\varepsilon^2\}$ represents the noise variance which is not influenced by the model. As given in [Nel01] the optimal solution is obtained when $\hat{y} = y_u$.

According to [Nel01] the term $E(y_u - \hat{y})^2$ can be further decomposed as

$$\begin{aligned} E\{(y_u-\hat{y})^2\} &= E\{[y_u - E\{\hat{y}\} - (\hat{y} - E\{\hat{y}\})]^2\} \\ &= E\{[y_u - E\{\hat{y}\}]^2\} + E\{[\hat{y} - E\{\hat{y}\}]^2\} \\ &= [y_u - E\{\hat{y}\}]^2 + E\{[\hat{y} - E\{\hat{y}\}]^2\}. \end{aligned} \tag{4.2}$$

The first term $[y_u - E\{\hat{y}\}]^2$ is the bias error, the second term $E\{[\hat{y} - E\{\hat{y}\}]^2$ is the variance error. The bias represents the accuracy of the SVM-based model over different datasets and the variance represents the sensitivity of the SVM-based model over different datasets. According to [TK08], a trade-off between the two terms occurs. Bias increase leads to a decrease of the variance and vice versa. This problem is solved methods making both of variables low at the same time. This can be achieved by increasing the number of N asymptotically to infinity [TK08]. In practice N is finite and the solution is the one with the best compromise. One of the most widely used methods to deal with bias-variance dilemma is cross-validation.

4.3 Summary

In this chapter, the concept of the controller for complex mechanical system has been presented in section 4.1. The main goal of the contrive controller is to enhance the performance of multi-input and multi-output (MIMO) system. Here two main modules with their functions are described. Later on in section 4.2 the effect of data-set size on SVM-based dynamic systems model has been discussed. The bias-variance dilemma has been repeated. The cross-validation approach has been selected to overcome the bias-variance dilemma.

5 Verification of the soft computing-based controller

In this chapter verification results of the soft computing-based controller for complex mechanical system is presented. Simulation benchmark and training/test signals and are given. The benchmark example is an elastic cantilever beam modeled using finite element method. The verification results of the state variables estimator base on SVM algorithm are provided. Verification results of the elastic structure dynamic model based on SVM algorithm are given. Finally the SVM-based inverse model controller for the elastic cantilever beam is presented. Parts from this chapter are based on text and material that are previously published [TS16b, TS15b, TS15a].

5.1 Elastic cantilever beam model

The equation of motion of the beam system is derived using Finite Element Method (FEM). Five finite elements as shown in figure 5.1 are assumed to model the beam discussed. The beam parameters are given in table 5.1. According to results in table 5.2 using five element model provides a sufficient dynamic accuracy and simple computational power. This can also be concluded from table 5.2, here a comparison between natural frequencies of the cantilever beam using analytical method and FEM is given. It can be noted that the difference between the analytical solution and the FEM-based solution is small. Therefore it can be concluded that the FEM model provides the desired accuracy. The global equation of motion is given by

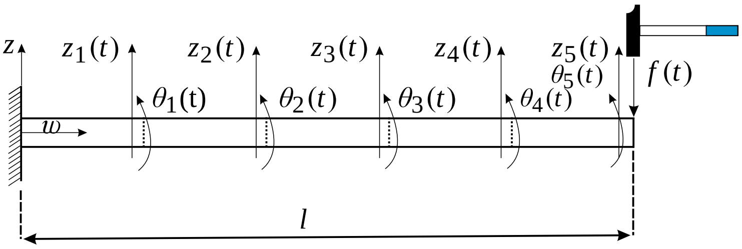

Figure 5.1: Cantilever beam (5 elements)[TS16b]

$$M\ddot{z} + C\dot{z} + Kz = F(t), \tag{5.1}$$

where M denotes the global mass matrix, C the global damping matrix, K the global stiffness matrix, F is the dynamic load vector, and z is the generalized global coordinate vector used to describe the dynamic behavior of the nodes of the cantilever beam.

Table 5.1: Beam parameters [TS16b]

Density	8050 Kg/m^3
Length	546 mm
Cross section area	30×5 mm^2
Modulus of elasticity	200×10^9 N/m^2
ς	5×10^{-4}
ν	1×10^{-4}

Table 5.2: Comparison between analytical solution and FEM-based numerical solution of cantilever beam considered [TS16b]

Mode	Analytical solution [Hz]	FEM solution [Hz]	Error [%]
1	2.638	2.653	0.566
2	16.597	16.634	0.225
3	46.497	46.672	0.4811
4	91.134	92.296	1.275
5	150.803	153.18	1.581

Each element is modeled using two has two degrees of freedom of the node, the displacement $z_{1...5}$ and the bending $\theta_{1...5}$, the vector describing the mechanical coordinates of the system z is given as

$$z = \begin{bmatrix} z_1 & \theta_1 & z_2 & \theta_2 & z_3 & \theta_3 & z_4 & \theta_4 & z_5 & \theta_5 \end{bmatrix}^T . \tag{5.2}$$

According to [KB00], the local mass matrix M_e is given by

$$\begin{aligned} M_e = \frac{\rho A l_e}{420} &\begin{bmatrix} 156 & 22l_e & 54 & -13l_e \\ 22l_e & 4l_e^2 & 13l_e & -3l_e^2 \\ 54 & 13l_e & 156 & -22l_e \\ -13 & -3l_e^2 & 22l_e & 4l_e^2 \end{bmatrix} \begin{bmatrix} \ddot{z}_i \\ \ddot{\theta}_i \\ \ddot{z}_{i+1} \\ \ddot{\theta}_{i+1} \end{bmatrix} \\ = \frac{\rho A l_e}{420} &\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} \ddot{z}_i \\ \ddot{\theta}_i \\ \ddot{z}_{i+1} \\ \ddot{\theta}_{i+1} \end{bmatrix} , \end{aligned} \tag{5.3}$$

and, the local stiffness matrix k_e is given by

$$\begin{aligned} k_e = \frac{EI}{l_e^3} &\begin{bmatrix} 12 & 6l_e & 12 & 6l_e \\ 6l_e & 4l_e^2 & -6l_e & 2l_e^2 \\ -12 & -6l_e & 12 & -6l_e \\ 6l_e & 2l_e^2 & -6l_e & 4l_e^2 \end{bmatrix} \begin{bmatrix} z_i \\ \theta_i \\ z_{i+1} \\ \theta_{i+1} \end{bmatrix} \\ = \frac{EI}{l_e^3} &\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} z_i \\ \theta_i \\ z_{i+1} \\ \theta_{i+1} \end{bmatrix} , \end{aligned} \tag{5.4}$$

where ρ is the density, l_e the element length, A the element cross section area, E modulus of elasticity, I area moment of inertia. whereby $\square_i$ describes the variables of the left side of the finite elements and $\square_{i+1}$ those of the right side.

The global mass, stiffness matrices, and applied forces vector are obtained by adding the local corresponding matrices in a diagonal way and applying the boundary conditions. The global mass matrix M is given in 5.5, while the global stiffness matrix K is in 5.6. The global applied forces F is given as

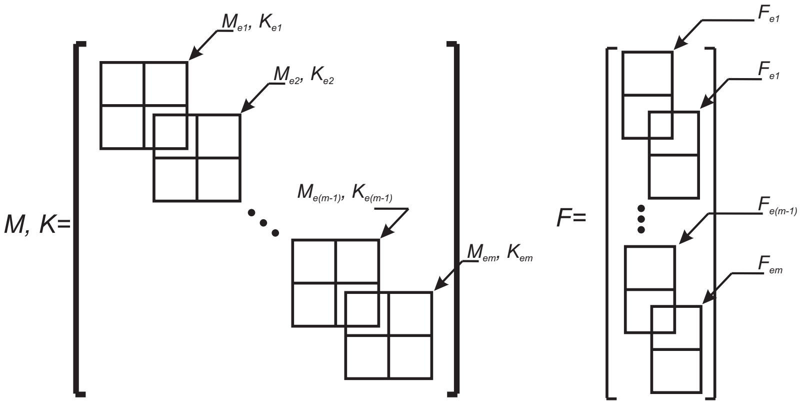

Figure 5.2: Global system matrices

$$M = \frac{\rho A l}{420}\begin{bmatrix} m_{33}+m_{11} & m_{34}+m_{12} & m_{13} & m_{14} & 0 & 0 & 0 & 0 & 0 & 0 \\ m_{43}+m_{21} & m_{44}+m_{22} & m_{23} & m_{24} & 0 & 0 & 0 & 0 & 0 & 0 \\ m_{31} & m_{32} & m_{33}+m_{11} & m_{34}+m_{12} & m_{13} & m_{14} & 0 & 0 & 0 & 0 \\ m_{41} & m_{42} & m_{43}+m_{21} & m_{44}+m_{22} & m_{23} & m_{24} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{31} & m_{32} & m_{33}+m_{11} & m_{34}+m_{12} & m_{13} & m_{14} & 0 & 0 \\ 0 & 0 & m_{41} & m_{42} & m_{43}+m_{21} & m_{44}+m_{22} & m_{23} & m_{24} & 0 & 0 \\ 0 & 0 & 0 & 0 & m_{31} & m_{32} & m_{33}+m_{11} & m_{34}+m_{12} & m_{13} & m_{14} \\ 0 & 0 & 0 & 0 & m_{41} & m_{42} & m_{43}+m_{21} & m_{44}+m_{22} & m_{23} & m_{24} \\ 0 & 0 & 0 & 0 & 0 & 0 & m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 0 & 0 & 0 & m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \tag{5.5}$$

$$K = \frac{EI}{l^3}\begin{bmatrix} k_{33}+k_{11} & k_{34}+k_{12} & k_{13} & k_{14} & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{43}+k_{21} & k_{44}+k_{22} & k_{23} & k_{24} & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{31} & k_{32} & k_{33}+k_{11} & k_{34}+k_{12} & k_{13} & k_{14} & 0 & 0 & 0 & 0 \\ k_{41} & k_{42} & k_{43}+k_{21} & k_{44}+k_{22} & k_{23} & k_{24} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{31} & k_{32} & k_{33}+k_{11} & k_{34}+k_{12} & k_{13} & k_{14} & 0 & 0 \\ 0 & 0 & k_{41} & k_{42} & k_{43}+k_{21} & k_{44}+k_{22} & k_{23} & k_{24} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{31} & k_{32} & k_{33}+k_{11} & k_{34}+k_{12} & k_{13} & k_{14} \\ 0 & 0 & 0 & 0 & k_{41} & k_{42} & k_{43}+k_{21} & k_{44}+k_{22} & k_{23} & k_{24} \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{31} & k_{32} & k_{33} & k_{34} \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \tag{5.6}$$

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T u(t). \tag{5.7}$$

According to [KB00], the analytical assignment of the damping matrix C is difficult. So, for simplification, Ralyleigh damping or proportional damping is used as follows

$$C = \varsigma[M] + \nu[K], \tag{5.8}$$

where ς and ν are constants as given in [KB00].

5.1.1 System output

The main goal of observers is to estimate the system state variables based on available inputs and outputs. To demonstrate the ability of the SVM-based observer two different relationships between system output and system states are considered. First, a direct relationship is used, when the output is equal one of the state variables $y = z_i$. Second, an indirect relationship when the relationship is a unknown function of one of the state variables $y = f(z_i)$.

5.1.1.1 Direct relationship

For this example, the tip (end effector) position of the beam is assumed to be measurable as system output is given as

$$y = z_5.$$

5.1.1.2 Indirect relationship

Here the measurement is based as a (unknown) function of beam strain ϵ is derived according Euler-Bernoulli theory as

$$y = \epsilon(w,t) = -c\beta_n^2 Re\Bigg[B\Big[\cosh(\beta_n w) - \cos(\beta_n w) + \sinh(\beta_n w) - \sin(\beta_n w)\Big]e^{-i\omega t}\Bigg], \tag{5.9}$$

where β_n and B are constants depending on the mode shape of cantilever beam as

$$\beta_n \approx \frac{(2n-1)\pi}{2}$$

and

$$B = \frac{\sinh(\beta_n l) + \sin(\beta_n l)}{\cosh(\beta_n l) + cos(\beta l)},$$

here w describes the position of stain along the beam.

5.1.2 Excitation signals

The identification of unknown system dynamics is done based on measured input/output signals. It is required to use input signals that excite the whole dynamics of the targeted system. The cantilever system is excited by applying a force f at the tip of the beam as shown in figure 5.1. To achieve a SVM-based observer with high performance the dataset should be divided into at least two parts 4.2. Here three data sets as shown in figure 5.3 are used, one used for training, the other two are used for evaluation of SVM-based observer.

In this work, three excitation signals (figure 5.3) are used for training and evaluating SVM-based observer: chirp (sinusoidal signal with a fixed amplitude and frequency range that varies over time) as shown in figure 5.3.a, pseudo-random signal (random amplitude and random width) as shown in figure 5.3.b, and the expanding sweep signal (both the amplitude and frequency vary over time) as shown in figure 5.3.c. The main goal of the excitation signals is to observe different dynamical behaviors of the beam and related different effects to the estimation procedure due to the different excitation signals. In the simulations and experiments the beam displacement (z and θ) and strain (ϵ) signals are collected.

5.2 Verification of the soft computing-based controller

In this section, the verification of the soft computing-based controller using SVM algorithm is discussed. First the SVM-based observer is presented and the verification results are given. Then the SVM-based dynamic system model and the automated dynamic systems identification based on SVM algorithm are introduced. Finally, the SVM-based direct inverse model control strategy is provided.

5.2.1 Verification of the SVM-based observer

State variables estimation using soft computing algorithms

A state observer is a mathematical model that uses the measured inputs and outputs of a dynamic system to estimate state variables based on a given dynamic model that internally relates inputs, states, and outputs. The accurate estimation of state variables of a dynamic system is important to achieve the desired close-loop control goals.

The performance of the observer depends on accurate modeling of the system. In complex modern systems, the process of deriving an accurate mathematical model is based on physical relations or descriptions that require a number of assumptions fixing the resulting model to a certain working point. So the accuracy of the observer

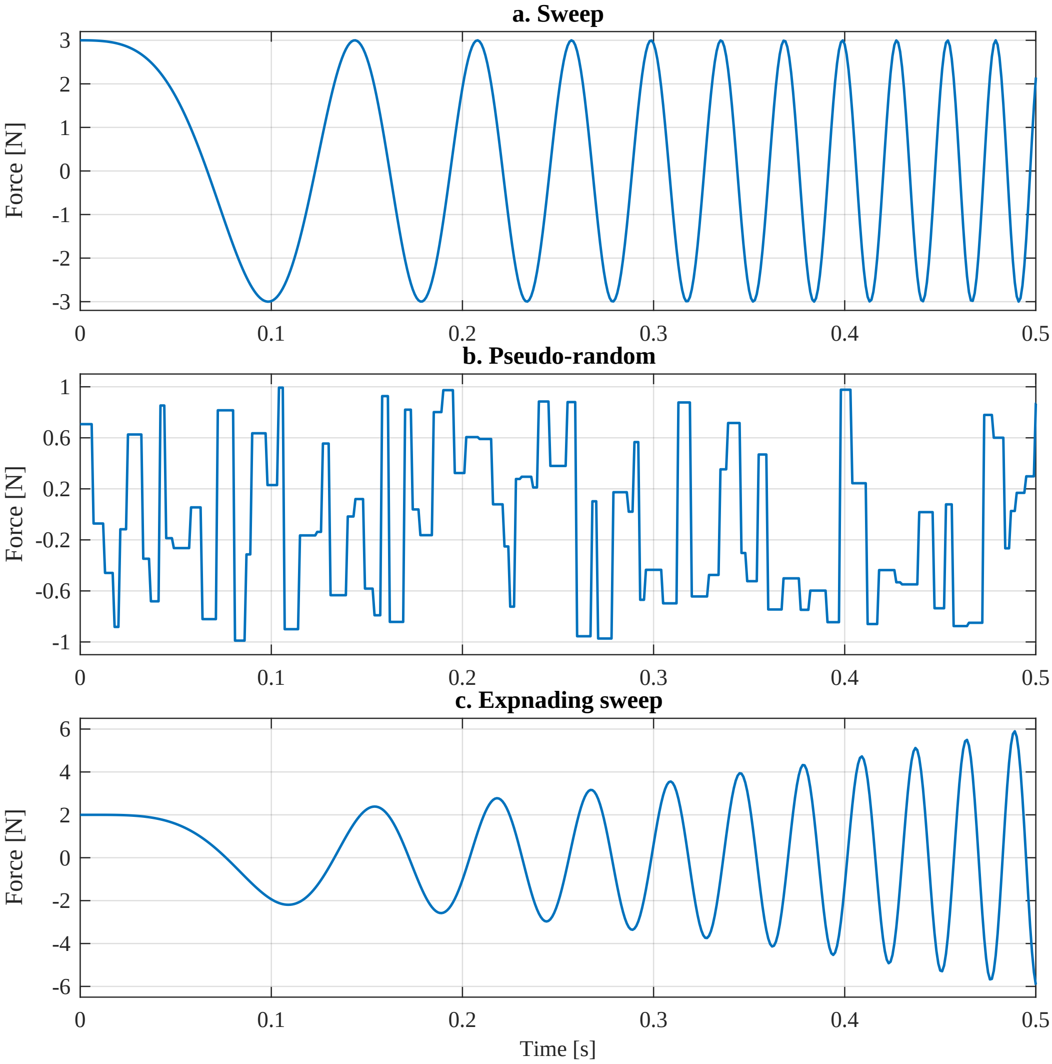

Figure 5.3: Excitation force signal [TS16b]

is degraded when the system is operating at different working points. A number of high performance observers have been proposed in the literature to overcome this challenge. Several approaches that have been proposed to enhance the performance of nonlinear observers. These approaches were summarized in [PSY01] as follows

- In [KI83], a nonlinear observer based on the error linearizion techniques is proposed. This nonlinear observer is based on the Lie algebra method. Unfortunately, no application is provided to demonstrate the efficiency of the proposed method.

- Using high gains as noted in [Tor89], where it is illustrated that the use of high gain induces a time scale separation between the nonlinear system and the observer. Consequently, a singular perturbation theory is applied for the stability analysis of the dynamic error. In [Tor89], it is shown that the error dynamics reached stable equilibrium very fast, ensuring that the slow dynamics of the observer is that of the given nonlinear system.

- As proposed [MM95], using a moving horizon optimization the state variables are estimated by minimizing a cost function over the preceding interval. No demonstration results are given.
- Using reduced-order technique as given in [GD95], where a nonlinear observer of reduced order, the goal of the observer is to estimate the system's state variables used in a trajectory tracking control loop. The technique is applied to estimate the position of scara type robot arm based on simulation data and to a non-affine second order dynamic system. In both simulation examples the observer has demonstrated high accurate tracking of the state variables system.

It is noted in [KP14] that high gain observers degraded of the noise rejection ability when the gain is set too high. Furthermore the occurrence of peaking responses in the absence of control saturation leads to non-optimal results. According to [PSY01] the above approaches deal with problems in which the system model has no uncertainties, however in practice the observer is working in the presence of noise and disturbances and unmodeled dynamics influencing the performance. To overcome this problem, machine learning techniques are integrated into control engineering. This integration allows the replacement of theoretical-based modeling by empirical-based modeling.

Soft computing-based observers

In the following is a brief summary of the soft computing-based observers that were developed in the recent years. These observers are based on the multilayer perceptrons networks, radial bases function and fuzzy inference system.

In [HS96], neuro-observer-based controller is proposed. The neuro-observer is constructed using a single-layered Radial Basis Function Network (RBFN) to estimate the state variables of a nonlinear dynamical system. For fast training convergence, a random reference input is employed. The simulation results show that the proposed observer is able to estimate the dynamic system state variables with acceptable accuracy. A robust asymptotic neuro-observer for a class of unknown nonlinear systems with noise disturbances on the output is proposed in [PSPW00]. By introducing a dynamic neural network of extended Luenberger structure with additional time delay compensation term, the robust asymptotic neuro-observer is realized. A dead-zone multiplier is added into the learning procedure to maintain the weight matrices of the dynamic neural network within boundaries. The simulation results show a good estimation performance for the systems state variables. The stability of the estimation error maintains within a given class of uncertainties.

Following the principle ideas given in [Poz98], a robust asymptotic neuro-observer is proposed in [YL14]. To design this observer, a dynamic neural network of the extended Luenberger structure is applied in [PSPW00]. Additionally, an updating law for the weights of the multilayer dynamic neural networks that specifically selects the weights to guarantee the robustness of the observer is proposed. It should be noted that the used dynamic neural network has an additional time delay compensation term. Similarly, a neuro-observer for a micro-cantilever beam is realized in [QGL+15] by combining a RBF neural network with available knowledge about the micro-cantilever beam following the guidelines provided in [Sch00]. The estimated states using this observer are used to build a general regression neural network model that describes the nonlinear behavior of the micro-cantilever beam. The simulation results demonstrate reasonable accuracy of the proposed method.

An Elman Neural Network (ENN)-based observer is proposed for the nonlinear feedback control of a mobile robot in [AMAH+15]. The ENN-based observer is used to overcome the uncertainty of the model, as well as estimating the time derivative of the mobile robot coordinates. The simulation results show that ENN-based observer can realize accurate estimations. A neural network-based robust unknown input observer for wind turbines is proposed in [WPW+15]. This observer is achieved first by training neural network to mimic non-linear system behavior. Once the weights of the neurons are obtained a constrained set representation is build using the sector nonlinear concept and finally multi-variable membership function is found.

In [WR16], a neural network observer-based optimal tracking control for multi-motor servomechanism is proposed. In this approach a neural network is combined with linear optimal observer to estimate the state variables of multi-motor servomechanism system. The simulation results show good tracking behavior, but no results are provided about the accuracy of the state variables estimation. In [WCLS16] an observer-based adaptive neural controller is proposed for a class of nonlinear pure-feedback systems. The observer is based on RBFN to estimate the systems state variables. Subsequently, adaptive neural control method and back-stepping method are joined to construct the control design scheme. The simulation results of observer-based adaptive neural controller on third order nonlinear electromechanical system show that the steady state errors of all the estimated systems state variables convert to zero.

A neuro-augmented observer is presented in [GXC06]. The neuro-augmented observer has two main parts, linear part and nonlinear part. The linear part is based on a conventional linear observer design, and the nonlinear part is realized using RBFN to approximate the nonlinearity of the dynamic system. This approach provides full access to the state variables for the system. The simulation results of the approach proposed show that the state variables estimation error approaches zero asymptotically. A neuro-sliding mode observer to state variables reconstruction of a quadcopter is proposed in [BB11]. The neuro-sliding observer has the same structure as a simple sliding mode observer. The main goal is to reduce the observer sensitivity to noise. To realize the neuro-sliding mode observe two parallel feed-forward artificial neural networks are used, the first network estimates on-line the equivalent control term and the second generates observers feedback. The learning law for the network is based on Lyapunov stability theory. Simulation results show that the measurement noise is suppressed without any performance degradation.

An Adaptive Neural Control (ANC) approach using a non-separation principle is proposed in [PSY15]. In this approach a hybrid estimation scheme integrating an adaptive NN observer with state variable filters is presented. The simulation results are done on a mass-spring-damper system. The results show that due to the low-gain feature peaking responses without control saturation can be avoided. Additionally, in comparison with the work published in [PPKM05] can be achieved. In [LF15] an adaptive backstepping control approach of Micro-Electro-Mechanical Systems (MEMS) of a gyroscope based on neural state observer is proposed. This control approach employs an RBFN-based observer to estimate the MEMS gyroscope states incorporated in the backstepping controller. The RBFN is used to estimate the nonlinear part of the gyroscope system model. The adaptation of the observer is investigated based on the Lyapunov stability framework in order to guarantee the accuracy of the observer. The simulation results demonstrate the accuracy and effectiveness of this approach.

In [PYHP06] an Adaptive Neuro Fuzzy Inference System (ANFIS) with backpropagation learning algorithm is used to estimate the rotor flux and rotation speed. The simulation results show that the ANFIS observer has a good estimation of the flux and motor speed variables. In [NK15] Coactive Adaptive Neuro-Fuzzy (CANFIS) Observer for fault detection of the three-tank interacting level process is proposed. A Coactive Adaptive Neuro-Fuzzy is a generalized ANFIS where both NN and FIS actively cooperate to achieve the desired goal. The simulation results illustrate reasonable estimations.

An adaptive fuzzy output feedback control approach for a class of MIMO uncertain stochastic nonlinear strict-feedback systems is presented in [LTL12]. The approximation of the unknown linear function is done using fuzzy logic systems. A fuzzy state observer is designed for estimating unmeasured states. It is demonstrated that by an appropriate choice of the design parameters, the observer errors of the system converge to a small neighborhood of the origin. In [GM12] Adaptive Fuzzy Sliding Mode (AFSM) observer which can be used for a class of MIMO nonlinear systems, is proposed. The observer is designed to estimate the nonlinear behavior of a system by adapting fuzzy rules based on the observer error. The main advantage of the

AFSM observer is that outputs are not limited to the first entries of a canonical-form state vector. The AFSM observer shows high effectiveness when it was applied to a real robot manipulator with multiple links.

In [Bey15], an Adaptive Fuzzy Terminal Sliding-Mode (AFT-SM) observer is introduced. The AFTSM-observer is considered as an improvement of the conventional gradient-decent-based adaptive fuzzy observer. This improvement is achieved by adding a switching term of the sliding-mode approach to the state of the observer, and also by designing the measurement error of the system as input of the observer instead of the system's output. The AFTSM observer is applied successfully to two experiments, flexible joint manipulator and flexible-link transmission system. In both experiments, it is able to estimate the systems state variables with sufficient accuracy.

