Application of Local Model Networks to Nonlinear System Identification and Control

Dissertation submitted to the Department of Electrical Engineering and Information Sciences Ruhr-Universität Bochum for the degree of Doktor-Ingenieur

by
Abid Ali
Haroonabad, Pakistan

Bochum 2003
Dissertation submitted on 15.01.2003
Oral examination held on 23.05.2003

Referees: Prof. Dr.-Ing C. Schmid (dissertation supervisor)
           Prof. Dr.-Ing. H.D. Fischer
Acknowledgments

This dissertation is the result of my research activities at the Institute of Automation and Computer Control, Department of Electrical Engineering and Information Sciences, Ruhr-Universität Bochum.

I express my heartiest gratitude to my supervisor Professor Christian Schmid for giving me the opportunity to work with him, for his great interest, encouragement and support in all matters and phases of this work.

I would like to thank Prof. H. D. Fischer for his interest in my research and willingness to act as referee for this dissertation. I appreciate the valuable and constructive suggestions made by him. My thanks are due to Prof. Heinz Unbehauen and Prof. Jan Lunze for their kind support and guidance throught my research activities at the Institute of Automation and Computer Control.

I owe a lot to my colleagues at the institute. They always helped me in one way or other, through fruitful discussions, suggestions and other favors. Special thanks in this regard go to Dr Johannes Dastych, Dr Thomas Junge, Prof. Herbert Werner, Dr Ulfur Halldorsson, Dr Rainer Pickhardt, Dipl.-Ing. Carsten Fritsch, Dr Christian Bohn, Dr Torsten Knohl, Dipl.-Ing. Jörg Uhlig, Dipl.-Ing. Mohammed Ashfaq, Dr Adrian Gambier and Dr Ferry Caswara.

Finally my heartfelt gratitude is due to my wife Sumera and children Usama and Hira. Without their love, support and understnding, I would not have succeeded to complete this dissertation.

Bochum, May 2003

Abid Ali
To my parents and teachers
Contents

Nomenclature V1

Einleitung X1

1 Introduction 1

2 Local Model Networks 10
   2.1 Basic Structure and Operating Regions 10
   2.2 Identification of Local Linear Input-Output Models 14
      2.2.1 Model Representation 14
      2.2.2 Dynamics of a Local Linear Model 15
   2.3 Learning Techniques 16
      2.3.1 Structure Identification 17
      2.3.2 Parameter Estimation 18
      2.3.3 Local and Global Learning 20

3 Identification of Local Linear State-Space Models 21
   3.1 Modelling in Observer Canonical Form 22
   3.2 Identification of Continuous-Time Models from Sampled Data 24
      3.2.1 Delta-Operator Model 25
      3.2.2 Continuous-Discrete Setup 25
   3.3 Local Learning Techniques 26
      3.3.1 Local Pole-Placement Method 26
      3.3.2 Local Recursive Prediction Error Method 30
3.4 Global Learning Techniques ........................................... 32
  3.4.1 RPE Method for Discrete-Time and Delta-Operator
  Models ................................................................. 34
  3.4.2 Continuous-Discrete Method .................................. 36
3.5 Local Versus Global Learning ......................................... 39
3.6 Chapter Summary ...................................................... 43

4 Controller Design Based on Input-Output Models ................. 44
  4.1 Model Reference Adaptive Control ............................... 46
  4.2 Nonlinear Adaptive Feedforward Cancellation ................. 50
  4.3 Nonlinear Model Predictive Control ............................ 55
  4.4 Chapter Summary and Discussion ................................ 60

5 Controller Design Based on State-Space Models ................. 62
  5.1 Local Linear-Quadratic Optimal Control ....................... 63
  5.2 Nonlinear Model Predictive Control ............................ 66
  5.3 Chapter Summary and Discussion ................................ 68

6 Simulation and Experimental Studies ............................... 70
  6.1 Simulation Studies ................................................ 71
    6.1.1 Identification of Local Linear State-Space Models ....... 71
    6.1.2 Model Reference Control .................................... 78
    6.1.3 Adaptive Feedforward Cancellation ......................... 81
    6.1.4 State-Feedback Control ..................................... 83
  6.2 Laboratory Experiments .......................................... 87
    6.2.1 Adaptive Control of a Hydraulic Positioning System ...... 87
    6.2.2 Identification and Control of a pH-Neutralisation Plant 91
  6.3 Chapter Summary and Discussion ................................ 99

7 Conclusions ............................................................. 102

Zusammenfassung ......................................................... 106

Bibliography ............................................................. 110
Nomenclature

Acronyms

AFC  Adaptive Feedforward Cancellation
CART Classification And Regression Trees
LLC  Local Linear Controller
LLF  Local Linear Feedforward Controller
LLM  Local Linear Model
LM   Local Model
LMN  Local Model Network
LOLIMOT Local Linear Model Tree
LQ   Linear Quadratic
MLP  Multi-Layer Perceptron
MRAC Model Reference Adaptive Control
NARX Nonlinear Auto Regressive with Exogeneous Inputs
NOCF Nonlinear Observer Canonical Form
ONALAL Online Adaptive k-tree Lattice Learning
RBF Radial Basis Function
RLS  Recursive Least Squares
RPE  Recursive Prediction Error