The literature review shows that most of the research work is based on employing machine learning side by side with mathematical modeling of the dynamic system to design an observer for nonlinear dynamic systems. The mathematical modeling part is primarily responsible for the estimation of the linear system state variables. The function of the machine learning part is the estimation unmodeled part or nonlinear parts of the dynamics. The requirement of a detailed understanding of the dynamic system mathematical-based observers in order to integrate the machine learning part without contradiction with mathematical-based observers, this appears as drawback. In [PSY15] it is noted that existing adaptive neural network observers are not able to effectively reduce the limitations of high-gain observers. The performance of the SVM-based observer is evaluated using system's input and output vectors. A second example is proposed where the system's state variables are estimated using the system output only, this demonstrate the ability of the introduced approach. So in the second example the strain signals are used to estimate the beam deflection.

5.2.1.1 SVM-based observer using the system's input and output vectors

Using the tip displacement z_5 as system's out and the force u as system's input, the estimation of the desired systems state variables (z_1 to z_4) and (θ_1 to θ_5) using SVM-based observer is realized using

$$\hat{x}(k+1) = g(u(k), u(k-1), ..., u(k-p), \\ y(k), y(k-1), ..., y(k-p)). \quad (5.10)$$

The training of the SVM-based observer is done using pseudo random signal shown in figure 5.3.b. The estimation results are obtained using the sweep signal shown in figures 5.4 and 5.5. It is observed that the observer reduces the estimation error in a relatively short time and produces an accurate estimation of the system state variables. To illustrate the efficiency of the SVM-based observer, the results are compared with those using a Kalman-Bucy observer as shown in figures 5.6 and 5.7.

Table 5.3: Sum of squared error of the angle estimations [TS16b]

	SVM-based observer	Kalman-Bucy observer
SSE_{z1}	5.61×10^{-7}	8.16×10^{-7}
SSE_{z2}	3.49×10^{-6}	5.22×10^{-6}
SSE_{z3}	6.11×10^{-6}	1.44×10^{-5}
SSE_{z4}	6.40×10^{-6}	3.27×10^{-5}

Table 5.4: Sum of squared error of the angle estimations [TS16b]

	SVM-based observer	Kalman-Bucy observer
$SSE_{\theta 1}$	1.17×10^{-4}	1.61×10^{-4}
$SSE_{\theta 2}$	9.81×10^{-5}	2.70×10^{-4}
$SSE_{\theta 3}$	6.92×10^{-5}	4.07×10^{-4}
$SSE_{\theta 4}$	1.59×10^{-4}	6.67×10^{-4}
$SSE_{\theta 5}$	2.54×10^{-5}	7.52×10^{-4}

In order to achieve numeral comparison between the SVM-based observer and the Kalman-Bucy observer the Sum of Squared Errors (SSE) is used. The SSE is given as

$$SSE = \sum_{i=1}^{N} (z(i) - \hat{z}(i))^2, \quad (5.11)$$

where N is the number of sample in the dataset.

In tables 5.3 and 5.4, the SSE of both observers is presented. It can be noted that the SVM-based observer can achieve a similar performance to Kalman-Bucy observer.

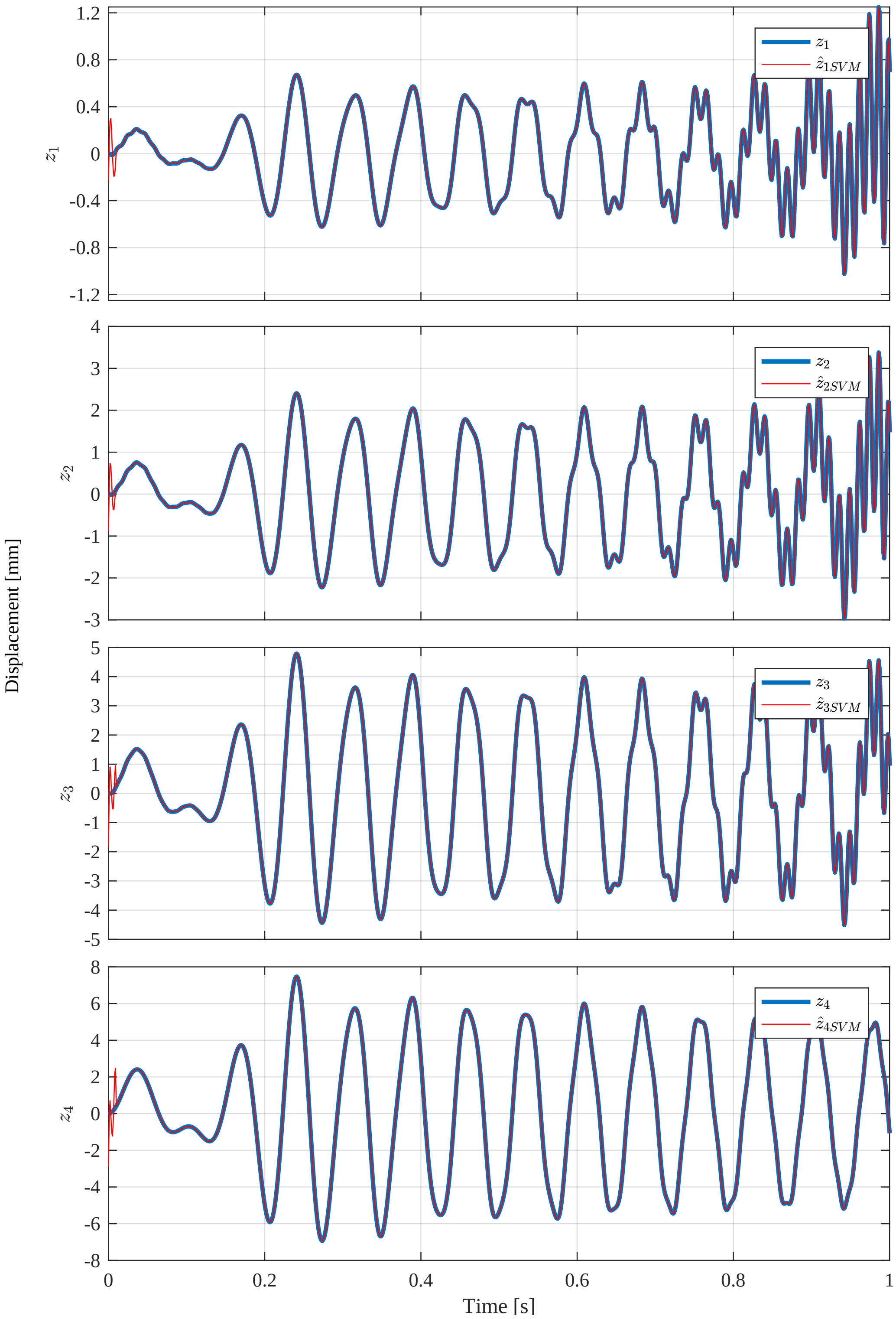

Figure 5.4: Estimation of beam displacement [TS16b]

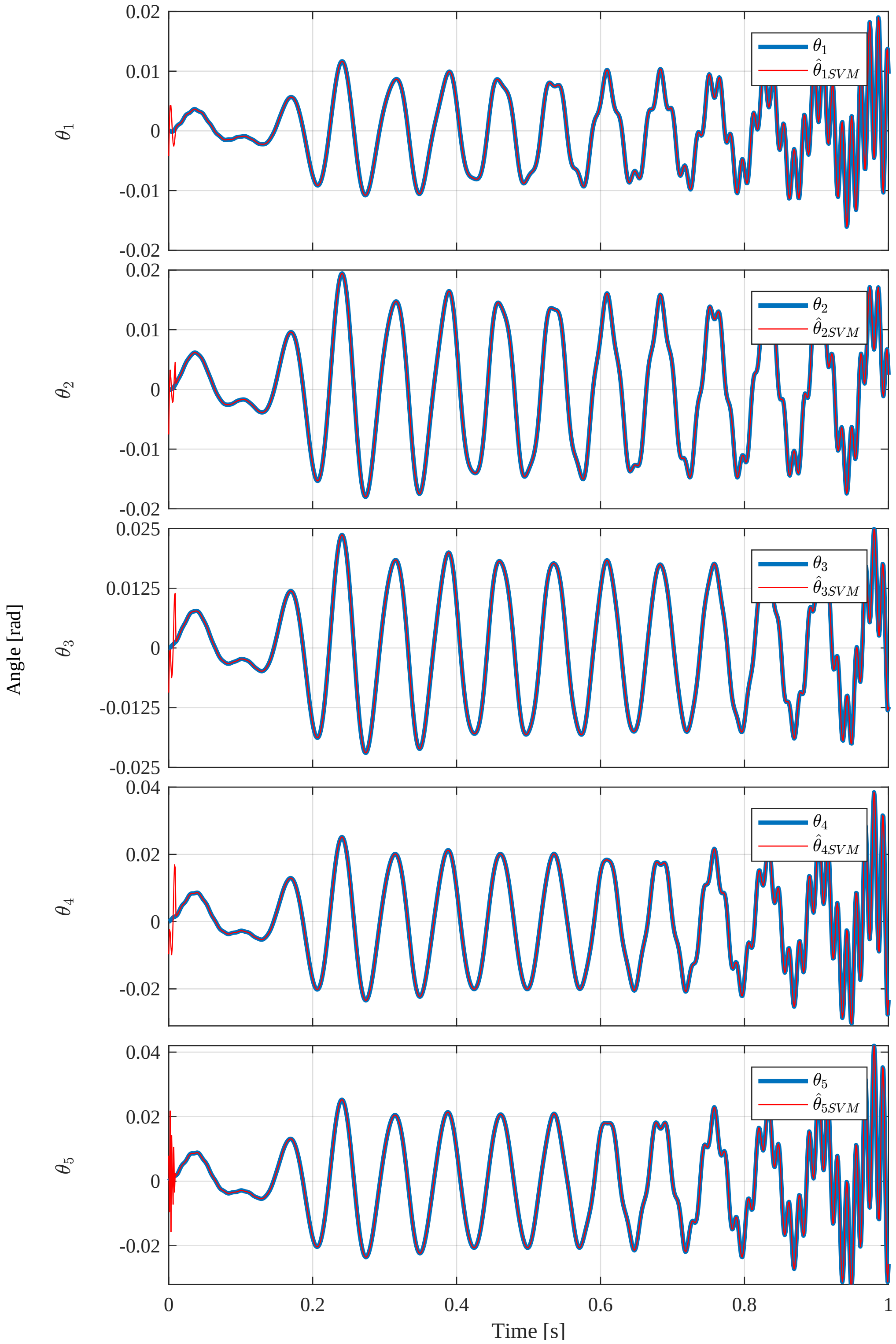

Figure 5.5: Estimation of the beam angle [TS16b]

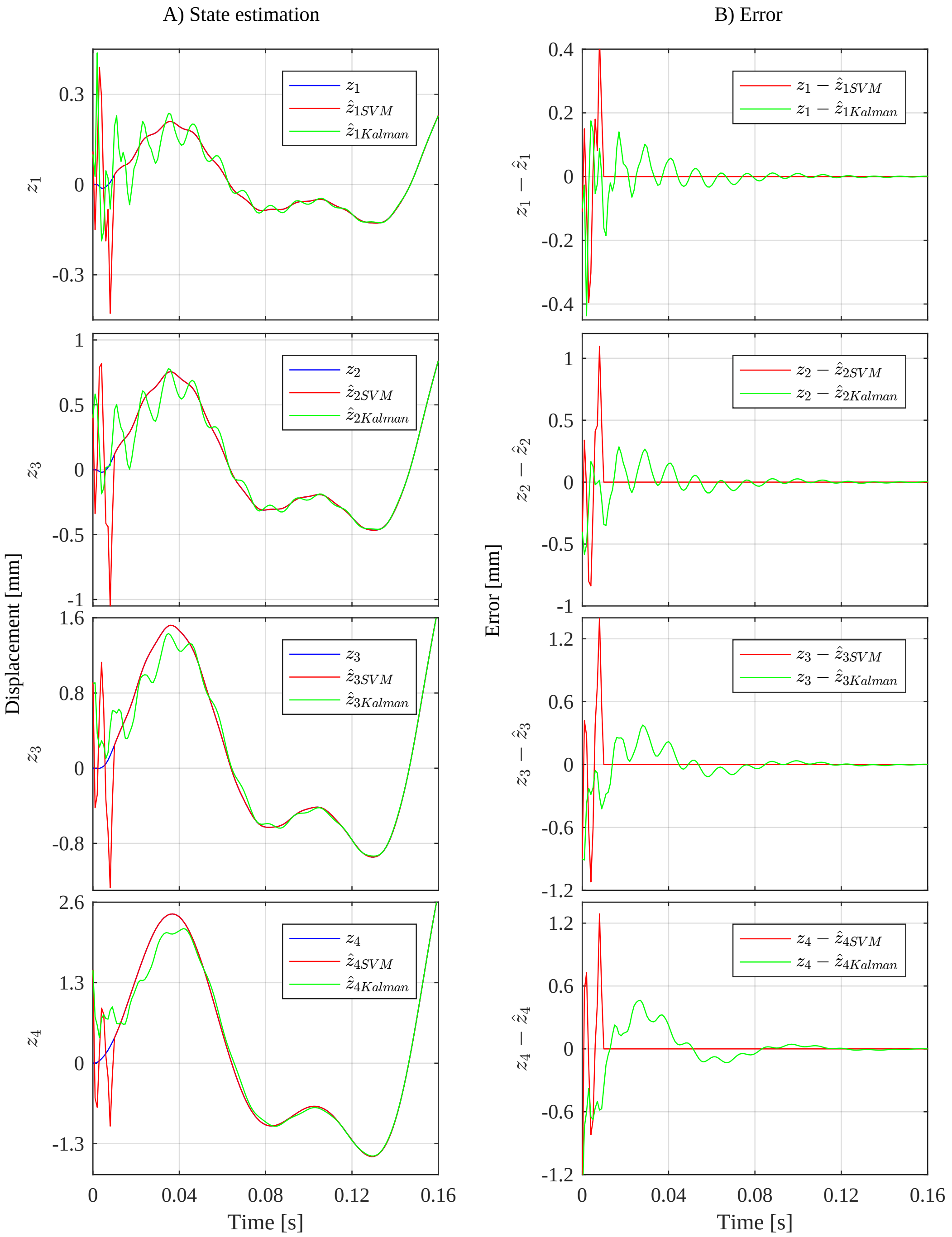

Figure 5.6: Displacement estimation (Zoom in)[TS16b]

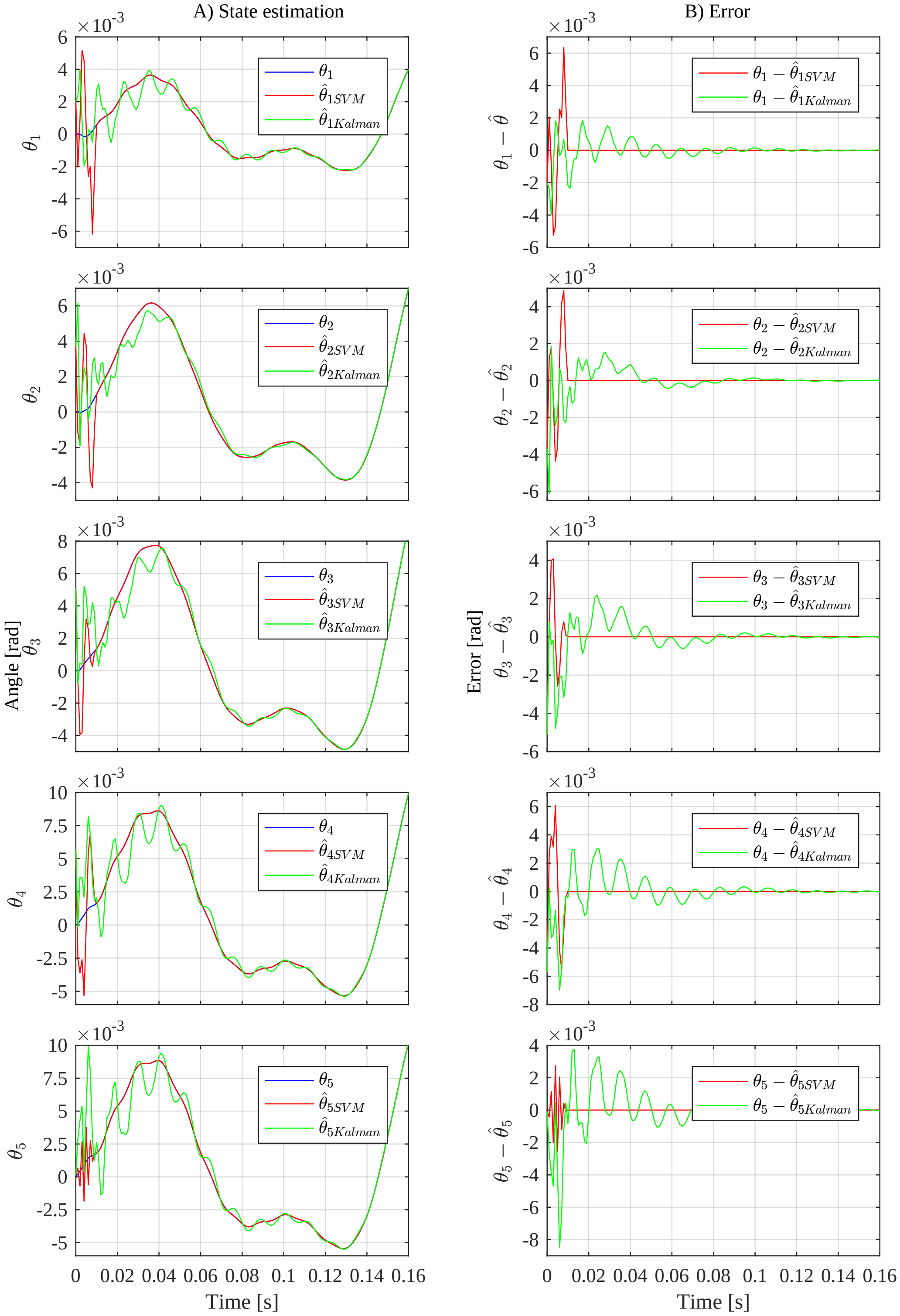

Figure 5.7: Angle estimation (Zoom in)[TS16b]

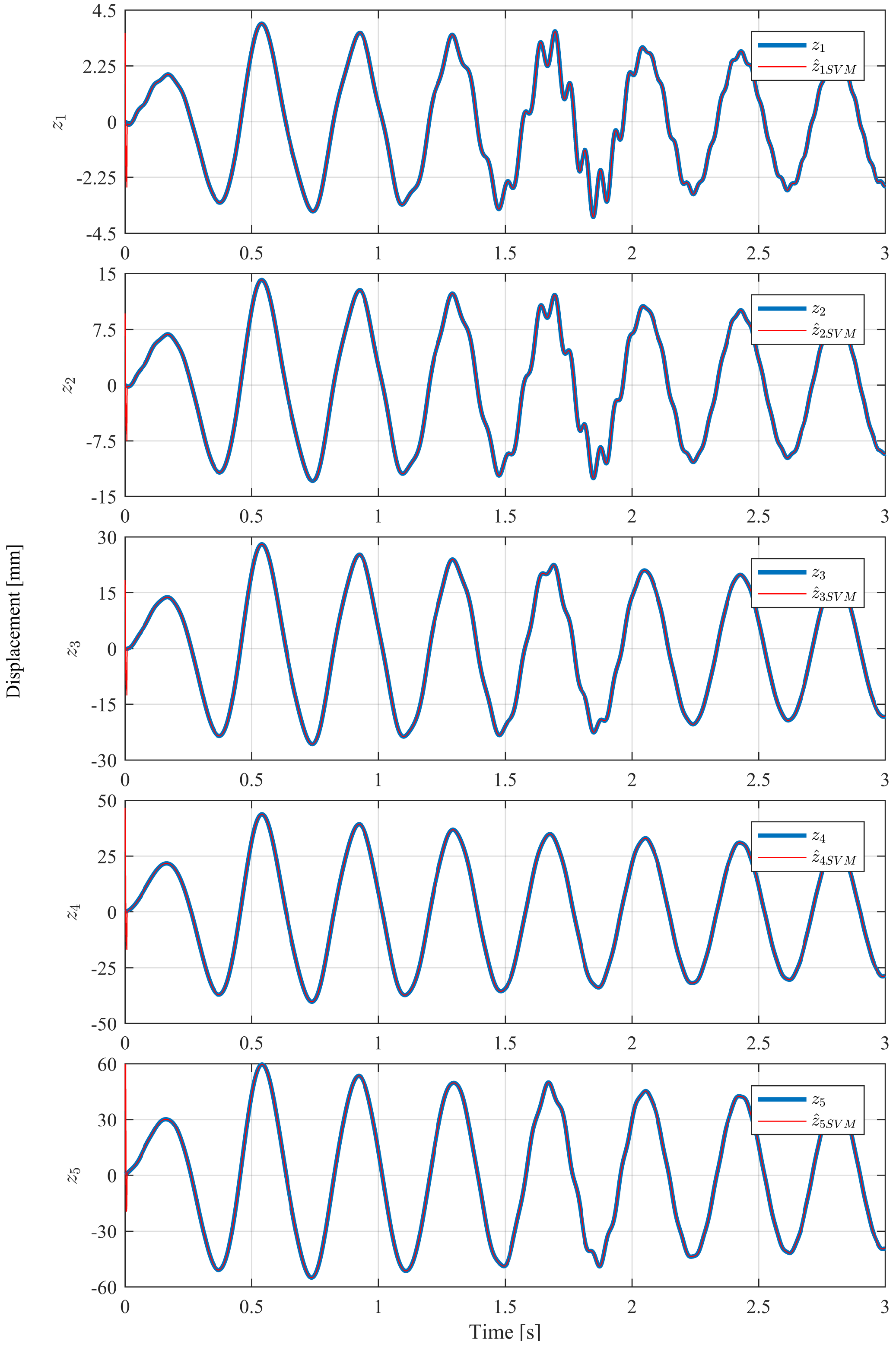

Figure 5.8: Estimation of the displacement using strain signal [TS16b]

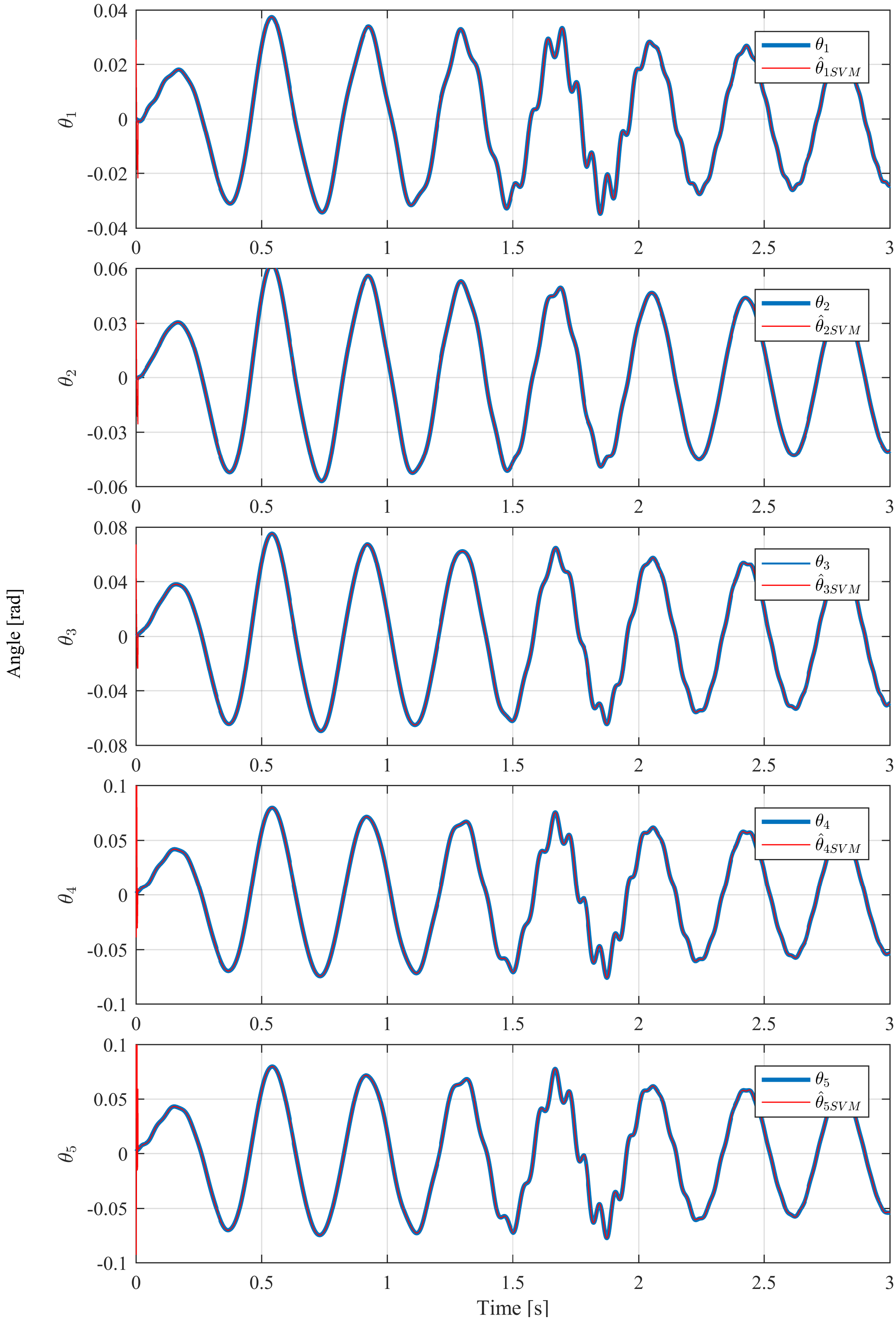

Figure 5.9: Angle estimation using strain signal [TS16b]

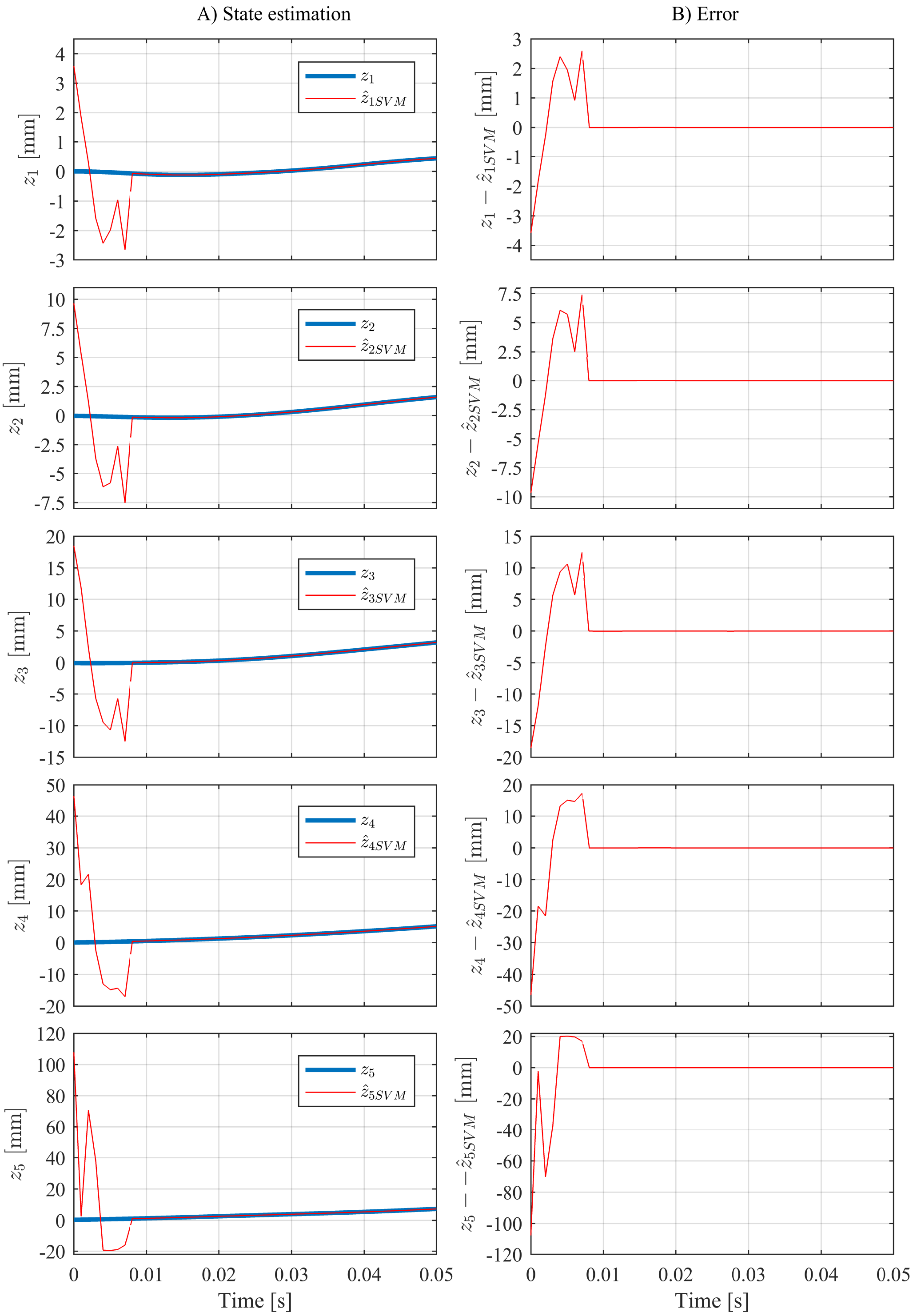

Figure 5.10: Displacement estimation using strain signal (Zoom in) [TS16b]

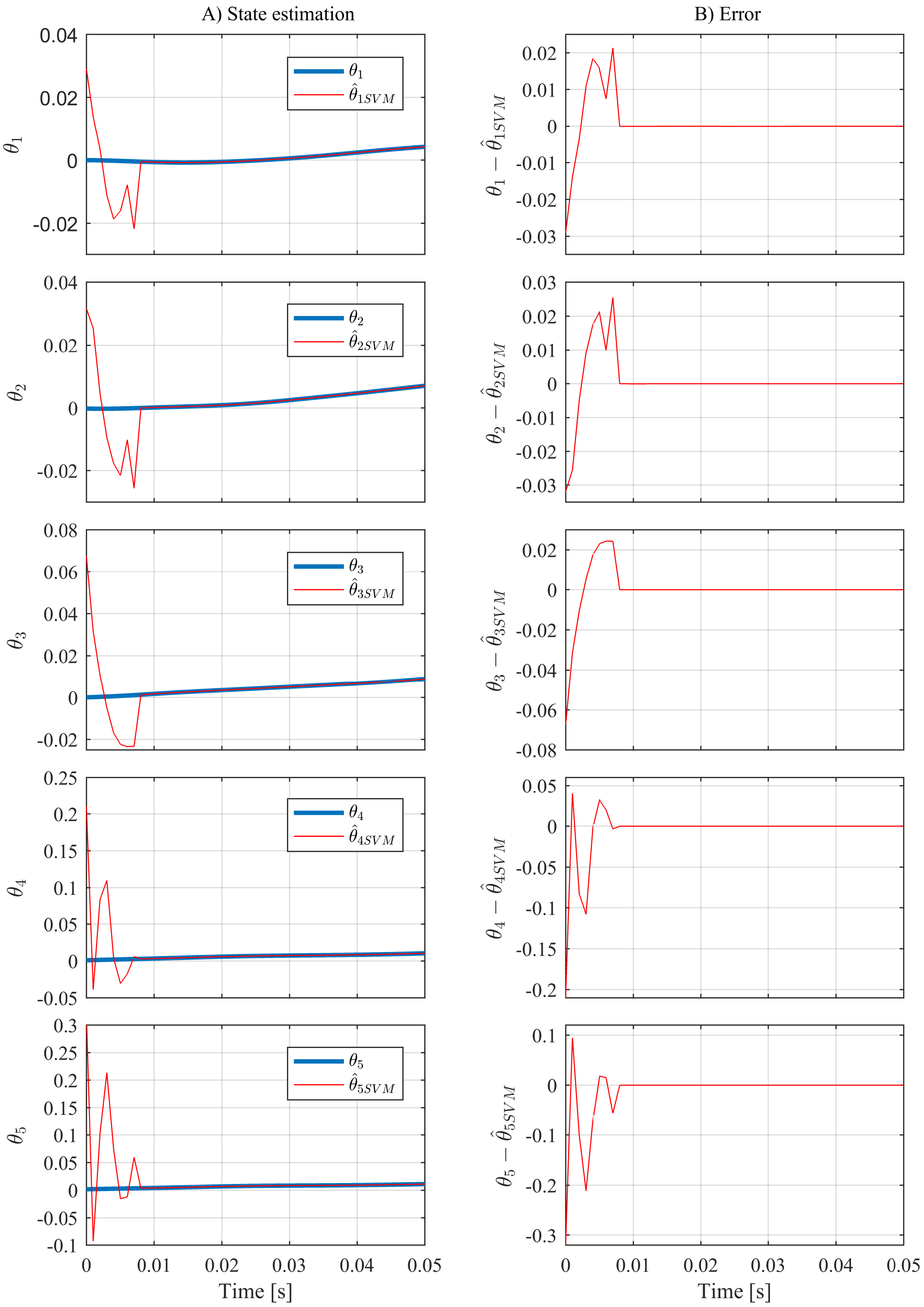

Figure 5.11: Angle estimation using strain signal (Zoom in)[TS16b]

Sensitivity to parameters uncertainty

The previous comparison is done under ideal condition that is when the parameters uncertainty is equal zero. In practice there always a margin for parameters uncertainty. The SVM-based observer is independent of the system parameters, and dependent only on the training data. This is a major advantage of the SVM-based observer over Kalman-Bucy observer.

In this example the parameters of the system are gradually increased by a step of 5.5 percent. In the example only the results of estimating the displacement at z_4 are illustrated in table 5 and figure 5.12. It is observed that SVM-based observer maintained its performance. On the other hand Kalman-Bucy observer performance is reduced with increase of the parameters uncertainty.

Table 5.5: Effect of parameters uncertainty on the observer (SSE_{z4}) [TS16b]

	SVM-based observer	Kalman-Bucy observer
$\Delta = 0$	0.1004×10^{-4}	0.1533×10^{-4}
$\Delta = 0.05$	0.1004×10^{-4}	0.1537×10^{-4}
$\Delta = 0.10$	0.1004×10^{-4}	0.1559×10^{-4}
$\Delta = 0.15$	0.1004×10^{-4}	0.1598×10^{-4}
$\Delta = 0.20$	0.1004×10^{-4}	0.1652×10^{-4}
$\Delta = 0.25$	0.1004×10^{-4}	0.1721×10^{-4}

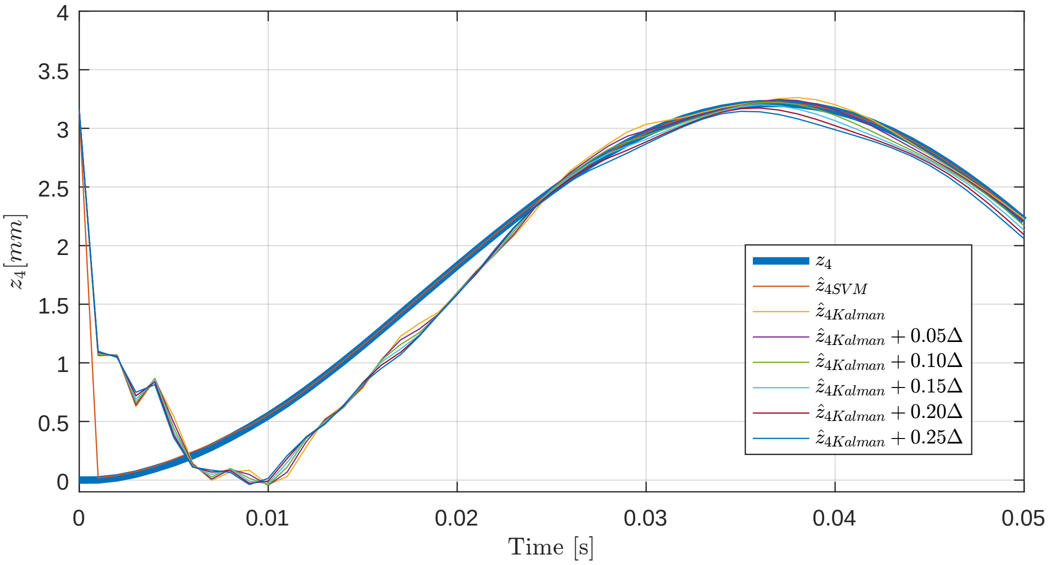

Figure 5.12: Effect of parameters uncertainty on the observer [TS16b]

5.2.1.2 SVM-based observer using the system's output vector

The SVM-based observer can be used to estimate the system state variables using more complex signals. To illustrate this ability in the case of a flexible cantilever beam, strain measurements of the beam are used as the system output. To demonstrate the ability of the SVM-based observer to estimate the output state variables of the systems using experimental data, are used to estimate beam displacement without the external force excitation.

The observer is given as

$$\hat{z}(k+1) = g(y(k), y(k-1), ..., y(k-p)), \tag{5.12}$$

where z is the state vector and y is the system output vector. Here as output y the beam strain (ϵ) signal is used. The training is carried out using training data using the pseudo random signal as input signal. The verification results are shown in figures 5.8, and 5.9. It can be noted that the SVM-based observer is able to estimate the displacements and bending accurately. A zoom into the displacements and bending estimation is provided in figures 5.10 and 5.11. It is noted that the SVM-based observer is able to reduce the error within a very short time (less than 0.18 second) after that the estimation of the state variables is accurate.

5.2.2 Verification of system dynamic model based on SVM

In this section the verification of the modeling of dynamic systems modeling using the SVM algorithm is presented. First the automated system identification algorithm is presented. Then the results of the modeling using SVM algorithm are given.

Modeling of flexible structures using soft computing algorithms

Flexible structures have many advantages over rigid structures such as less material, and lighter weight which results in reducing the required actuating energy. Due to these advantages, rigid structures in the field of mechanics and robotics are replaced by flexible structures. On the other hand flexible structures suffer from undesired vibrations. Therefore vibration suppression techniques are required for flexible structures. One of the major methods in vibration suppression techniques is Active Vibration Control (AVC). To achieve an effective active control strategy for vibration suppression of flexible structures, a precise dynamic model of the targeted structure is required.

System identification approaches define dynamic model describing the input-output behavior of a dynamic system. The implementation of soft computing algorithms for system identification/system modeling is attractive due to facts that no knowledge about the physical background of the system is required and that the approaches can be used to identify linear or nonlinear dynamic systems. This identification approach is categorized as non-parametric modeling which means that the model parameters have no physical meaning.

Different flexible structures, such as aircraft wing or flexible manipulator link, can be represented as a flexible cantilever beam. Modeling of flexible cantilever beams using soft computing algorithms received attention in the past few years. In [IID06] and [JD13] multilayer perceptron (MLP) and Elman neural networks were used to build the model. Elman neural network is a MLP that has an additional connection from its hidden unit to itself to build a directed cycle network. Two major conclusions were noted in [JD13]; first the neuro-modeling techniques have good performance in approximating the system response. Second, the Elman neural network achieved a more accurate estimation than the MLP network. In [JFD13] a MLP network is used to identify a numerical model of the cantilever beam that was obtained using the Finite Difference (FD) method. In [JDM12] a MLP network was used to approximate a finite element method model the flexible cantilever beam. Later on the modeling was extended using an Adaptive Neuro Fuzzy Inference System which has achieved better results. This successful use of the LS-SVM algorithm to produce an equivalent Finite element model of the cantilever beam was presented in [TS15a]. The modeling of the flexible cantilever beam in the above literature was based on simulation data and without considering the effect of noise on the measurements.

5.2.2.1 Automated system identification based on LS-SVM algorithm

In Fig. 5.13 the flow chart of the proposed identification approach is shown. The first step in the approach is to divide the data into training and testing data. This step is important to solve the Bias-variance dilemma which is one of the major issues to be considered when soft computing algorithms are involved. Bias is the expected error due the model mismatch, or it can be understood as the accuracy of the model over different data sets. Variance is the error or variation due to training data set, so it the sensitivity of the model over different data sets.

Bias-variance dilemma is a result of the interaction between the two terms; the increase of bias leads to decrease of the variance and vice versa [TK08].To overcome the bias-variance dilemma the k-fold cross-validation technique is used. In the k-fold cross-validation the data set is divided into k subsets, then each subset is used to train the model and the others are used to test the model validity.

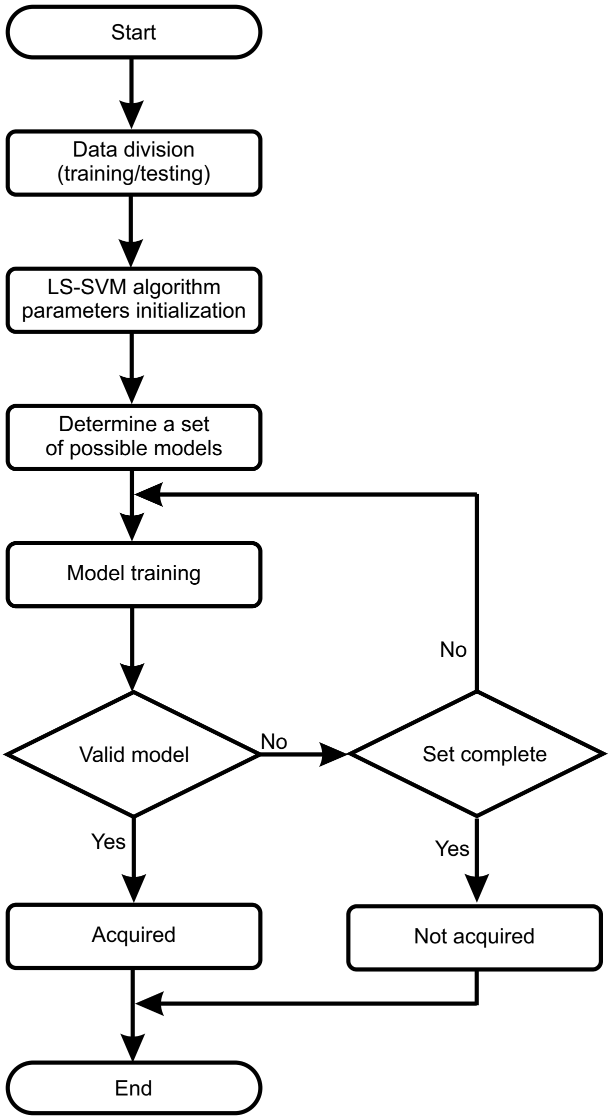
Start
Data division (training/testing)
LS-SVM algorithm parameters initialization
Determine a set of possible models
Model training
Valid model
No
Set complete
No
Yes
Yes
Acquired
Not acquired
End

Figure 5.13: Automated system identification flow chart [TS15b]

The model with the highest generalization and accuracy over all validation sets is selected [Hay08]. In this work the 2-fold cross-validation, which is the simplest form of k-fold cross-validation, is used. In this method the data set is divided into two parts; the first part is used for training, the second part is used for validation, and then the process is reversed. The next step is to initialize the training parameters. The LS-SVM algorithm has two training parameters, the ϵ-tube and the cost. The ϵ-tube can be visualized as a tube of radius ϵ containing the desired output. If the predictions lie within this tube there will be no penalty, otherwise there will be penalty. The amount of the penalty is determined by the cost. It should be noted that choosing high cost may lead to overfitting of the data which results in bad generalization, and choosing a very low cost will lead to underfitting of the data which results in bad results [SV99].

The dynamic system model is unknown therefore a set of models is initialized; this can be given as $p \in [p_{min} \; p_{max}]$. It should be mentioned here that for a perfect dynamic model, the input error cross-correlation should show no correlation and the error autocorrelation should show one nonzero value at zero lag [BHD14]. The autocorrelation function represents the relationship between the prediction error and time. While, the input-error cross-correlation function illustrates the correlation strength between the prediction errors and the model input. The dynamic model accuracy can be improved by increasing the number of delay p [BHD14].

As last step, the model validity is tested; this is done by measuring the discrepancy between the test data y and the model predictions $\hat{y}$. In this work the residual sum of squares (RSS)

$$RSS = \sum_{j=1}^{N} \Big(\hat{y}(j) - y(j)\Big)^2 \tag{5.13}$$

is used. The ideal case is when RSS is equal zero. This could not be achieve in practical cases due to noise and possibly systematic errors. In this work, RSS is assumed as

$$RSS_{desired} = N \times (error_{single\ step})^2.$$

Once a model with RSS equal or less than the desired RSS is determined, the system identification process can be realized. The automated system identification algorithm will update the system model using the number of previous steps p and repeat the training. This step will be repeated until a suitable model is found or the set of models $[p_{min} \; p_{max}]$ is complete.

5.2.2.2 Simulation results

A linear kernel is chosen for the LS-SVM model based on the knowledge that the dynamic system is linear. The number of previous input/output delay p is selected as a function of the sum of squared error (SSE) which is given as

$$SSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{x}(i) - x(i))^2. \tag{5.14}$$

The SSE is considered as a performance measure of the LS-SVM model. The process of selecting p is an iterative process, p is increased until the performance is maximized.

In figure 5.14 the variation of the SSE with changing p is shown. In figure 5.15 the displacement of x_1 for three LS-SVM models; $p = 5$ (green), $p = 12$ (cyan), and

$p = 53$ (red) is shown. The blue curve in figure 5.15 represents the targeted dynamic system response acquired from simulation. It can be observed that when $p = 5$ the LS-SVM model is inaccurate and falling to mimic the dynamic system. In the case $p = 12$, the error in the LS-SVM model is accumulated leading to an increasing output with time. On the other hand for $p = 53$ the SSE shows the lowest value thus best performance. This can be observed in figure 5.15 where the prediction curve (red) is fitting the target curve (blue).

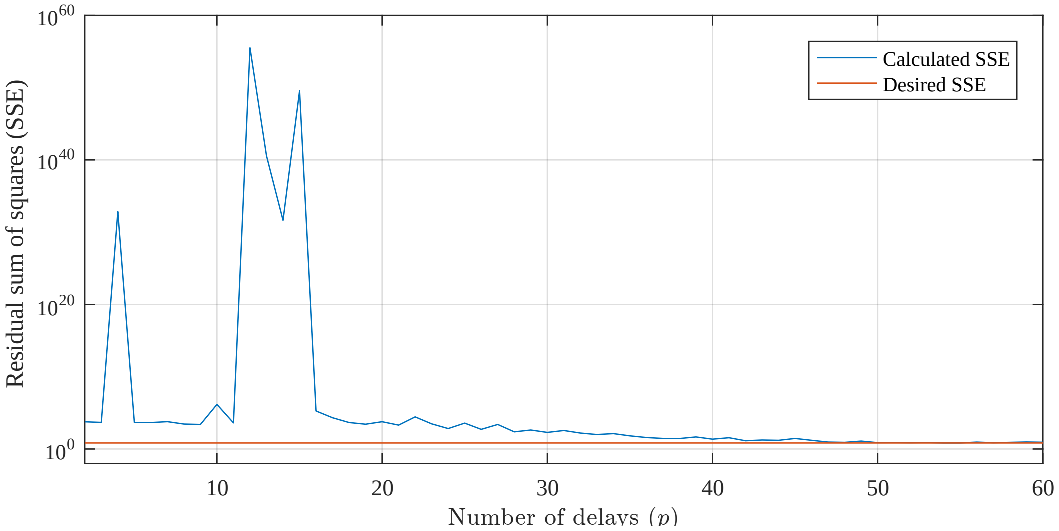

Figure 5.14: SSE vs. number of delays [TS15a]

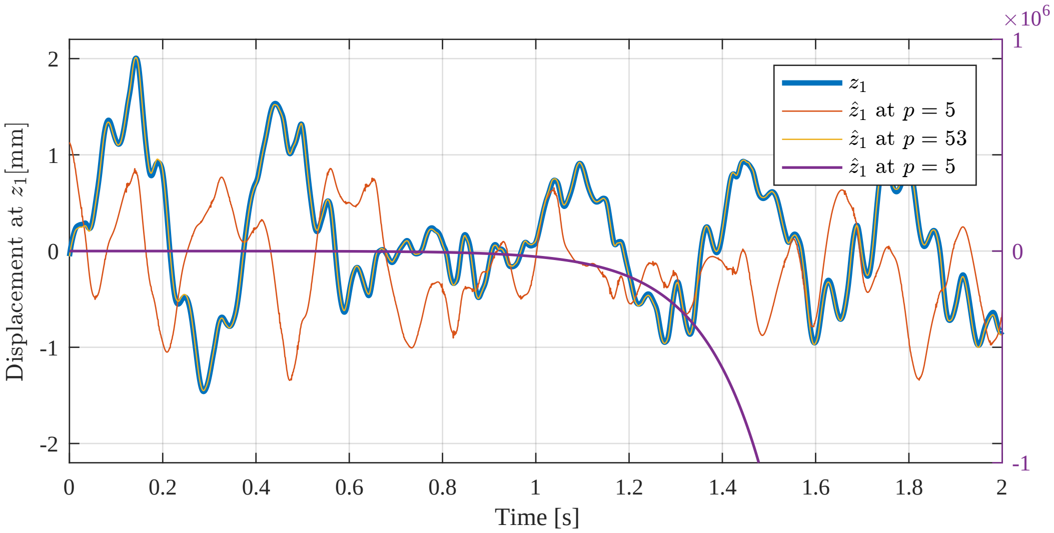

Figure 5.15: Effect of number of delay on the model [TS15a]

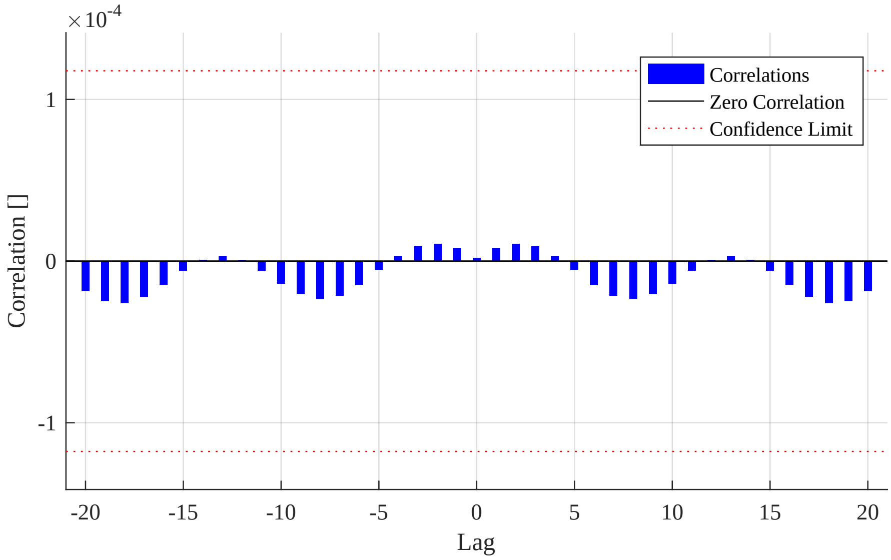

Figure 5.16: Cross-correlation between the input and error [TS15a]

For illustration purposes, the cross-correlation between the input and the error for the LS-SVM model when $p = 53$ is shown in Figure 5.16. It can be observed that all correlation values are under the confidence bounders, this will ensure that the error act white noise and will not accumulate in the prediction. Once the optimal number of delays is determined, an accurate LS-SVM model prediction of the displacements from x_1 to x_5 can be build, the results are shown in figure 5.17. The figure shows the prediction of the LS-SVM model green curve and the target curve in blue. It can observed that prediction of the LS-SVM model is accurate.

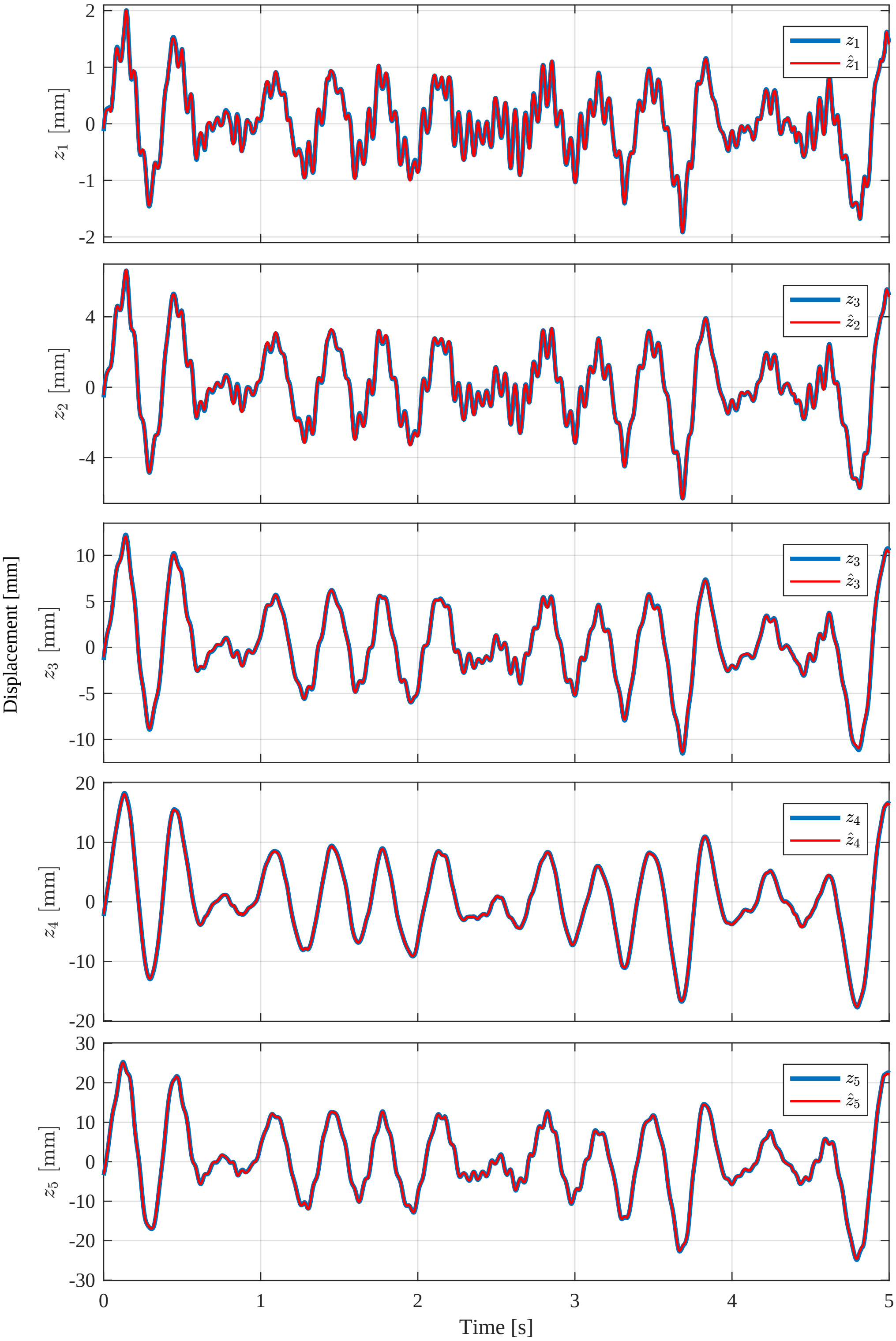

Figure 5.17: LS-SVM model prediction vs. simulation model [TS15a]

5.2.3 Verification of the SVM-based controller

In this section the verification results using the SVM algorithm are introduced. The direct inverse model controller based on SVM is verified using an elastic cantilever beam model shown in figure 5.18. The elastic cantilever beam is similar to the model represented in figure5.1 but in figure 5.18 the disturbance force ($d(t)$) is applied at the tip of the beam and the actuator ($u(t)$) is applied to the second node.

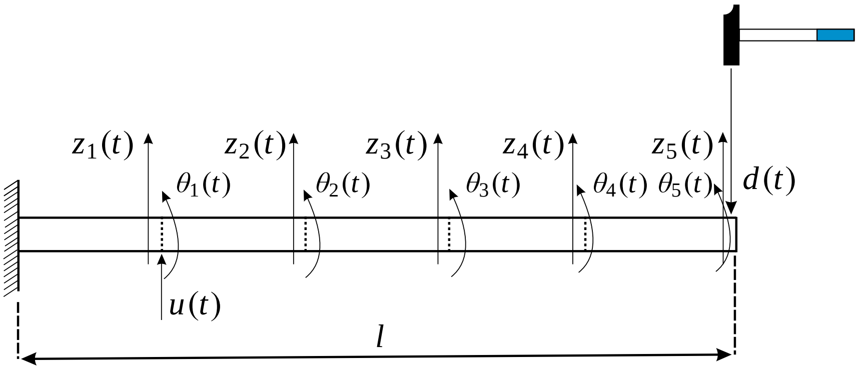

Figure 5.18: Cantilever beam (5 elements) for control

Two SVM-based controllers are proposed to suppress vibrations of elastic cantilever beam. The performances of SVM-based controllers are compared to the open-loop response and different LQR-controllers. The open-loop response (blue), SVM-based controllers closed-loop response (inverse model control in yellow, model reference control violet), and a LQR-controller closed-loop response (red) are shown in figure 5.19. From the figure it can be noted that all three controllers are able to suppress the beam vibrations. It can be observed that the LQR-controller has the fastest vibration suppression time, followed by the model reference controller and then last in the response of the inverse model controller.

In figure 5.20 the actuating signal of SVM-based controllers and a LQR-controller required to suppress the beam vibrations are shown. From the figure it can be observed that the LQR-controller requires less energy that the SVM-based controllers. The model reference controller based on SVM requires less energy than the inverse model controller based on SVM.

The evaluation of the different controllers is done using the criteria developed in [LS09] where the error and control effort are graphically represented. In figure 5.21 the performance of the SVM-based controllers can be evaluated and compared to different LQR-controllers. The LQR-controllers performance (blue) is obtained by changing the R parameter while maintaining the Q matrix fixed. From the

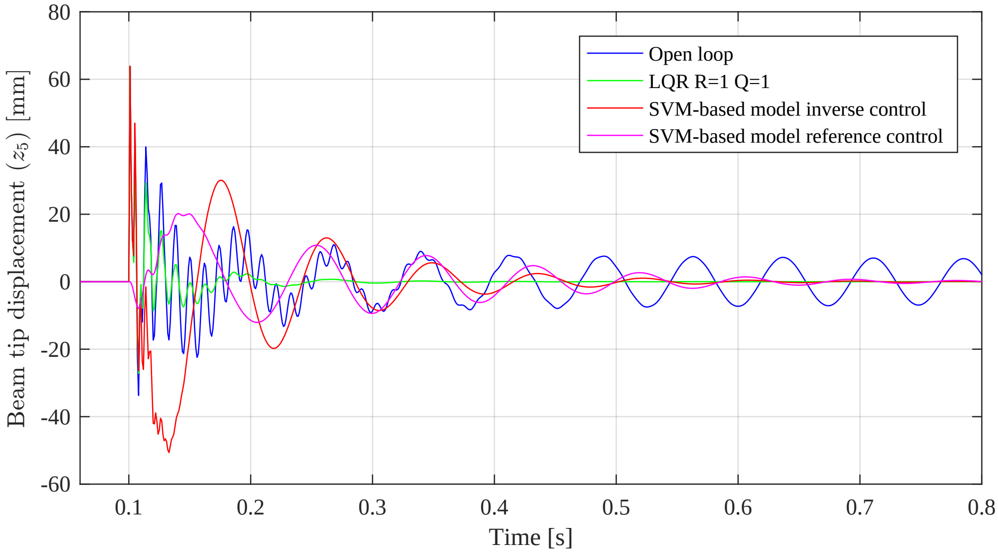

Figure 5.19: Time response of the controllers

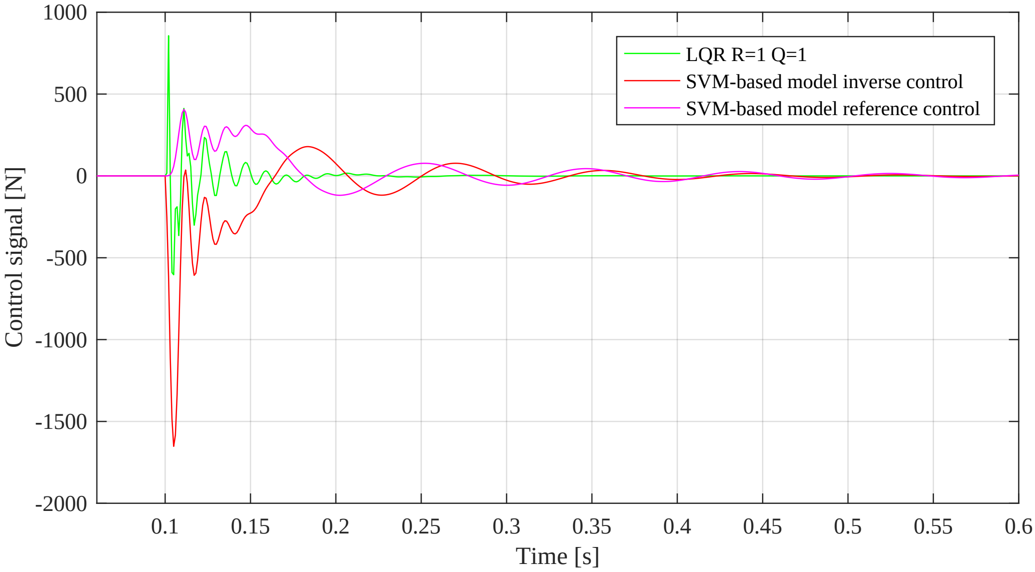

Figure 5.20: Actuator signals

figure it can be observe that SVM-based controllers performance is near the LQR-controller with R=$10^{0.5}$. In figure 5.22 a zoom in to the area is provided. For the inverse model controller is farther from the LQR-controller, it can be observed that increasing the number of support vector improves the performance of the inverse model controller. The model reference controller has better performance than the inverse model controller and it has performance near the LQR controller. Also it is noted that the number of the support vector in not influencing the performance of the model reference controller.

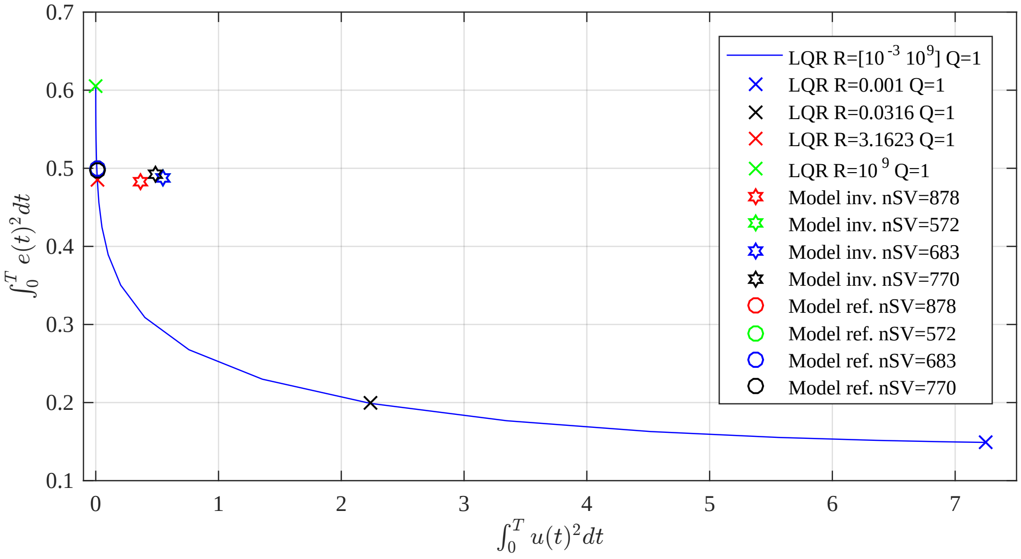

Figure 5.21: Controllers performance comparison

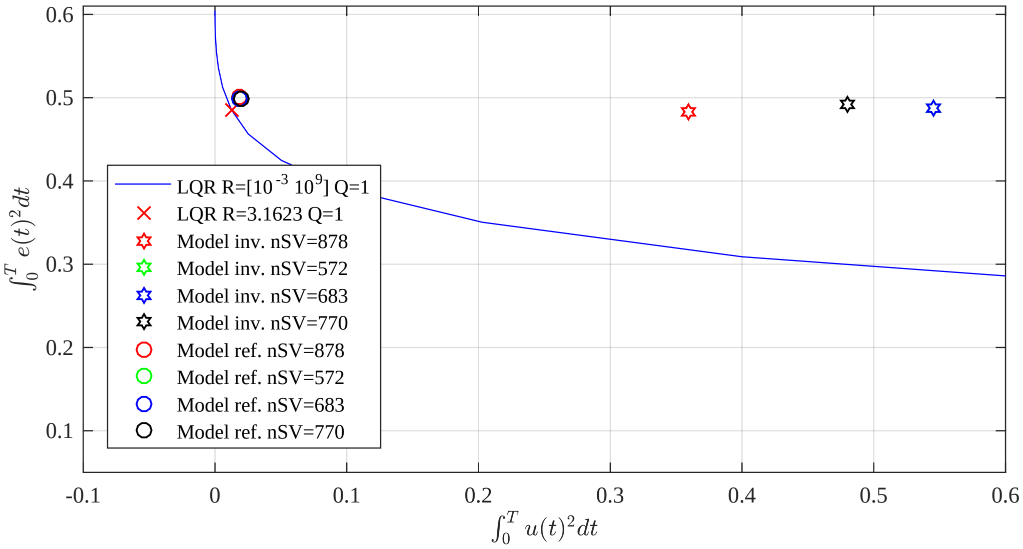

Figure 5.22: Controllers performance comparison (Zoom in)

5.3 Summary

This chapter presents a soft computing-based controller using SVM algorithm that able estimate that dynamic system sate variable, model the dynamic system and control the dynamics system is presented. To verify the consent of soft computing-based controller a cantilever beam based on FEM is used as benchmark.

It is shown that SVM-based observer able to estimate the state variables of the system using output data/measurements. The main advantage of the SVM-based observer is that it requires no theoretical knowledge about the dynamic system, only the use of measured data are sufficient to build the SVM-based observer. Modeling of dynamic system based on LS-SVM is given, and the automated system identification approach based on LS-SVM is presented. This approach is used to find the best model that fit an unknown dynamic system. The acquired model using the automated system identification algorithm based on LS-SVM demonstrates high accuracy.Finally, the control of the cantilever beam base on the SVM-based controller is demonstrated. It is shown that the model reference control based on SVM algorithm is able to achieve good performance.

6 Validation of the soft computing-based controller

In this chapter the experimental results of the validation of the introduced controller scheme for complex mechanical system are presented. First, the test rigs used for validation are described. The test rigs are the elastic cantilever beam and the inverted elastic cantilever beam. The validation results of the introduced controller scheme is given, this includes validation for SVM-based observer, SVM-based dynamic model, and SVM-based controller. Parts from this chapter are based on text and material that are previously published [TS16b, TS15b].

6.1 Test rigs

The soft computing-based controller concept is validated using two test rigs which are the elastic cantilever beam and the inverted elastic cantilever beam. The validation of the SVM-based observer and dynamic model are done using the elastic cantilever beam. The validation of the SVM-based controller is realized using the inverted elastic cantilever beam.

6.1.1 Elastic cantilever beam test rig

The cantilever beam test bench used to validate the approach is shown in Fig. 6.1. The test bench consists of a flexible beam clamped on one side, two optical sensors to measure the beam displacement at two different positions, three full bridge strain gauges fixed at three different positions to measure the beam strain. An impact hammer is used to apply impact forces at a position of choice. A sketch of the test bench is shown in Fig. 6.2.

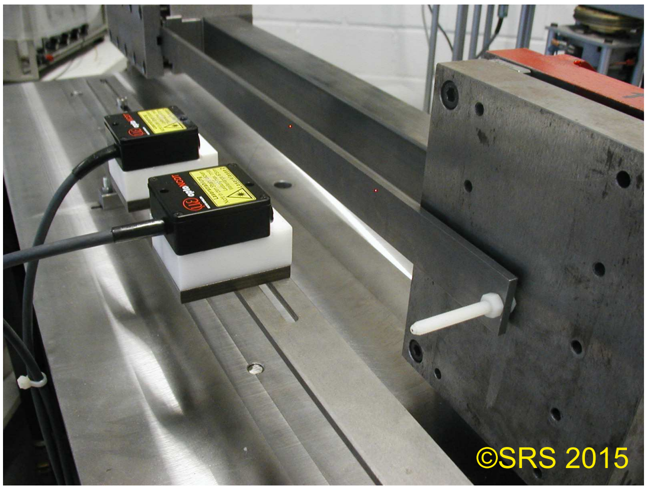
©SRS 2015

Figure 6.1: Cantilever beam test rig, Chair of Dynamics and Control, University Duisburg-Essen

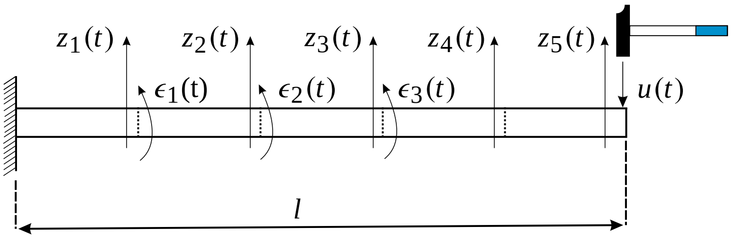

Figure 6.2: Sketch of the test bench [TS15b]

The parameters of the steel beam are given in table 6.1. In this experiment, the impact force was applied at the tip of the beam at position z_5, the tip position z_5 was measured. During the experiment 100 seconds of input output data was recorded at a sample time of 1 ms. The measured input force is shown in 6.3 and the measured tip vibration is shown in Fig.6.4. Vibration of the beam tip is observed between -5 to 5 mm, while the force range is between -8 to 8 N. There are 4 time periods where no force is applied. A zoom into the measured force is shown Fig. 6.5 and a zoom into the measured tip vibration Fig. 6.6.

Table 6.1: Beam parameters

Density (ρ)	8050 Kg/m^3
Length (l)	546 mm
Cross section area (A)	30×5 mm^2
Modulus of elasticity (E)	200×10^9 N/m^2

The spectral analysis resulting from a measurment of the beam tip displacement is shown in Fig. 6.7. From the amplitude diagram shown, the first three frequencies can be observed. Then the spectral analysis results were compared with results obtained using analytical approach. For the analytical approach, first the beam was assumed to be a continuous system, thus the natural frequencies ω_n are given as

$$\omega_n = k^2 \sqrt{\frac{EI}{A\rho}}, \tag{6.1}$$

where $k = \frac{(2n-1)\pi}{2}$, E is the young modulus of elasticity, I is the area moments of inertia, A is the cross section area, and ρ is the density. Then a finite element model of the beam was built, the model consisted of five element. According to [KB00], the

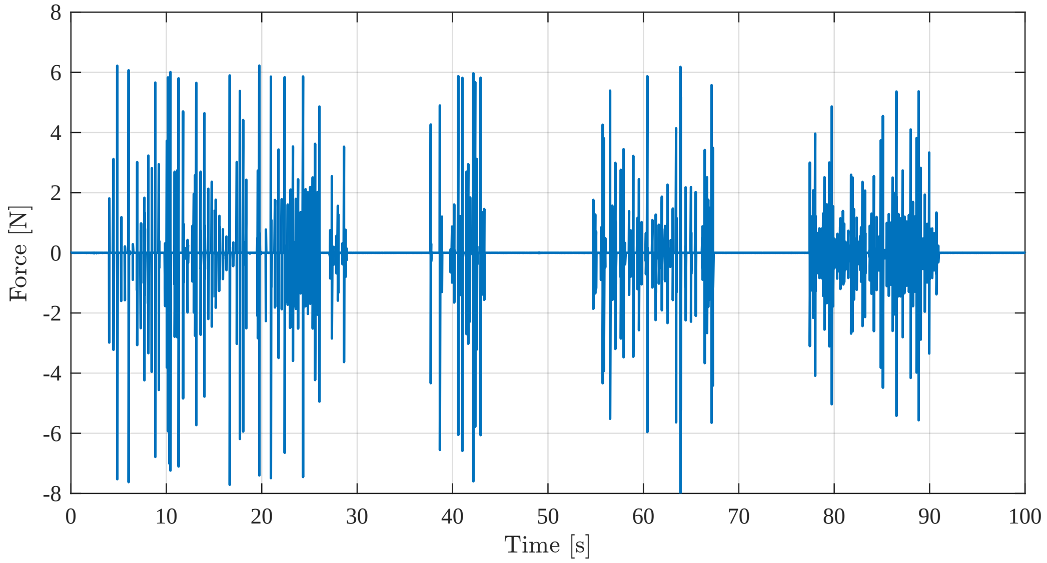

Figure 6.3: Input signal: Excitation force [TS15b]

estimation of the damping matrix C for a physical system is not easy. for simplicity, the Raleigh damping was defined as

$$C = \alpha[M] + \beta[K], \tag{6.2}$$

where α and β are constants to tune the damping behavior of the system as best as possible, M is the global mass matrix and K is the global stiffness matrix.

The result of the comparison is shown in Tab. 6.2. The difference between the

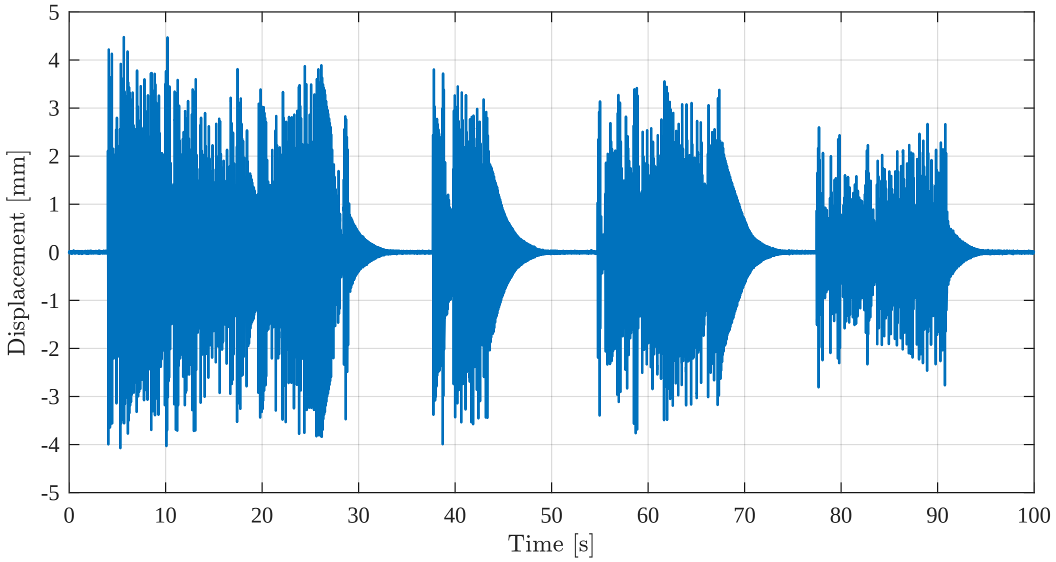

Figure 6.4: Beam tip vibration [TS15b]

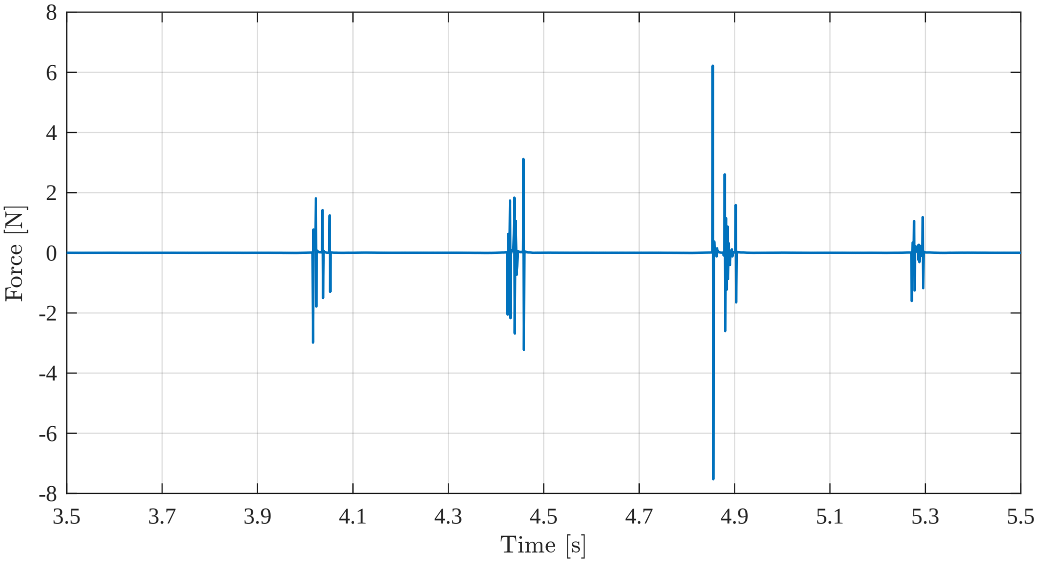

Figure 6.5: Zoom in: Input signal [TS15b]

analytical results and experimental results is due to the damping hypothesis. On the other hand the FEM model results are better for the first and third resonance frequencies, but the FEM model second resonance frequency has the largest error. This analysis is very important to ensure that the desired dynamics is included in the measured data. The analysis shows that the measured data includes the desired dynamics which will improve the results of the model generated by the identification algorithm.

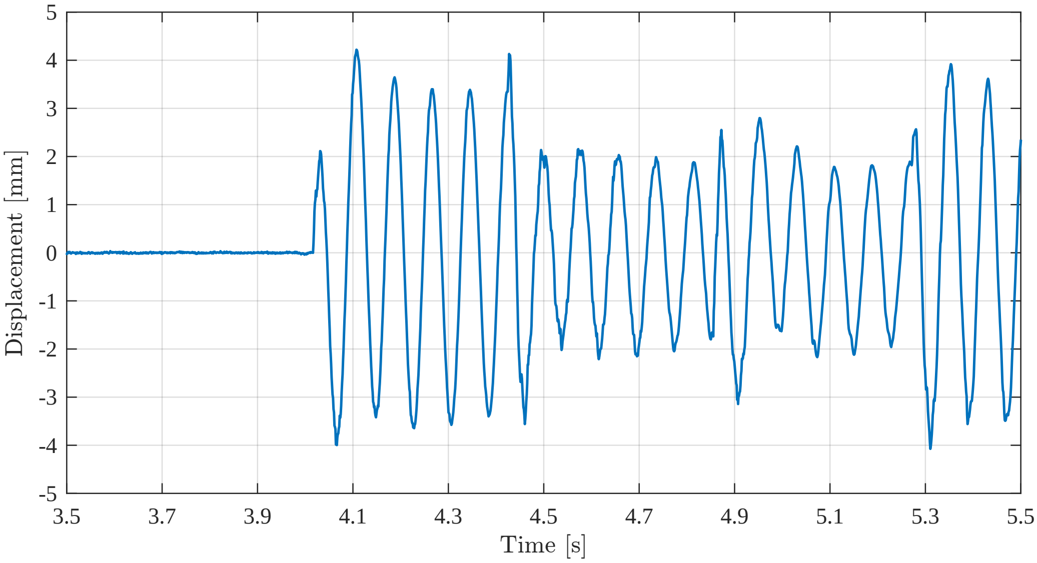

Figure 6.6: Zoom in: Beam tip vibration [TS15b]

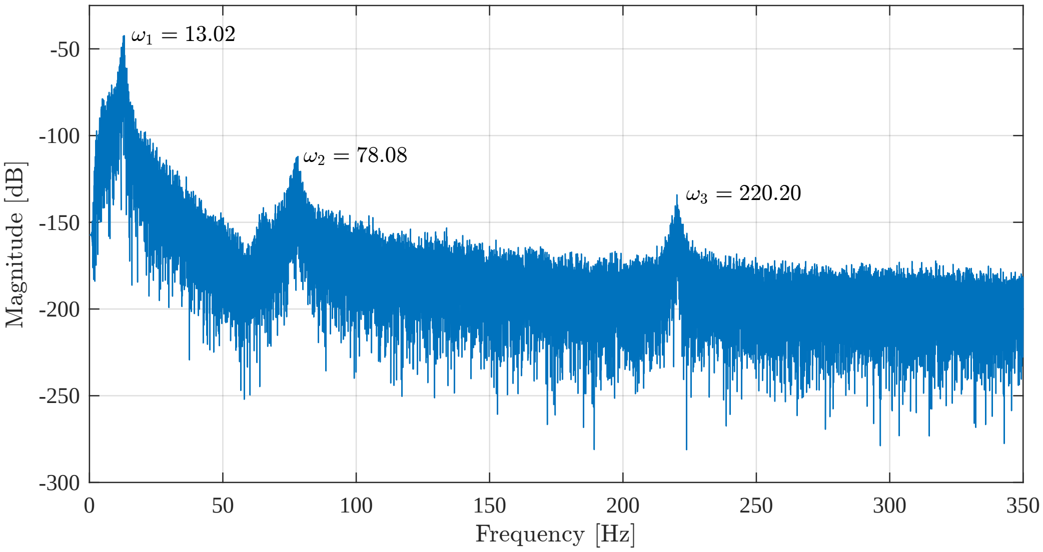

Figure 6.7: Frequency response

Table 6.2: Resonance frequencies of beam [TS15b]

Resonance frequencies	Analytical results	FEM results	Experimental results
First frequency (ω_1)	13.43 Hz	13.02 Hz	13.02 Hz
Second frequency (ω_2)	84.48 Hz	84.63 Hz	78.08 Hz
Third frequency (ω_3)	236.68 Hz	222.25 Hz	220.20 Hz

6.1.2 Inverted elastic cantilever beam test rig

The SVM-based controller is validated using the inverted elastic cantilever beam test rig shown in figure 6.8. The test rig consists of an elastic beam fixed at one end to moving cart, a mass attached to the other end. The cart is to do translation moving. The translation of mass attached to the end of the beam is measured using a displacement laser sensor. The cart position is measured by transforming the rotation of the motor measured by optical encoder into translation motion. The deflection of the beam is measured using strain gauge fixed near the fixed end of the beam. A sketch of the inverted cantilever beam is shown in figure 6.9. The hardware in the simulation is realized using dSPACE system. The parameters of the inverted cantilever beam are given in table 6.3. In figure 6.10 a sample of open loop excitation signal is shown. The resulting tip beam deflection is shown in figure 6.11.

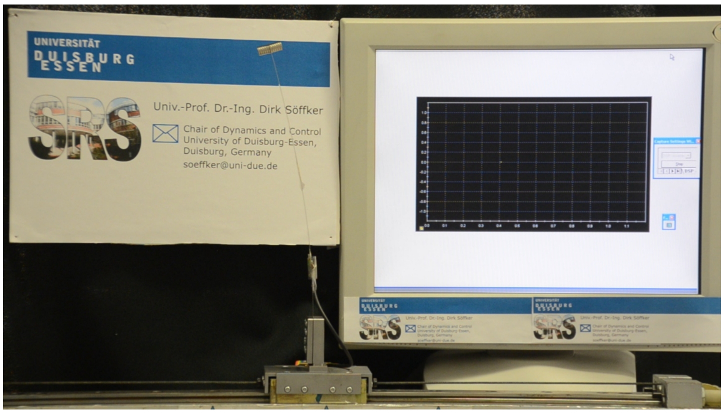
UNIVERSITÄT
DUISBURG
ESSEN
SRS
Univ.-Prof. Dr.-Ing. Dirk Söffker
Chair of Dynamics and Control
University of Duisburg-Essen,
Duisburg, Germany
soeffker@uni-due.de

Figure 6.8: Inverted cantilever beam test rig, Chair of Dynamics and Control, University Duisburg-Essen

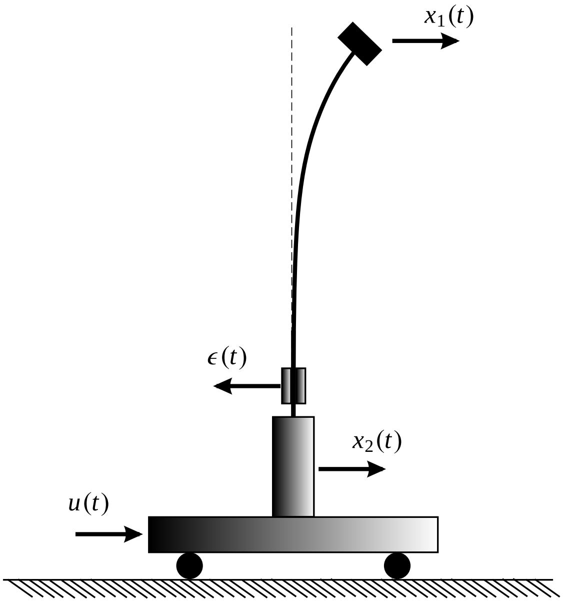

Figure 6.9: Sketch of the inverted cantilever beam test rig

Table 6.3: Inversed cantilever beam parameters

Density (ρ)	8050 Kg/m^3
Length (l)	260 mm
Cross section area (A)	0.03×0.2 mm^2
Modulus of elasticity (E)	200×10^9 N/m^2

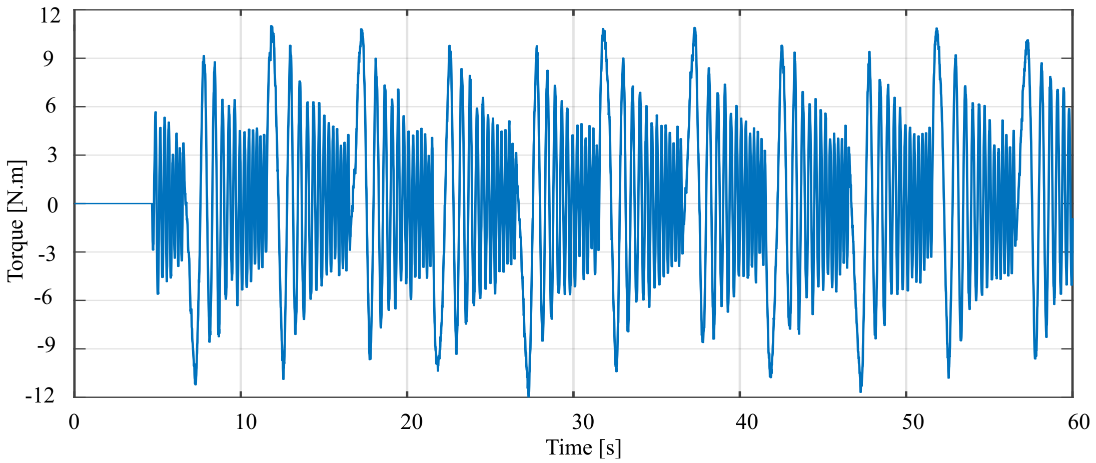

Figure 6.10: Input signal: Excitation Torque

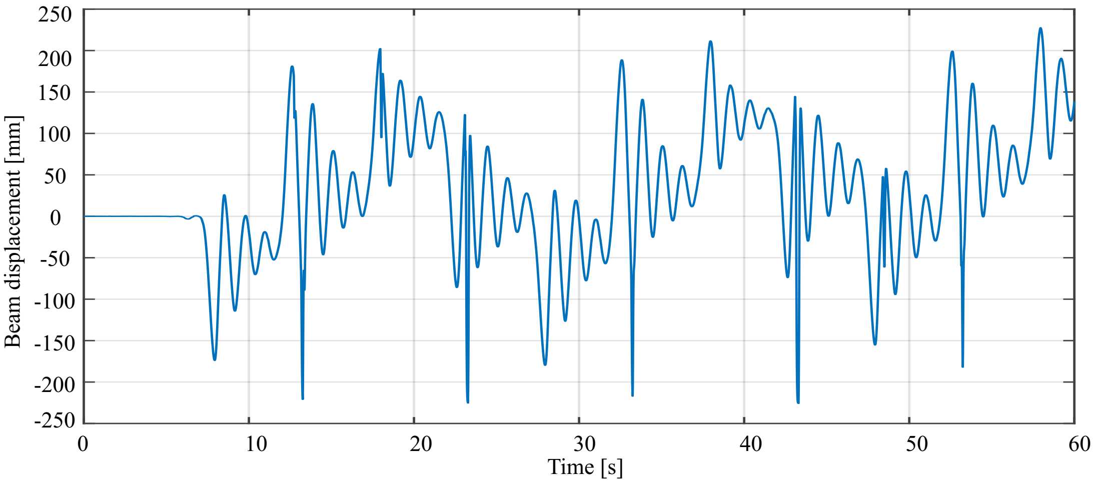

Figure 6.11: Beam tip vibration

Table 6.4: Resonance frequencies of inverted cantilever beam

Resonance frequencies	FEM results	Experimental results
First frequency (ω_1)	1.14 Hz	0.9 Hz
Second frequency (ω_2)	12.16 Hz	11.9 Hz
Third frequency (ω_3)	36.42 Hz	50.3 Hz
Forth frequency (ω_4)	74.8 Hz	150.2 Hz
Fifth frequency (ω_5)	128.54 Hz	249.8 Hz
Six frequency (ω_6)	312.36 Hz	288.9 Hz
Seventh frequency (ω_7)	455.95 Hz	449.6 Hz

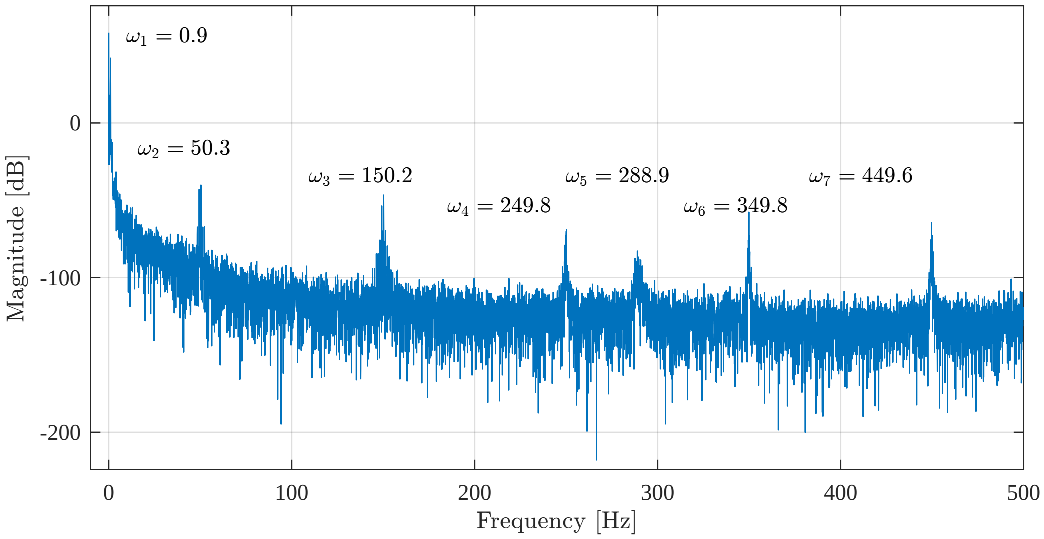

Figure 6.12: Frequency response

6.2 Validation of the soft computing-based controller

In this section the validation results of soft computing-based controller are presented. The section is divided into three parts. In the first part, the validation results of the SVM-based observer is given. In the second part, the validation results of the SVM-based dynamic model is shown. In the third part, the validation results of the SVM-based contrller is shown.

6.2.1 Validation of the SVM-based observer

To validate the SVM-observer introduced a test bench is used, experimental results are presented.

Experimental results

The results of the SVM-based observer using the force and tip displacement are given. The results of the SVM-based observer using the strain signals are outlined.

Experimental results of the SVM-based observer using force and tip displacement

The experimental design of an SVM-based observer for the cantilever beam using experimental data is discussed. As described in 5.2.1.1 the SVM-based observer is trained using the input f and output tip displacement ($y = z_5$) signals to estimate the cantilever beam states. A sample of 6 seconds is randomly selected to train the SVM-based observer. The validation results are shown in figure 6.13. It is noted that the SVM-based observer accurately estimates the states z_1, z_2, and z_3.

To evaluate the performance a comparison between the SVM-based observer and Kalman-Bucy observer is done, the results are shown in figure 6.14. The conditions (used measurements) are equal. As model for the Kalman-Bucy observer the model based on the FEM is used. It is observed that both observers have good performance in estimating state variables. It is also noted that although the initial conditions of the observes are different from those of the actual system, the observers are able to reduce the error in very short time. Independent from similar results obtained, it should be noted that the SVM-based observer result is not based on any kind of analytical assumptions or numerical first order model as the Kalman-Bucy observer. Both approaches use inputs and output signals to make the estimation for the state variables.

Table 6.5: Sum of squared error of the displacement estimations using experimental data [TS16b]

	SSE_{z1}	SSE_{z2}	SSE_{z3}
SVM-based observer	9.22	4.73	3.87
Kalman-Bucy observer	9.21	28.14	5.18

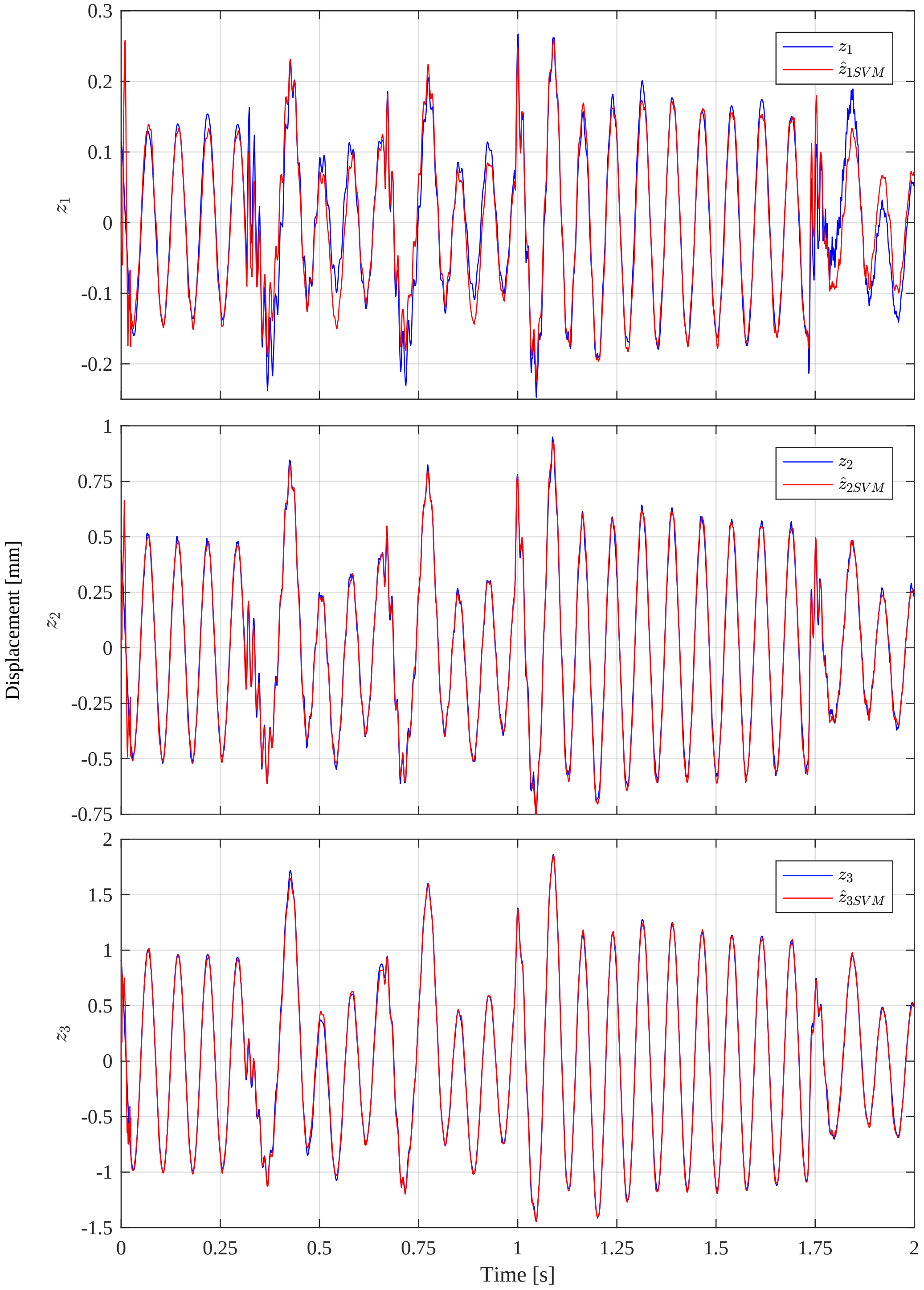

Figure 6.13: Estimation of the beam displacement using experimental data [TS16b]

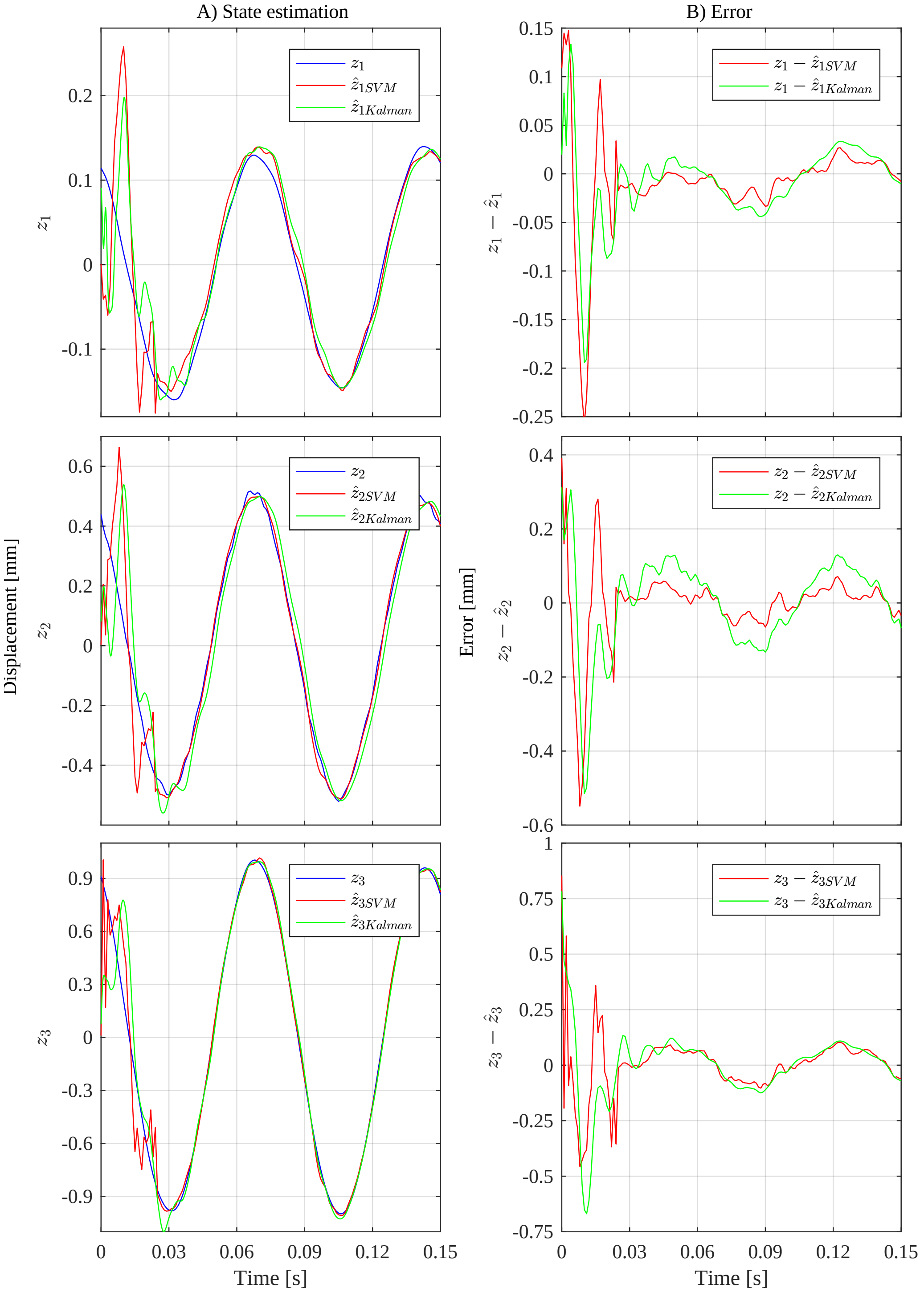

Figure 6.14: Estimation of the beam displacement using experimental data (Zoom in)[TS16b]

Experimental results of the SVM-based observer using strain signal

The SVM-based observer is trained using the strain signals to estimate the elastic beam displacements without the use of the input force signal. The same data set used in the previous subsection is used. The results of the beam displacement at the positions z_1, z_2, z_3, and z_4 are shown in figure 6.15. It can be noted that the accuracy of the estimation is satisfactory. The least accuracy is observer in the estimation of z_1 due to the high noise level.

In figure 6.16 the first 0.15 seconds of the results is shown, the transient response of the SVM-based observer due to the different initial conditions between the system and the SVM-based observer can be noted. The SVM-based observer has fast transient response, it requires less than 0.018 of a second to reduce the transient error for states z_2, z_3 and z_4 and around 0.11 of a second for state z_1.

The SSE results using the experimental data of the positions z_1, z_2, and z_3 are shown in table 6.5. It is noted, that the SVM-based has relatively better performance, it is also noted, that the performance of displacement z_3 is the best. It can be concluded that, the SVM-based observer is able to estimate the state variables of the elastic cantilever beam using the strain signals.

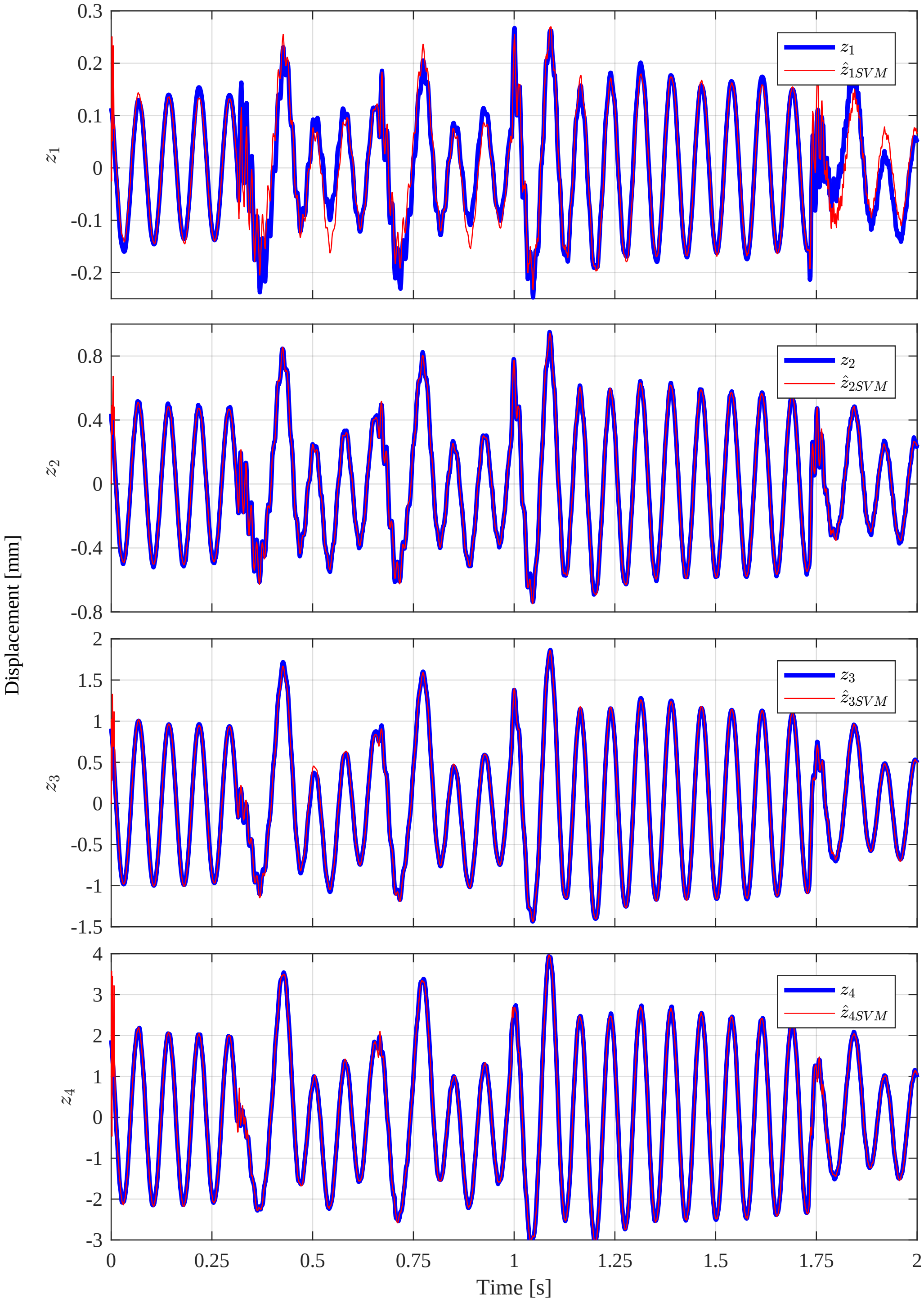

Figure 6.15: Estimation of the displacement using strain signals [TS16b]

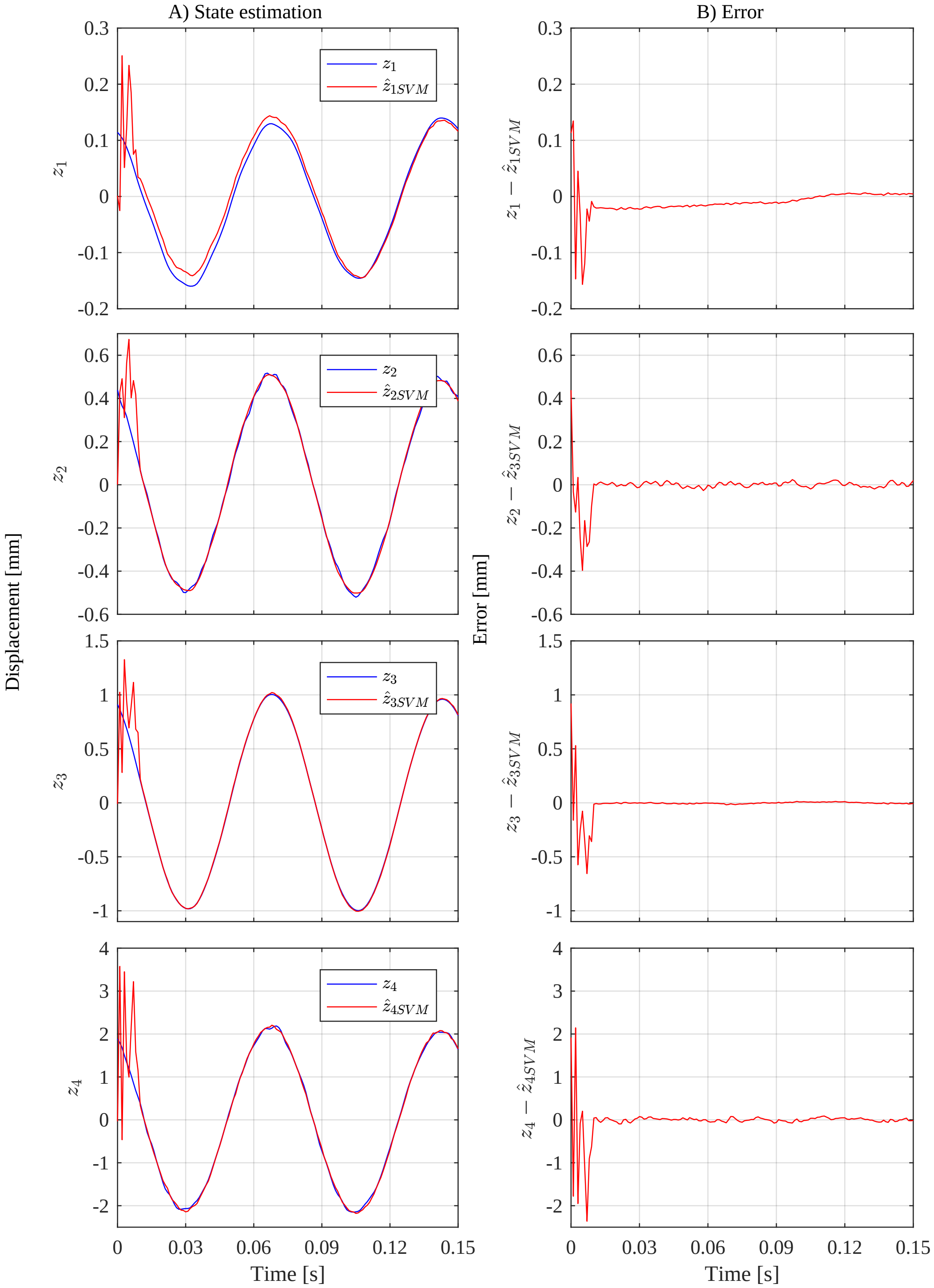

Figure 6.16: Estimation of the displacement using strain signals (Zoom in)[TS16b]

6.2.2 Validation of the SVM-based model

Discussion of results

Once the analysis is done, the data was applied to the automated system identification algorithm. For training data, the measurement between 3.8 to 5.8 seconds (2000 samples) are selected. as for validation, the measurement between 54.5 to 58.5 seconds (4000 samples) are selected. The set of models is defined between p=2 to 25. The desired RSS is below 0.2 for 2000 samples, which gives average prediction error of 0.01 mm for each prediction.

A model with the desired requirements was determined at $p = 20$. The model consists of 650 support vectors (SV) as training results and has achieved RSS=0.0897 on training data and RSS= 0.1051 on the validation data. In Fig. 6.17 and Fig. 6.18 the predicted displacement and measured displacement are shown, in both figures, the accuracy of the prediction is observed. The prediction errors of the validation data is shown in figure 6.20. The prediction erorrs are in the range of [-0.02 0.025] mm, the average prediction error is 0.002 mm. Further analysis of the model prediction errors autocorrelation is shown in Fig. 6.21, where it can be noted that the prediction errors have significant correlation with each other and therefore the model is not suitable to produce multi-step ahead predictions.

The results demonstrate the ability of the LS-SVM algorithm in modeling dynamic systems. Further optimization of the automated system identification algorithm is required.

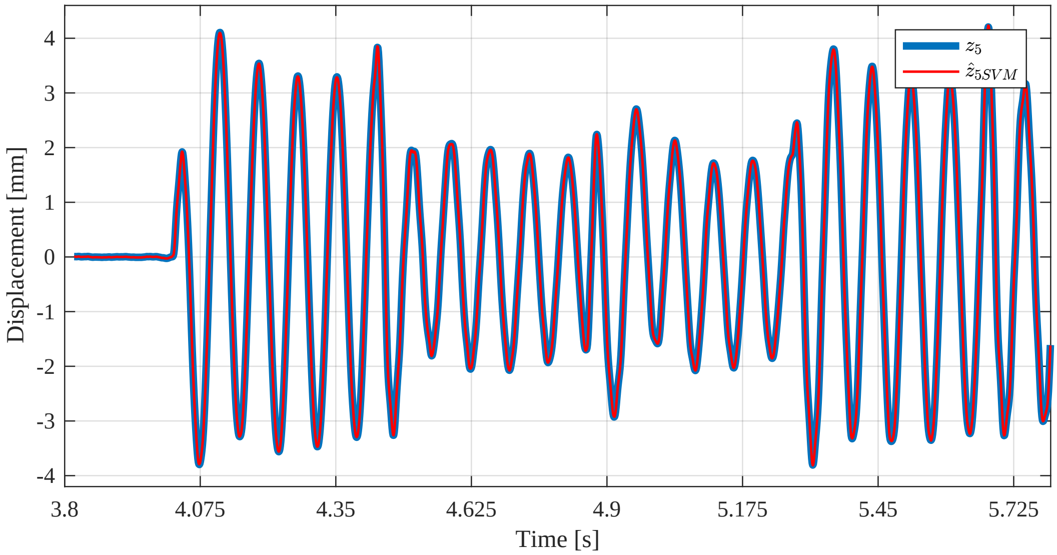

Figure 6.17: Training results [TS15b]

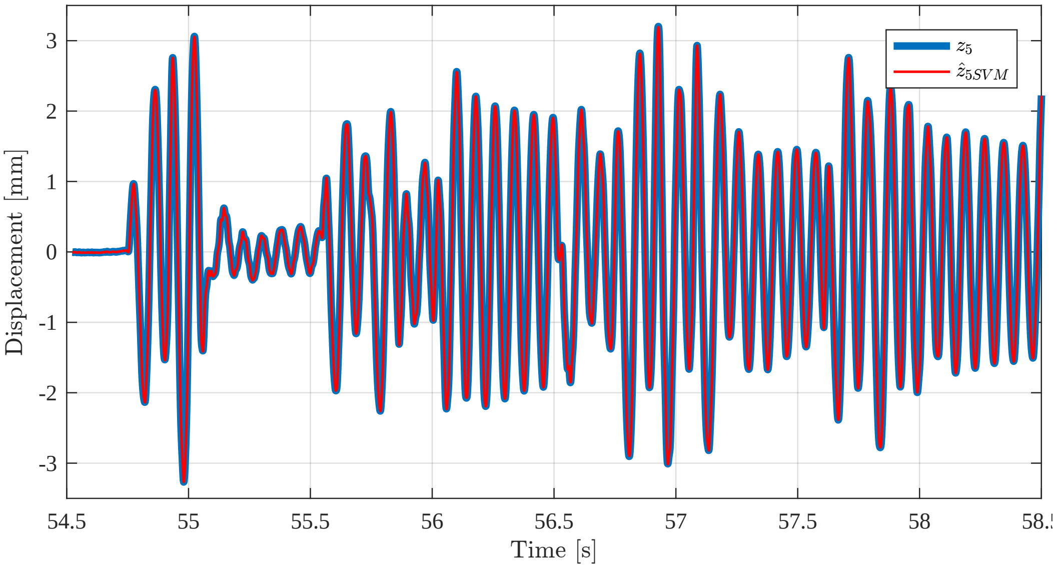

Figure 6.18: Validation results [TS15b]

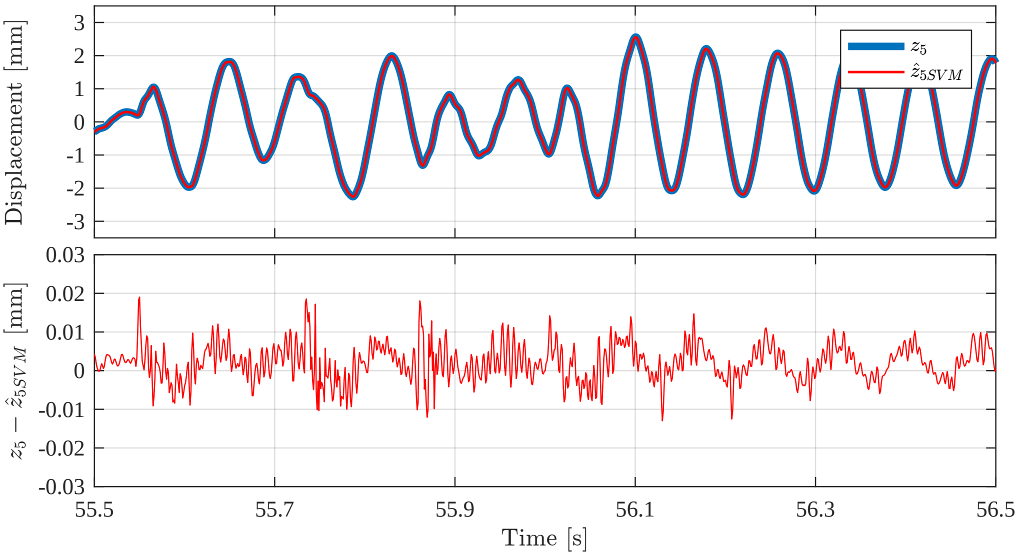

Figure 6.19: Validation results (zoom-in) [TS15b]

Comparison with Multilayer perceptron network

In this section, a short comparison between a cantilever beam model obtained using the multilayer perceptron (MLP) network and the model obtained using LS-SVM is given. The MLP network is considered as the classical modeling soft computing tool for dynamic systems.

The weighting matrix v, the output weights vector w, and the bias b are calculated using Levenberg-Marquardt backpropagation algorithm. To model the elastic can-

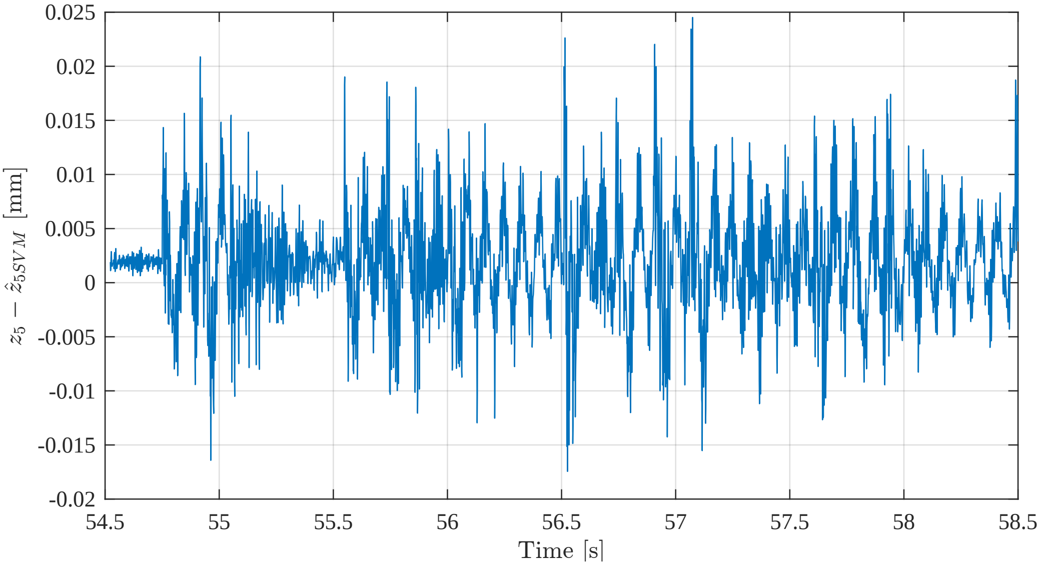

Figure 6.20: SVM-based model predication error[TS15b]

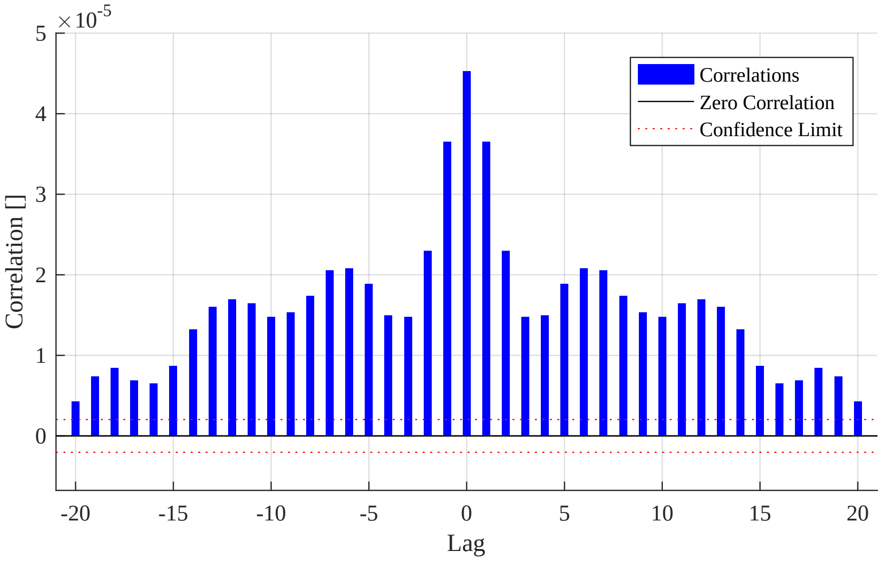

Figure 6.21: Autocorrelation in prediction error [TS15b]

tilever beam the number of delays p was set to 20, the number of neurons in the hidden layer J was set to 10 neurones. The validation results are shown in Fig. 6.22. In Fig. 6.23 the prediction errors of the MLP network is shown. It can be observed that the error prediction has a wider range than the SVM-based model prediction error shown in Fig.6.20. In Tab. 6.6 the results from the SVM-based model are compared to the MLP-based model. From the table is can be noted that the MLP-based model has a simpler structure than the SVM-based model. The performance (RSE) of the MLP-based model is better than the SVM-based model based on the training data set. When it was applied on the test data the SVM-based model a better performance can be achieved. This means that the SVM-based model allows a better generalization than the MLP-based model. In the last row of the table, the time required to calculate the models is given. To calculate the SVM-based model 0.38 s is required, while to calculate the MLP-based model 2.59 s is required. This shows a difference in the computational load required to calculate each model. The training time of SVM-based model is approximately 7 times shorter than the training time of MLP-based model.

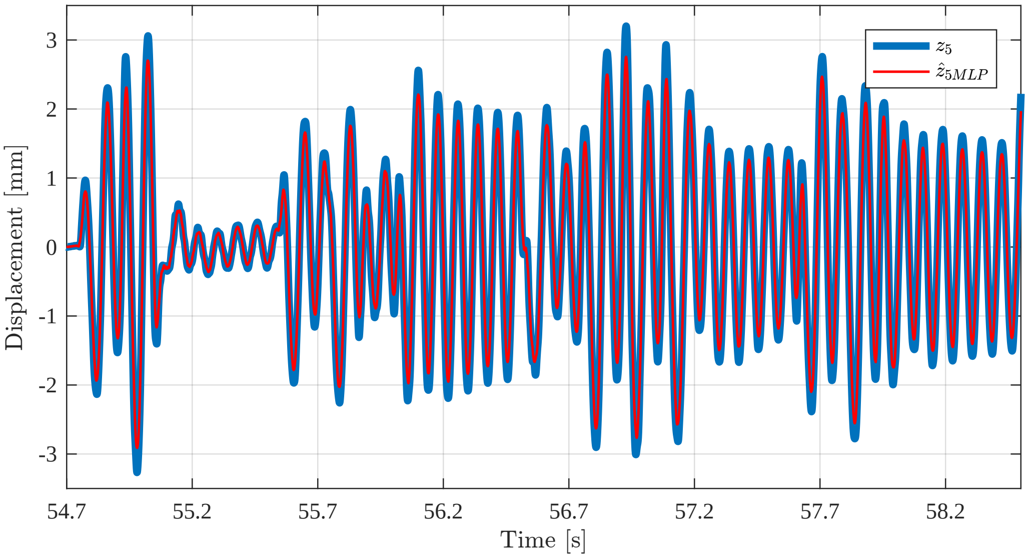

Figure 6.22: Multilayer perceptron network validation results [TS15b]

Table 6.6: SVM-based model vs. MLP-model

	SVM-based model	MLP model
Structure	650 support vector	10 neurons
RSE training	0.0897	0.0646
RSE test	0.1051	0.1345
Training time [s]	0.38	2.59

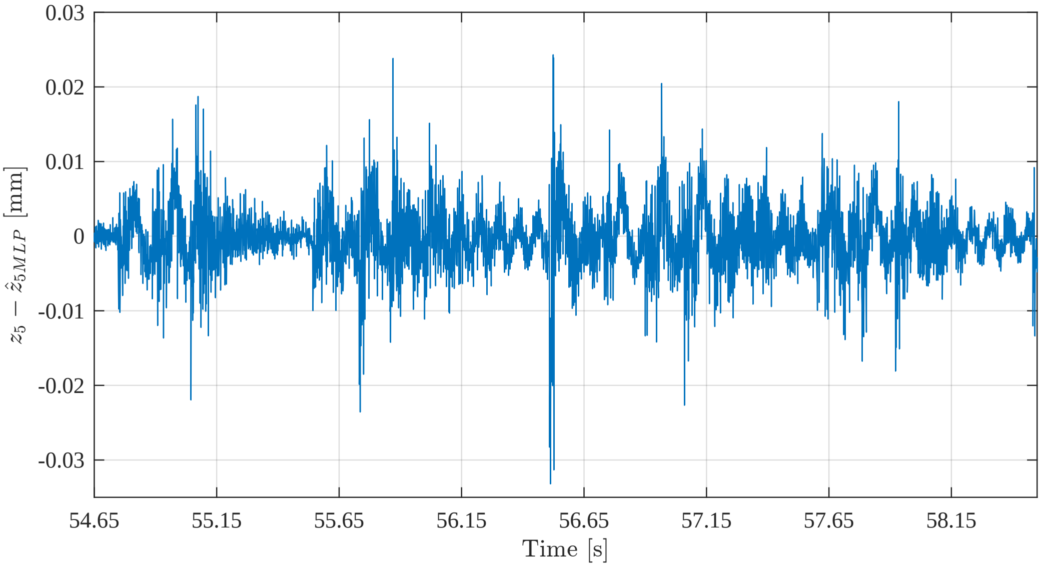

Figure 6.23: Multilayer perceptron network prediction error [TS15b]

6.2.3 Validation of the SVM-based controller

Based on the direct inverse model control using SVM algorithm a suitable is obtained. This SVM-based controller is implemented using dSPACE system. The controller is compared wit LQR controller with $R = 0.01$ and $Q = diag(0.50.1051)$. The comparison results are shown in figures 6.24, 6.25, and 6.26. In figure 6.24 it can be observed that the LQR-controller has higher steady state error that the SVM-based controller. The beam tip oscillations is shown in figure 6.25, it can be noted that the LQR-controller manage to reduce that beam tip oscillation faster that the SVM-based controller, also it can be noted that the LQR-controller has a higher steady state error that the SVM-based controller. The actuating signals of both controller are shown in figure 6.26, It can be observed that the LQR-controller uses less energy than the SVM-based controller.

To overcome the steady state error of the LQR-controller, an integral element is added to the closed-loop the result is an LQR-controller with integral action [YW72]. The results of the comparison of the SVM-based controller with LQR-controller with integral action are shown in figures 6.27,6.28, and 6.29. With the addition of the integral element it can be noted that the LQR-controller has better performance the SVM-based controller. It can be noted that the steady state error of the LQR-controller with internal action is zero. In figure 6.28 it can be observed that the LQR-controller with internal action has less oscillation than the SVM-based controller. As for the actuating signals are shown in 6.29, it can be observed the the LQR-controller with internal action has less energy than the SVM-based controller.

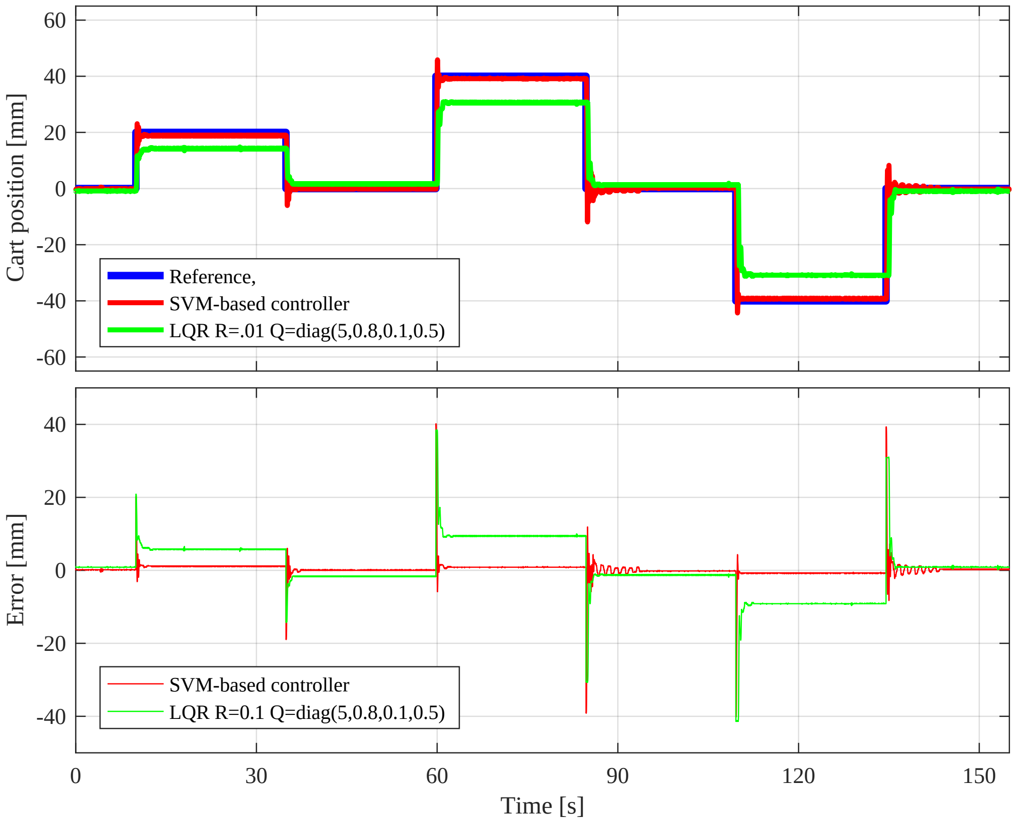

Figure 6.24: SVM-based controller vs. LQR controller: Cart displacement

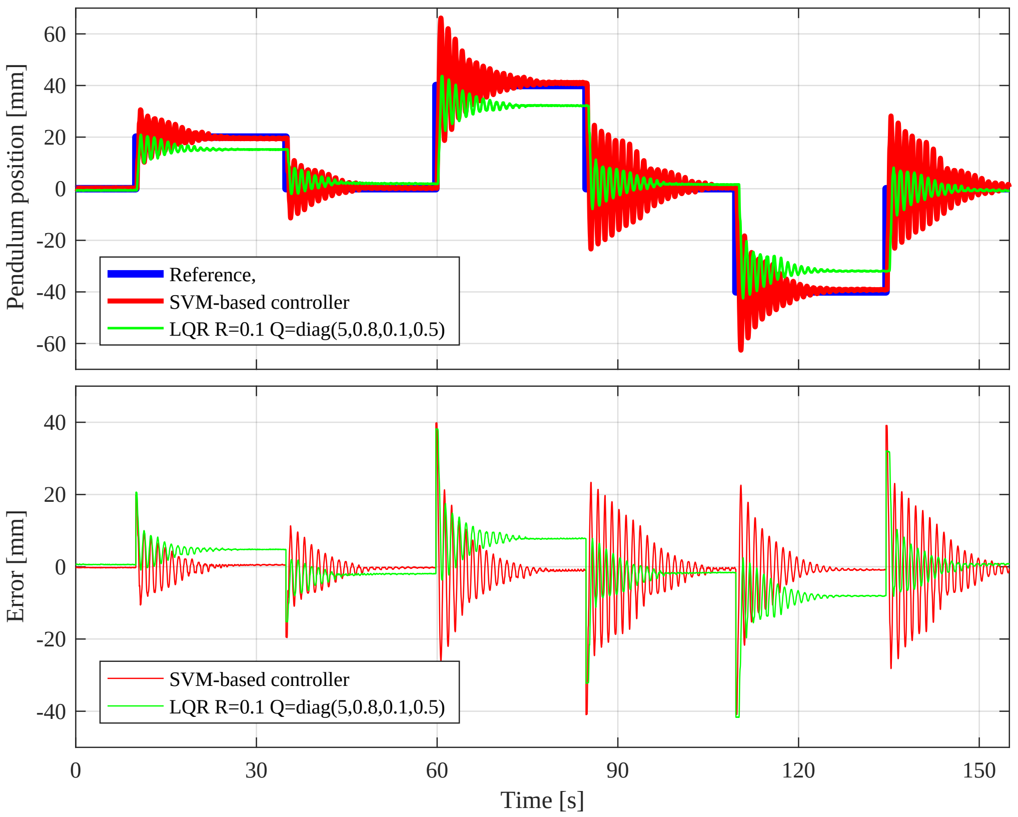

Figure 6.25: SVM-based controller vs. LQR controller: Beam tip displacement

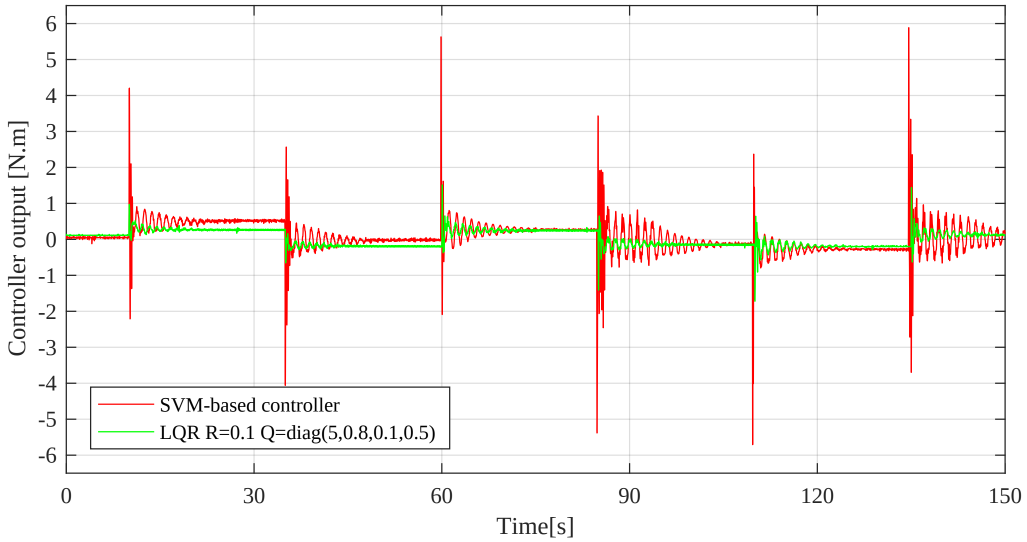

Figure 6.26: SVM-based controller vs. LQR controller: Actuating signal

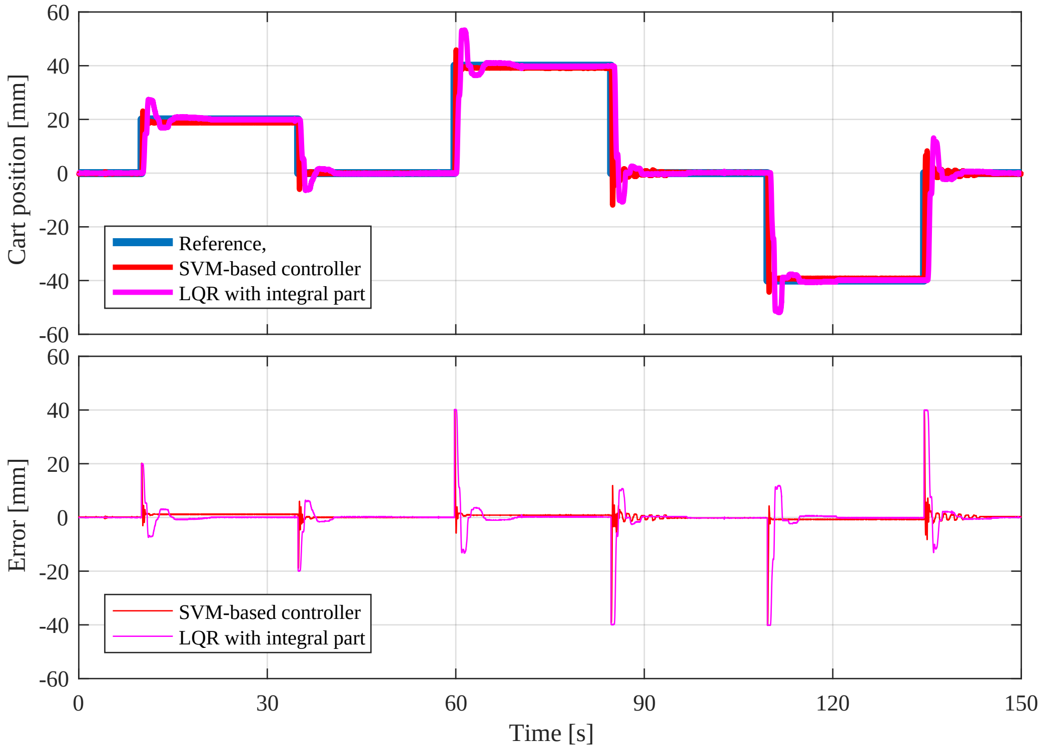

Figure 6.27: SVM-based controller vs. LQR controller with integral action: Cart displacement

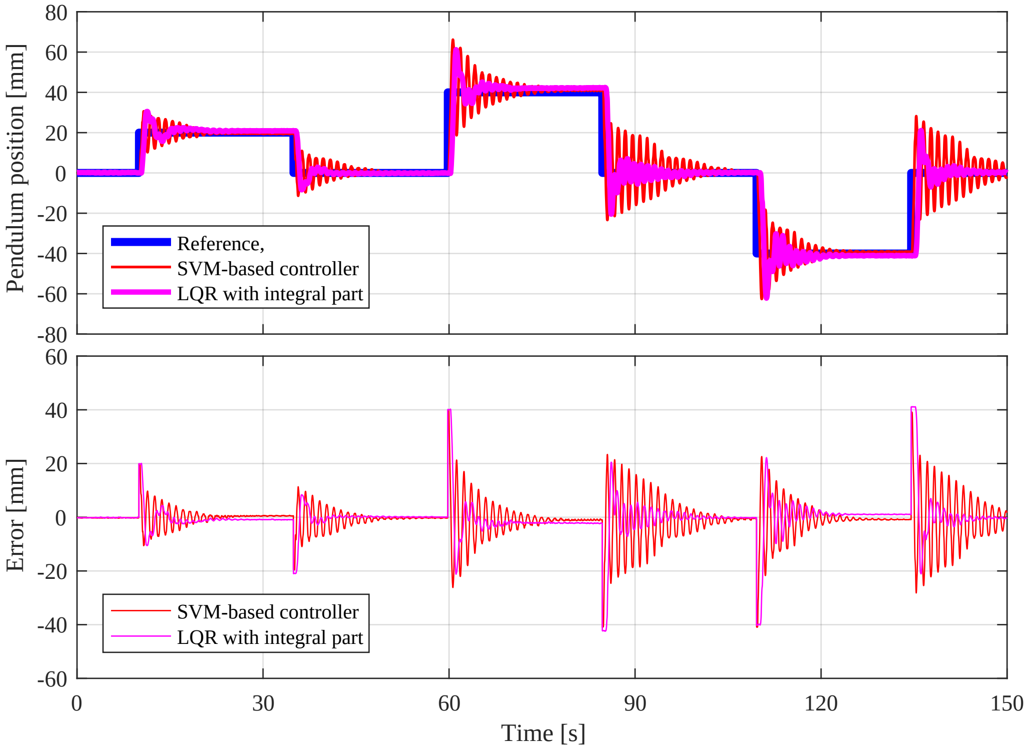

Figure 6.28: SVM-based controller vs. LQR controller with integral action: Beam tip displacement

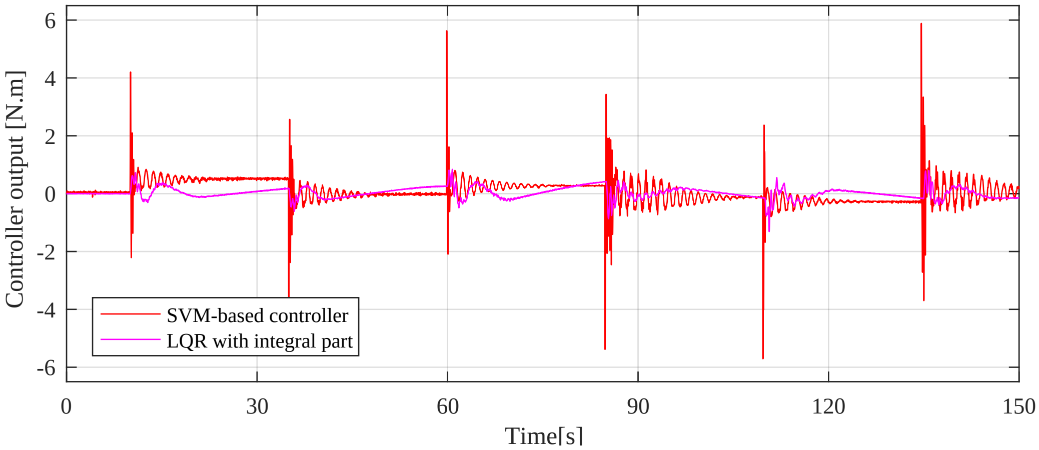

Figure 6.29: SVM-based controller vs. LQR controller with integral action: Actuating signal

In figure 6.30 different LQR-controllers and the SVM-based controller are evaluated using the criteria developed in [LS09]. It can be observed that the LQR-controller are more energy efficient than the SVM-based controller, on the hand, the SVM-based controller has better accuracy than the LQR-controllers.

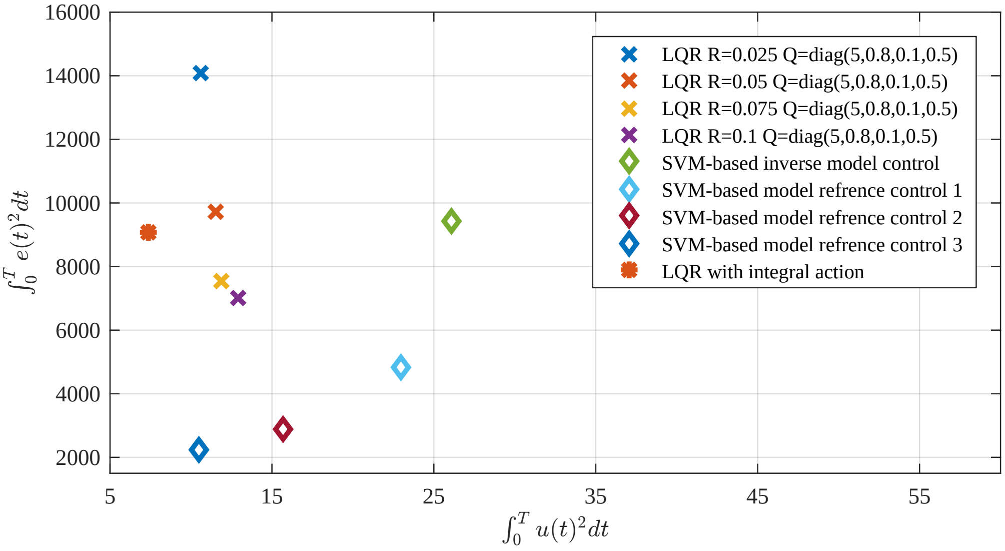

Figure 6.30: Performance SVM-based controller vs. LQR controller

6.3 Summary

In this chapter, modeling of the flexible cantilever beam, which can be a representation of manipulator link or aircraft wing, using experimental data is presented. Identification of dynamic system based on LS-SVM is summarized, establishing that the LS-SVM algorithm can achieve global and unique solution due to the fact the LS-SVM uses quadratic programming to obtain the solution, additionally LS-SVM algorithm has two tuning parameters which make the training process comfortable. The automated system identification approach based on LS-SVM is proposed. This approach is used to find the best model that fit an unknown dynamic system. To demonstrate the efficiency of the proposed approach, a flexible cantilever beam based on experimental data is used. At first the data obtained from test flexible cantilever beam bench was analyzed. It can be observed that the data contain information about the first three modes of the beam. The acquired model using the automated system identification algorithm based on LS-SVM demonstrates high accuracy.

7 Summary and Outlook

7.1 Summary

In chapter 1, the focus of this thesis which is a soft computing-based controller for complex mechanical system was given. It was provided that a soft computing-based controller is a controller that able to develop a suitable control strategy by direct interaction with the targeted dynamic system using machine learning algorithms. It was also established that the main goal of a soft computing-based controller is to improve the performance of the close loop system. Additionally to provide the controller with adaptation ability and to reduce the need for mathematical modeling based on physical laws.

Chapter 2 five different soft computing algorithms used in the field of modeling and controlling dynamic systems were discussed. These algorithms were Multi-Layer Perceptron (MLP) network, Support Vector Machine (SVM), Gaussian process, Radial Basis Function (RBF) and Fuzzy Inference System (FIS). The basic mathematical description of each algorithm was given. Additionally the most recent applications in modeling and controlling of dynamic system were summarized. In chapter 3 a systematic evaluation of the five algorithms was proposed. The goal of the evaluation was to provide quantitative measure of the performance of soft computing algorithms when used in modeling and controlling dynamic systems. Based on the evaluation the SVM algorithm was selected to be the core learning algorithm for the soft computing-based controller.

The structure of the soft computing-based controller was introduced in chapter 4. The main goal of the contrive controller is to enhance the performance of multi-input and multi-output (MIMO) system. Here two main modules with their functions were described. The effect of data-set size on dynamic systems model based on SVM algorithm was discussed. The cross-validation approach was selected to overcome the bias-variance dilemma.

The verification of the soft computing-based controller was done using an elastic cantilever beam modeled using Finite element method in chapter 5. An elastic cantilever beam can be representation of flexible single-link manipulator or aircraft wing. In the core of the modeling unit, an automatic system identification algorithm which allows a systematic modeling approach of dynamic systems was presented. The results showed that the system dynamic model using SVM algorithm is accurate. As for the SVM-based observer the results showed it has good estimation results in comparison with respect to different Kalman-Bucy observers. The sensitivity to parameters variations analysis showed that the SVM-based observer has better performance than Kalman-Bucy observer. The SVM-based controller was used to control the vibration of the cantilever beam; the results showed that the model reference controller using SVM has a similar performance to LQR-controller.

In chapter 6 the validation of the soft computing-based controller was carried on using the elastic cantilever beam test rig and the inverted cantilever beam test rig. The states estimation using SVM-based observer of the elastic cantilever beam test rig was successful and accurate, the results were compared to a Kalman observer. Modeling of the elastic cantilever beam using the SVM algorithm showed very good accuracy. The control part of the soft computing-based controller was performed on the inverted cantilever beam test rig; the results show that a suitable control strategy can be obtained. The analysis showed that due to limitation on the test rig computational power the performance increase of the SVM-based controller was not possible.

7.2 Outlook

According to the results achieved in this work it can be concluded that further research is required in the following topics:

- The elastic cantilever test rig has no actuator due to this only the modeling unit of the soft computing-based controller is evaluated on this test rig. Installing of piezoelectric actuators at different positions will allows better evaluation and comparison of the soft computing-based controller concept.
- The sensors fixed on the inverted cantilever test rig are limited on number due to that only the control unit is evaluated on this test rig. Increasing the number of sensors will allow the implementation of the modeling unit of the soft computing-based controller thus the whole soft computing-based controller can be evaluated.
- The data acquiring system used has relatively poor computational power; this does not allow for online adaptation of the controller and even limit the number of support vectors used to build the SVM-based controller.

Bibliography

[AEGAA15] M. Abd-Elmeged, E. Gaber, H. Ashraf, and F. Asmaa. Self-tuning DC servo motor design based on radial basis function neural network. *The International Journal of Control, Automation, and Systems (IJCAS)*, 4(2):1–6, 2015.

[AF11] T. Aboueldahab and M. Fakhreldin. Identification and adaptive control of dynamic nonlinear systems using sigmoid diagonal recurrent neural network. *Intelligent Control and Automation*, 2(3):176–181, 2011.

[AK11] K. Ažman and J. Kocijan. Dynamical systems identification using Gaussian process models with incorporated local models. *Engineering Applications of Artificial Intelligence*, 24(2):398–408, 2011.

[ALS93] R. Asim, S. Lark, and M. Somnath. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. *Neural Networks*, 6(4):535–545, 1993.

[AMAH$^+$15] K. Al-Mutib, F. Abdessemed, R Hedjar, M. Alsulaiman, M. Bencherif, M. Faisal, M. Algabri, and M. Mekhtiche. Mobile robot nonlinear feedback control based on Elman neural network observer. *Advances in Mechanical Engineering*, 7(12):1–14, 2015.

[ARM09] D. Anuradha, G. Reddy, and J. Murthy. Direct inverse neural network control of a continuous stirred tank reactor (CSTR). In *Proceedings of the International Multiconference of Engineers and Computer Scientists*, volume 2, pages 1–5. International Association of Engineers (IAENG), 2009.

[AS06] E. Ahle and D. Söffker. A cognitive-oriented architecture to realize autonomous behavior-part ii: Application to mobile robotics. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2221–2227. Institute of Electrical and Electronics Engineers (IEEE), 2006.

[ASKS13] A. Alexandridis, M. Stogiannos, A. Kyriou, and H. Sarimveis. An offset-free neural controller based on a non-extrapolating scheme for approximating the inverse process dynamics. *Journal of Process Control*, 23(7):968–979, 2013.

[ASL$^+$14] A. Alexandridis, M. Stogiannos, A. Loukidis, K. Ninos, E. Zervas, and H. Sarimveis. Direct versus indirect neural control based on radial basis function networks. In *IEEE Computer Science and*

Electronic Engineering Conference (CEEC), pages 91–96. Institute of Electrical and Electronics Engineers (IEEE), 2014.

[BB11] O. Bouhali and H. Boudjedir. Neural network control with neuro-sliding mode observer applied to quadrotor helicopter. *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 1(1):24–28, 2011.

[Bey15] S. Beyhan. Adaptive fuzzy terminal sliding-mode observer with experimental applications. *International Journal of Fuzzy Systems*, 1(1):1–10, 2015.

[BHD14] M. Beale, M. Hagan, and H. Demuth. *Neural network toolbox: Getting started guide.* Mathworks, 2014.

[CBG90] S. Chen, S. Billings, and P. Grant. Non-linear system identification using neural networks. *International Journal of Control*, 51(6):1191–1214, 1990.

[CDJQYDB10] L. Cheng-Dong, Y. Jian-Qiang, Y. Yi, and Z. Dong-Bin. Inverse control of cable-driven parallel mechanism using type-2 fuzzy neural network. *Acta Automatica Sinica*, 36(3):459–464, 2010.

[CFCT98] J. Chia-Feng and L. Chin-Teng. An online self-constructing neural fuzzy inference network and its applications. In *IEEE Transactions on Fuzzy Systems*, volume 6, pages 12–32. Institute of Electrical and Electronics Engineers (IEEE), 1998.

[CGH08] X. Cao, S. Gan, and D. He. Research and simulation of human-simulated intelligent controller for time-varying processes with long time-delay. In *IEEE World Congress on Intelligent Control and Automation*, pages 3284–3287. Institute of Electrical and Electronics Engineers (IEEE), 2008.

[Che90] D. Chester. Why two hidden layers are better than one. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, volume 1, pages 265–268. Institute of Electrical and Electronics Engineers (IEEE), 1990.

[CHLH09] S. Chen, X. Hong, B. Luk, and C. Harris. Non-linear system identification using particle swarm optimisation tuned radial basis function models. *International Journal of Bio-Inspired Computation*, 1(4):246–258, 2009.

[Chu07] C. Chuang. Fuzzy weighted support vector regression with a fuzzy partition. In *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, volume 37, pages 630–640. Institute of Electrical and Electronics Engineers (IEEE), 2007.

[CLC13] L. Chan, Yi Liu, and J. Chen. Nonlinear system identification with selective recursive Gaussian process models. *Industrial & Engineering Chemistry Research*, 52(51):18276–18286, 2013.

[Cob13] R. Coban. A context layered locally recurrent neural network for dynamic system identification. 26(1):241–250, 2013.

[DLHJ$^+$13] J. Deng, K. Li, E. Harkin-Jones, M. Fei, and S. Li. A novel two stage algorithm for construction of RBF neural models based on A-optimality criterion. In *IEEE International Conference on Natural Computation*, pages 1–7. Institute of Electrical and Electronics Engineers (IEEE), 2013.

[DM14] E. Drayabeigi and G. Markadeh. Speed control of brushless DC motors using emotional intelligent controller. *Intelligent Systems in Electrical Engineering*, 4(4):2921–2925, 2014.

[DZR11] X. Dong, Y. Zhao, and G. Rui. Internal model control based on RBF neural network inverse system decoupling in a 3-DOF helicopter system. In *IEEE World Congress on Intelligent Control and Automation (WCICA)*, pages 570–574. Institute of Electrical and Electronics Engineers (IEEE), 2011.

[Fre90] M. Frean. The upstart algorithm: A method for constructing and training feedforward neural networks. *Neural Computation*, 2(2):198–209, 1990.

[GD95] R. Garcia and C. D'attellis. Trajectory tracking in nonlinear systems via nonlinear reduced-order observers. *International Journal of Control*, 62(3):685–715, 1995.

[Ged11] M. Gedikpinar. The speed control of DC motors with support vector machine. *Przeglad Elektrotechniczny*, 87(5):269–271, 2011.

[GHJ03] M. Gupta, N. Homma, and L. Jin. *Static and dynamic neural networks: From fundamentals to advanced theory.* John Wiley & Sons, Inc., 1st edition, 2003.

[GL02] G. Gregorcic and G. Lightbody. Gaussian processes for modelling of dynamic non-linear systems. In *Proceedings of the Irish Signals and Systems Conference*, pages 141–147. Department of Electrical & Electronic Engineering, University College Cork, 2002.

[GM12] A. Gholami and A. Markazi. A new adaptive fuzzy sliding mode observer for a class of MIMO nonlinear systems. *Nonlinear Dynamics*, 70(3):2095–2105, 2012.

[GML10] Y. Ge, S. Ma, and X. Luo. Direct inverse model control based on a new improved CMAC neural network. In *Advanced Intelligent Computing Theories and Applications*, pages 25–32. Springer, 2010.

[Gol89] D. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley Longman Publishing Co., Inc., 1st edition, 1989.

[GXC06] H. Gong, H. Xu, and F. Chowdhury. A neuro-augmented observer for a class of nonlinear systems. *International Joint Conference on Neural Networks*, 1(1):2497–2500, 2006.

[Har94] C. Harris. *Advances in intelligent control.* CRC Press, 1994.

[Has11] K. Hassan. Self learning of ANFIS inverse control using iterative learning technique. *International Journal of Computer Applications*, 21(8):24–29, 2011.

[Hay98] S. Haykin. *Neural networks: A comprehensive foundation.* Prentice Hall PTR, 2nd edition, 1998.

[Hay08] S. Haykin. *Neural networks and learning machines.* Prentice Hall, 3rd edition, 2008.

[HCL10] C. Hsu, C. Chang, and C. Lin. A practical guide to support vector classification. Technical report, National Taiwan University, Department of Computer Science, 2010.

[HCQ10] H. Han, Q. Chen, and J. Qiao. Research on an online self-organizing radial basis function neural network. *Neural Computing and Applications*, 19(5):667–676, 2010.

[HK78] A. Hagan and J. Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society*, 1(1):1–42, 1978.

[HS96] C. Hwang and F. Sung. Neuro-observer controller design for nonlinear dynamical systems. In *Proceedings of the IEEE Conference on Decision and Control*, volume 3, pages 3310–3315. Institute of Electrical and Electronics Engineers (IEEE), 1996.

[HS14] M. Hu and Z. Sun. Multimodel nonlinear predictive control with gaussian process model. In *Unifying Electrical Engineering and Electronics Engineering*, pages 1895–1902. Springer, 2014.

[Hub13] M. Huber. Recursive gaussian process regression. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3362–3366. Institute of Electrical and Electronics Engineers (IEEE), 2013.

[HW14] M. Han and X. Wang. Weighted support vector echo state machine for multivariate dynamic system modeling. In *IEEE American Control Conference*, pages 4824–4828. Institute of Electrical and Electronics Engineers (IEEE), 2014.

[IID06] R. Ismail, A. Ismail, and I. Darus. Identification algorithms of flexible structure using neural networks. In *Proceeding of the IEEE Student Conference on Research and Development,*, pages 162–168. Institute of Electrical and Electronics Engineers (IEEE), 2006.

[JAL10] M. Javan, A. Arami, and C. Lucas. Imitative learning based emotional controller for unknown systems with unstable equilibrium. *International Journal of Intelligent Computing and Cybernetics*, 3(2):334–359, 2010.

[Jan93] J. Jang. ANFIS: Adaptive-network-based fuzzy inference system. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 23, pages 665–685. Institute of Electrical and Electronics Engineers (IEEE), 1993.

[JD13] N. Jalil and I. Darus. Non-parametric neuro-model of a flexible beam structure. In *IEEE Symposium on Computers & Informatics*, pages 54–50. Institute of Electrical and Electronics Engineers (IEEE), 2013.

[JDM12] N. Jalil, I. Darus, and M. Mohamad. Neuro-fuzzy identification of flexible beam structure. In *IEEE Conference on Control, Systems & Industrial Informatics*, pages 185–190. Institute of Electrical and Electronics Engineers (IEEE), 2012.

[JFD13] N. Jalil, M. Fadil, and I. Darus. Intelligent PID controller using iterative learning algorithm for active vibration controller of flexible beam. In *IEEE Symposium on Computers and Informatics*, pages 80–85. Institute of Electrical and Electronics Engineers (IEEE), 2013.

[JRAL09] M. Javan-Roshtkhari, A. Arami, and C. Lucas. Emotional control of inverted pendulum system: A soft switching from imitative to emotional learning. In *IEEE International Conference on Autonomous Robots and Agents*, pages 651–656. Institute of Electrical and Electronics Engineers (IEEE), 2009.

[JRP$^+$15] P. Jagtap, P. Raut, G. Pillai, F. Kazi, and N. Singh. Extreme-ANFIS: A novel learning approach for inverse model control of nonlinear dynamical systems. In *IEEE International Conference on Industrial Instrumentation and Control (ICIC)*, pages 718–723. Institute of Electrical and Electronics Engineers (IEEE), 2015.

[JS01] R. Jeyoung and W. Sangchul. Partitioning of linearly transformed input space in adaptive network based fuzzy inference system. *IEICE Transactions on Information and Systems*, 84(1):213–216, 2001.

[JSX15] F. Jinhui, J. Songmin, and L. Xiuzhi. Direct adaptive control based on improved RBF neural network for omni-directional mobile robot. *International Conference on Mechatronics, Electronic, Industrial and Control Engineering*, 1(1):1108–1112, 2015.

[KB00] Y. Kwon and H. Bang. *The finite element method using MATLAB.* Mechanical and Aerospace Engineering Series. CRC Press, 2000. July.

[Kec01] V. Kecman. *Learning and soft computing: Support vector machines, neural networks, and fuzzy logic models.* MIT press, 2001.

[KGBMS05] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith. Dynamic systems identification with gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424, 2005.

[KI83] A. Krener and A. Isidori. Linearization by output injection and nonlinear observers. *Systems & Control Letters*, 3(1):47–52, 1983.

[KK98] N. Kasabov and R. Kozma. Introduction: Hybrid intelligent adaptive systems. *International Journal of Intelligent Systems*, 13(6):453–454, 1998.

[KL09] C. Kim and R. Langari. Target tracking control of a mobile robot using a brain limbic system based control strategy. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, pages 5059–5064. Institute of Electrical and Electronics Engineers (IEEE), 2009.

[KMD$^+$96] C. Kambhampati, S. Manchanda, A. Delgado, G. Green, K. Warwick, and M. Tham. The relative order and inverses of recurrent networks. *Automatica*, 32:117–123, 1996.

[Kod14] Y. Kodratoff. *Introduction to machine learning.* Morgan Kaufmann, 2014.

[KP14] H. Khalil and L. Praly. High-gain observers in nonlinear feedback control. *International Journal of Robust and Nonlinear Control*, 24(6):993–1015, 2014.

[KS14] A. Kroll and H. Schulte. Benchmark problems for nonlinear system identification and control using soft computing methods: Need and overview. *Applied Soft Computing*, 25:496–513, 2014.

[LCLI97] Z. Li, Q. Chen, X. Li, and H. Inooka. Human simulating intelligent control and its application to swinging-up of cart-pendulum. In *Proceeding of the IEEE International Workshop on Robot and Human Communication (Ro-Man)*, volume 97, pages 218–223. Institute of Electrical and Electronics Engineers (IEEE), 1997.

[LF88] A. Lapedes and R. Farber. How neural nets work. In *Neural Information Processing Systems*, pages 442–456. American Institute of Physics, 1988.

[LF15] C. Lu and J. Fei. Adaptive backstepping control of MEMS gyroscope based on neural state observer. In *IEEE Control Conference (CCC)*, pages 377–382. Institute of Electrical and Electronics Engineers (IEEE), 2015.

[LJ02] J. Li and Y. Jinshou. Nonlinear hybrid adaptive inverse control using neural fuzzy system and its application to CSTR systems. In *Proceedings of the IEEE World Congress on Intelligent Control and Automation*, volume 3, pages 1896–1900. Institute of Electrical and Electronics Engineers (IEEE), 2002.

[LK90] J. Leonard and M. Kramer. Improvement of the backpropagation algorithm for training neural networks. *Computers & Chemical Engineering*, 14(3):337– 341, 1990.

[LLSŻ08] J. Lian, Y. Lee, S. Sudhoff, and S. Żak. Self-organizing radial basis function network for real-time approximation of continuous-time dynamical systems. In *IEEE Transactions on Neural Networks*, volume 19, pages 460–474. Institute of Electrical and Electronics Engineers (IEEE), 2008.

[LMSL00] D. Leith, R. Murray-Smith, and W. Leithead. Nonlinear structure identification: A Gaussian process prior/velocity-based approach. Technical report, University of Glasgow, Department of Computing Science, 2000.

[LNG95] J. Liang, P. Nikiforuk, and M. Gupta. Approximation of discrete-time state-space trajectories using dynamic recurrent neural networks. In *IEEE Transactions on Automatic Control*, volume 40, pages 1266–1270. Institute of Electrical and Electronics Engineers (IEEE), 1995.

[LPB06] K. Li, J. Peng, and E. Bai. A two-stage algorithm for identification of nonlinear dynamic systems. *Automatica*, 42(7):1189–1197, 2006.

[LPCL15] C. Lin, C. Peng, C. Chen, and H. Lin. A self-organizing recurrent wavelet neural network for nonlinear dynamic system identification. *Applied Mathematics & Information Sciences*, 9(1L):125–132, 2015.

[LRH13] T. Ling, M. Rahmat, and A. Husain. ANFIS modeling and direct anfis inverse control of an electro-hydraulic actuator system. In *IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 370–375. Institute of Electrical and Electronics Engineers (IEEE), 2013.

[LS09] Y. Liu and D. Söffker. Robust control approach for input-output linearizable nonlinear systems with modeling errors based on high-gain PI-observer. In *Vienna International Conference on Mathematical Modeling*, pages 193–199. University of Vienna, 2009.

[LSB09] Z. Lu, J. Sun, and K. Butts. Linear programming support vector regression with wavelet kernel: A new approach to nonlinear dynamical systems identification. *Mathematics and Computers in Simulation*, 79(7):2051–2063, 2009.

[LSC08] C. Liu, X. Sun, and J. Chen. Application of human-simulating intelligent control in heating the super-high pressure kettle. In *IEEE World Congress on Intelligent Control and Automation*, pages 7074–7078. Institute of Electrical and Electronics Engineers (IEEE), 2008.

[LSS04] C. Lucas, D. Shahmirzadi, and N. Sheikholeslami. Introducing BELBIC: Brain emotional learning based intelligent controller. *Intelligent Automation & Soft Computing*, 10(1):11–21, 2004.

[LTL12] Y. Li, S. Tong, and Y. Li. Observer-based adaptive fuzzy backstepping control of MIMO stochastic nonlinear strict-feedback systems. *Nonlinear Dynamics*, 67(2):1579–1593, 2012.

[LX12] F. Liu and X. Xue. Constructing kernels by fuzzy rules for support vector regressions. *International Journal of Innovative Computing, Information and Control*, 8(7):4811–4822, 2012.

[Mar63] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[MB00] J. Moren and C. Balkenius. A computational model of emotional learning in the amygdala. *From Animals to Animats*, 6:115–124, 2000.

[ME14] T. Mahmoud and L. Elshenawy. Direct adaptive control based on LS-SVM inverse model for nonlinear systems. In *IEEE International Conference On Methods and Models in Automation and Robotics (MMAR)*, pages 693–698. Institute of Electrical and Electronics Engineers (IEEE), 2014.

[ML06] Y. Mo and H. Liu. The design of neural network direct inverse controller based on complex particle swarm optimization algorithm. In *IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 382–388. Institute of Electrical and Electronics Engineers (IEEE), 2006.

[MM95] H. Michalska and D. Mayne. Moving horizon observers and observer-based control. In *IEEE Transactions on Automatic Control*, volume 40, pages 995–1006. Institute of Electrical and Electronics Engineers (IEEE), 1995.

[MSJS99] R. Murray-Smith, T. Johansen, and R. Shorten. On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. In *IEEE European Control Conference (ECC)*, pages 3569–3574. Institute of Electrical and Electronics Engineers (IEEE), 1999.

[MSP15] P. Madhavan, A. Sinha, and J. Pandian. Direct inverse control of a 3 DOF tandem helicopter model using wavelet neural network. *International Journal of Applied Engineering Research*, 10(12):31257–31267, 2015.

[Nel01] O. Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models.* Springer, 2001.

[NK15] S. Nagarajan and B. Karthikeyan. Control of interacting level process under ssensor failure conditions using coactive adaptive neuro-fuzzy observer. *International Journal of New Technologies in Science and Engineering*, 2(4):242–251, 2015.

[NP90] K. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. In *IEEE Transactions on Neural Networks*, volume 1, pages 4–27. Institute of Electrical and Electronics Engineers (IEEE), 1990.

[NRPH00] P. Nørgård, O. Ravn, Niels K. Poulsen, and L. Hansen. *Neural networks for modelling and control of dynamic systems- a practitioner's handbook.* Springer-London, 2000.

[OMTT05] K. Ogasawara, T. Murata, J. Tamura, and T. Tsuchiya. High performance control of permanent magnet synchronous motor based on magnetic energy model by sliding mode control. In *IEEE European Conference on Power Electronics and Applications*, pages 1–10. Institute of Electrical and Electronics Engineers (IEEE), 2005.

[PB03] V. Prasad and B. Bequette. Nonlinear system identification and model reduction using artificial neural networks. *Computers & Chemical Engineering*, 27(12):1741–1754, 2003.

[Pea95] B. Pearlmutter. Gradient calculations for dynamic recurrent neural networks: A survey. In *IEEE Transactions on Neural Networks*, volume 6, pages 1212–1228. Institute of Electrical and Electronics Engineers (IEEE), 1995.

[Poz98] A. Poznyak. Learning for dynamic neural networks. In *Yale Workshop on Adaptive and Learning System*, pages 38–47. Yale University, 1998.

[PPKM05] J. Park, G. Park, S. Kim, and C. Moon. Output-feedback control of uncertain nonlinear systems using a self-structuring adaptive fuzzy observer. *Fuzzy Sets and Systems*, 151(1):21–42, 2005.

[PS91] J. Park and I. Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[PŠ14] J. Prüher and M. Šimandl. Gaussian process based recursive system identification. In *Journal of Physics: Conference Series*, volume 570, page 012002. IOP Publishing, 2014.

[PSPW00] A. Poznyak, E. Sanchez, O. Palma, and Y. Wen. Robust asymptotic neuro observer with time delay term. In *Proceedings of the IEEE International Symposium on Intelligent Control*, volume 26, pages 19–24. Institute of Electrical and Electronics Engineers (IEEE), 2000.

[PSY01] A. Poznyak, E. Sanchez, and W. Yu. *Differential neural networks for robust nonlinear control: Identification, state estimation and trajectory tracking.* World Scientific, 2001.

[PSY15] Y. Pan, T. Sun, and H. Yu. Peaking-free output-feedback adaptive neural control under a nonseparation principle. In *IEEE transactions on neural networks and learning systems*, volume 26, pages 3097–3108. Institute of Electrical and Electronics Engineers (IEEE), 2015.

[PYHP06] E. Purwanto, S. Yukihiko, P. Herry, and G. Prabowo. The development of speed sensor observer for induction motor by using adaptive neuro fuzzy with back propagation learning method. *The International Conference on Properties and Applications of Dielectric Materials*, 1(1):787–792, 2006.

[QGL+15] C. Qi, F. Gao, H. Li, X. Zhao, and L. Deng. A neural network-based distributed parameter model identification approach for microcantilever. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 0(0):1–14, 2015.

[Ras96] C. Rasmussen. Evaluation of gaussian processes and other methods for non-linear regression. Technical report, University of Toronto, Department of Computer Science, 1996.

[Ras04] C. Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer, 2004.

[RHJ16] N. Ramli, M. Hussain, and B. Jan. Multivariable control of a debutanizer column using equation based artificial neural network model inverse control strategies. *Neurocomputing*, 194:135–150, 2016.

[RHW86] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, chapter 8, pages 318–362. MIT Press, 1986.

[RN15] S. Russell and P. Norvig. *Artificial intelligence: A modern approach.* Prentice-Hall, Egnlewood Cliffs, 3rd edition, 2015.

[RPMP15] R. Rajesh, R. Preethi, P. Mehata, and B. Pandian. Artificial neural network based inverse model control of a nonlinear process. In *IEEE International Conference on Computer, Communication and Control (IC4)*, pages 1–6. Institute of Electrical and Electronics Engineers (IEEE), 2015.

[Sch00] H. Schröder. Intelligent modelling, observation and control for nonlinear systems. In *Proceedings of the International Workshop on Advanced Motion Control*, 2000.

[SKP01] F. Schwenker, H. Kestler, and G. Palm. Three learning phases for radial-basis-function networks. *Neural networks*, 14(4):439–458, 2001.

[SM97] H. Su and T. McAvoy. Artificial neural networks for nonlinear process identification and control. *Nonlinear Process Control*, 1(7):371–428, 1997.

[SMSL$^+$03] E. Solak, R. Murray-Smith, W. Leithead, D. Leith, and C. Rasmussen. Derivative observations in gaussian process models of dynamic systems. *Conference on Neural Information Processing Systems*, 16(1):1–8, 2003.

[SS15] N. Sozhamadevi and S. Sathiyamoorthy. A probabilistic fuzzy inference system for modeling and control of nonlinear process. *Arabian Journal for Science and Engineering*, 40(6):1777–1791, 2015.

[SV99] J. Suykens and J. Vandewalle. Least squares support vector machine classifiers. In *Neural Processing Letters*, volume 9, pages 293–300, 1999.

[SV02] J. Suykens and J. Vandewalle. Recurrent least squares support vector machines. In *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, volume 47, pages 1109–1114. Institute of Electrical and Electronics Engineers (IEEE), 2002.

[SVdM96] J. Suykens, J. Vandewalle, and B. de Moor. *Artificial neural networks for modelling and control of non-linear systems.* Springer Science & Business Media, 1996.

[SW11] C. Sammut and G. Webb. *Encyclopedia of machine learning.* Springer Science & Business Media, 2011.

[SWC07] S. Shen, G. Wang, and H. Chen. Support vector machine based identification of inverse dynamic model of thermal system and its application. In *Challenges of Power Engineering and Environment*, pages 914–917. Springer, 2007.

[SWK15] H. Suprijono, W. Wahab, and B. Kusumoputro. Optimized direct inverse control to control altitude of a small helicopter. In *MATEC Web of Conferences*, volume 34. Édition Diffusion Presse (EDP) Sciences, 2015.

[SY93] M. Sugeno and T. Yasukawa. A fuzzy-logic-based approach to qualitative modeling. In *IEEE Transactions on Fuzzy Systems*, volume 1, pages 7–31. Institute of Electrical and Electronics Engineers (IEEE), 1993.

[SZL$^+$95] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, 31(12):1691–1724, 1995.

[TK08] S. Theodoridis and K. Koutroumbas. *Pattern recognition.* Academic Press, 4 edition, 2008.

[Tor89] A. Tornambe. Use of asymptotic observers having-high-gains in the state and parameter estimation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1791–1794. Institute of Electrical and Electronics Engineers (IEEE), 1989.

[TR10] A. Thammano and P. Ruxpakawong. Nonlinear dynamic system identification using recurrent neural network with multi-segment piecewise-linear connection weight. *Memetic Computing*, 2(4):273–282, 2010.

[TS85] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. In *IEEE Transactions on Systems, Man and Cybernetics*, volume SMC-15, pages 116–132. Institute of Electrical and Electronics Engineers (IEEE), 1985.

[TS14] H. Tamimi and D. Söffker. A soft-computing algorithm-based cognitive observer for an elastic robotic arm. *PAMM*, 14(1):933–934, 2014.

[TS15a] H. Tamimi and D. Söffker. Modeling of elastic robotic arm using a soft-computing algorithm. *IFAC-PapersOnLine*, 48(1):655–656, 2015.

[TS15b] H. Tamimi and D. Söffker. Modeling of flexible structures by means of least square support vector machine. In *Dynamic Systems and Control Conference*, number DSCC2015-9673, page V002T34A005. American Society of Mechanical Engineers (ASME), 2015.

[TS16a] H. Tamimi and D. Söffker. Control of dynamic systems using soft-computing algorithms with an example. *Applied Soft Computing*, 2016. submitted.

[TS16b] H. Tamimi and D. Söffker. Estimating of flexible structures state variables by means of least square support vector machine algorithm. *Nonlinear Dynamics*, 2016. submitted.

[TS16c] H. Tamimi and D. Söffker. Modeling of dynamic systems using soft-computing algorithms: Theory and example. *Nonlinear Dynamics*, 2016. submitted.

[Vap98a] V. Vapnik. *Statistical learning theory.* Wiley-Interscience, 1st edition, 1998.

[Vap98b] V. Vapnik. The support vector method of function estimation. *Nonlinear Modeling*, 1:55–85, 1998.

[Vap99] V. Vapnik. *The nature of statistical learning theory.* Information Science and Statistics. Springer, 2 edition, 1999.

[VGS96] V. Vapnik, S. Golowich, and A. Smola. On predicting rare classes with SVM ensembles in scene classification. *Advances in Neural Information Processing Systems*, 9:281–287, 1996.

[WCLS16] H. Wang, B. Chen, C. Lin, and Y. Sun. Observer-based adaptive neural control for a class of nonlinear pure-feedback systems. *Neurocomputing*, 171:1517–1523, 2016.

[Wei94] A. Weigend. On overfitting and the effective number of hidden units. In *Proceedings of the Connectionist Models Summer School*, volume 1, pages 335–342. Psychology Press, 1994.

[Wil98] C. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning in Graphical Models*, pages 599–621. Springer, 1998.

[WM11] S. Wang and B. Meng. Dynamic modeling method based on support vector machine. *Procedia Environmental Sciences*, 11:531–537, 2011.

[WPS07] H. Wang, D. Pi, and Y. Sun. Online SVM regression algorithm-based adaptive inverse control. *Neurocomputing*, 70(4):952–959, 2007.

[WPW^{+}15] P. Witczak, K. Patan, M. Witczak, V. Puig, and J. Korbicz. A neural network-based robust unknown input observer design: Application to wind turbine. *IFAC-PapersOnLine*, 48(21):263–270, 2015.

[WR16] M. Wang and X. Ren. Neural network observer based optimal tracking control for multi-motor servomechanism with backlash. In *Proceedings of the Chinese Intelligent Systems Conference*, pages 453–462. Springer, 2016.

[WT15] T. Wang and S. Tong. Direct inverse control of cable-driven parallel system based on type-2 fuzzy systems. *Information Sciences*, 310:1–15, 2015.

[WY04] X. Wang and M. Ye. Nonlinear dynamic system identification using least squares support vector machine regression. *Proceedings of International Conference on Machine Learning and Cybernetics*, 2:941–945, 2004.

[Xia97] M. Xiaomin. Inverse identification and closed-loop control of dynamic systems using neural networks. *Control Theory & Applications*, 14(6):829–836, 1997.

[XLX12] X. Xue, J. Lu, and W. Xiang. Nonlinear system identification with modified differential evolution and RBF networks. In *IEEE International Conference on Advanced Computational Intelligence*, pages 332–335. Institute of Electrical and Electronics Engineers (IEEE), 2012.

[XYWL10] Y. Xiaofang, W. Yaonan, S. Wei, and W. Lianghong. RBF networks-based adaptive inverse model control system for electronic throttle. In *IEEE Transactions on Control Systems Technology*, volume 18, pages 750–756. Institute of Electrical and Electronics Engineers (IEEE), 2010.

[XZZW09] W. Xie, Y. Zhu, Z. Zhao, and Y. Wong. Nonlinear system identification using optimized dynamic neural network. *Neurocomputing*, 72(13-15):3277–3287, 2009.

[YKS91] H. Yoshio, Y. Koichi, and H. Shimpei. Back-propagation algorithm which varies the number of hidden units. *Neural Networks*, 4(1):61–66, 1991.

[YL14] W. Yu and X. Li. Robust asymptotic neuro observer for unknown nonlinear systems. In *IEEE World Congress on Intelligent Control and Automation (WCICA)*, pages 2115–2120. Institute of Electrical and Electronics Engineers (IEEE), 2014.

[YLJH03] R. Yan, Y. Liu, R. Jin, and A. Hauptmann. On predicting rare classes with SVM ensembles in scene classification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 21–24. Institute of Electrical and Electronics Engineers (IEEE), 2003.

[YLX$^+$09] H. Yang, Z. Li, F. Xue, G. Luo, and Z. Zhong. Human-simulated intelligent control of brushless dc motor system. In *Proceedings of the IEEE Annual International Conference on Chinese Control and Decision Conference*, pages 329–334. Institute of Electrical and Electronics Engineers (IEEE), 2009.

[YM11] A. Yazdani and S. Mahmoudzadeh. Emotional learning as a cognitive approach for high performance control of permanent magnet stepper motor. *International Journal of Computer Technology and Applications*, 2(4):1008–1015, 2011.

[YNXF08] W. Yao-Nan and Y. Xiao-Fang. SVM approximate-based internal model control strategy. *Acta Automatica Sinica*, 34(2):172–179, 2008.

[YO14] S. Yilmaz and Y. Oysal. Nonlinear system modeling with dynamic adaptive neuro-fuzzy inference system. In *Proceedings of IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pages 205–211. Institute of Electrical and Electronics Engineers (IEEE), 2014.

[YP10] H. Sadoghi Yazdi and R. Pourreza. Unsupervised adaptive neural-fuzzy inference system for solving differential equations. *Applied Soft Computing*, 10(1):267–275, 2010.

[YW72] P. Young and J. Willems. An approach to the linear multivariable servomechanism problem. *International Journal of Control*, 15(5):961–979, 1972.

[YXPW11] H. Yu, T. Xie, S. Paszczynski, and B. Wilamowski. Advantages of radial basis function networks for dynamic system design. In *IEEE Transactions on Industrial Electronics*, volume 58, pages 5438–5450. Institute of Electrical and Electronics Engineers (IEEE), 2011.

[ZLZ08] Y. Zhai, H. Li, and Q. Zhou. Research on SVM algorithm with particle swarm optimization. In *Joint International Conference on Information Sciences*. Atlantis Press, 2008.

[Zur92] J. Zurada. *Introduction to artificial neural systems*. West St. Paul, 1st edition, 1992.

[ZZ09] H. Zhao and J. Zhang. Nonlinear dynamic system identification using pipelined functional link artificial recurrent neural network. *Neurocomputing*, 72(13):3046–3054, 2009.

This thesis is based on the results and development steps presented in the following previous publications:

Journal articles

[TS16a] H. Tamimi and D. Söffker: Control of dynamic systems using soft-computing algorithms with an example. In: *Applied Soft Computing*, 2016. submitted.

[TS16b] H. Tamimi and D. Söffker: Estimating of flexible structures state variables by means of least square support vector. In: *Nonlinear Dynamics*, 2016. submitted.

[TS16c] H. Tamimi and D. Söffker: Modeling of dynamic systems using soft-computing algorithms: Theory and example. In *Nonlinear Dynamics*, 2016. submitted.

Conference papers

[TS14] H. Tamimi and D. Söffker: A soft-computing algorithm-based cognitive observer for an elastic robotic arm. *PAMM*, 14(1):933-934, 2014.

[TS15a] H. Tamimi and D. Söffker: Modeling of elastic robotic arm using a soft-computing algorithm. In: *IFAC-PapersOnLine*, 48(1):655-656, 2015.

[TS15b] H. Tamimi and D. Söffker: Modeling of flexible structures by means of least square support vector machine. In: *Dynamic Systems and Control Conference*, number DSCC2015-9673, page V002T34A005., American Society of Mechanical Engineers (ASME), 2015.

In the context of research projects at the Chair of Dynamics and Control, the following student thesis has been supervised by Hammam Tamimi and Univ.-Prof. Dr.-Ing. Dirk Söffker. Development steps and results of the research projects and the student theses are integrated with each other and hence are also part of this thesis.

[Gun14] T. Gunder, Literaturrecherche, Programmierung sowie Dokumentation von auf Gauß-Prozessen basierten Identifikations- und Vorhersagemethoden. Bachelor Thesis, Bachelor Thesis, 2014.