General Notation

\( a, b, \theta, \ldots \) Scalars
\( \mathbf{a}, \mathbf{b}, \mathbf{\theta}, \ldots \) Vectors
\[ \mathbf{A, B, \ldots} \quad \text{Matrices} \]
\[ \mathbf{A}(s), \mathbf{B}(z^{-1}), \ldots \quad \text{Polynomials} \]
\[ \mathbf{A}^{(n \times m)} \quad \text{A matrices of dimension } n \text{ by } m \]
\[ x_i \quad \text{The } i\text{th element of vector } \mathbf{x} \]
\[ \mathbf{I}_n \quad \text{Identity matrix of dimension } n \]
\[ 0 \quad \text{Zero vector or matrix} \]
\[ \approx \quad \text{Approximate value} \]
\[ \Re^{n \times N} \quad \text{Real space of dimension } n \times N \]

**Functions and Operators**

\[ \cdot \quad \text{Derivative with respect to time} \]
\[ \top \quad \text{Transpose operator} \]
\[ \sin \quad \text{Sine} \]
\[ \cos \quad \text{Cosine} \]
\[ \tanh \quad \text{Hyperbolic tangent} \]
\[ \partial \quad \text{Partial derivative} \]
\[ \sum \quad \text{Summation} \]
\[ \otimes \quad \text{Kronecker tensor product} \]
\[ \text{col} \quad \text{col operator, which stacks all the columns of matrix to give a column vector} \]
\[ \text{diag} \quad \text{diag operator, which converts a vector to a diagonal matrix or vice versa} \]
\[ p \quad \text{Placeholder for operators } q \text{ and } \frac{d}{dt} \]
\[ q \quad \text{Forward shift operator} \]
\[ \delta \quad \text{Delta operator} \]
\[ \gamma \quad z\text{-domain equivalent of } \delta \]

**Variables**

\[ \mathbf{a} \quad \text{Parameter vector of an I.I.M in observer canonical form} \]
\[ A \quad \text{Denominator polynomial of a system; also a port of the hydraulic actuator} \]
\( A \) System matrix; also parameter matrix of LLM-based observer
\( \tilde{A} \) Incremental extended system matrix
\( A_1, A_2 \) Piston impact areas of the hydraulic positioning system
\( A_k \) Denominator polynomial of the corrective filter
\( A_m \) Denominator polynomial of the reference model
\( b \) Parameter vector of an LLM in observer canonical form
\( B \) Numerator polynomial of a system; also a port of the hydraulic actuator
\( B \) Parameter matrix of LLM-based observer
\( \tilde{B} \) Input matrix of the incremental extended system
\( B_k \) Numerator polynomial of the corrective filter
\( B_m \) Numerator polynomial of the reference model
\( c \) Centre of an operating region
\( C \) Numerator polynomial of the disturbance model
\( d \) Transmission delay of a system
\( d_k \) Transmission delay of the corrective filter
\( d_m \) Transmission delay of the reference model
\( D \) Denominator polynomial of the disturbance model
\( e \) Estimation error
\( f_a \) Activation function
\( f_I \) Integrator flag
\( f_{\text{LM}} \) LLM approximation of a nonlinear function
\( f_{\text{LMO}} \) LLM approximation of the nonlinear observer function
\( F \) System matrix of the closed-loop observer
\( G \) Jacobian of output/state sequence with respect to control sequence
\( G_m \) Transfer function of the reference model
\( J \) Objective or cost function
\( k \) Observer feedback gain
\( K \) Parameter matrix of LLM-based observer gain; also the state-feedback gain matrix
\( l \) Number of measurable disturbances
NOMENCLATURE

\( l \)  
Parameter adaptation gain

\( L \)  
Number of local models in an L.MN

\( m \)  
Number of system outputs

\( M \)  
Mass in the hydraulic positioning system

\( n \)  
Number of system states

\( n_p \)  
Number of parameters

\( n_u \)  
Number of maximum lags of the input signal

\( n_y \)  
Number of maximum lags of the output signal

\( N \)  
Prediction horizon

\( p_c \)  
Parameter vector of a local controller

\( P \)  
Covariance matrix; also the solution of the Riccati equation

\( P_0, P_1, P_2 \)  
Oil pressures in the hydraulic positioning system

\( Q \)  
Weighting matrix of a predictive or LQ controller

\( r \)  
Number of system inputs

\( r \)  
Offset vector of a local linear model

\( R \)  
Controller polynomial

\( R \)  
Weighting matrix of a predictive or LQ controller; also offset matrix of L.I.M-based observer

\( S \)  
Controller polynomial

\( T \)  
Controller polynomial; also sampling period; also a port of the hydraulic actuator

\( T \)  
Similarity transformation matrix

\( u \)  
System input

\( \tilde{u} \)  
Incremental extended input vector

\( u_b \)  
Control signal generated by the feedback controller

\( u_f \)  
Control signal generated by the feedforward controller

\( U \)  
Input domain

\( w \)  
Command input or reference signal

\( W \)  
General weight/parameter matrix of a local linear model

\( \mathcal{W} \)  
Sensitivity matrix \( \frac{\delta \theta}{\delta \theta^T} \)

\( x \)  
State vector

\( \tilde{x} \)  
Incremental extended state vector
\[ y \] System output
\[ y_k \] Output of the corrective filter
\[ y_m \] Output of the reference model
\[ \mathcal{Y} \] Output domain
\[ z \] Disturbance signal; also z-domain variable
\[ \beta \] Overlapping factor of local models
\[ \psi_u, \psi_y, \psi_r \] States of the state-variable filters
\[ \psi \] \( \frac{dy}{d\theta} \): Gradient of predicted output with respect to the parameter vector \( \theta \)
\[ \psi_K \] \( \frac{dy}{d\theta_K} \): Gradient of predicted output with respect to the parameter vector \( \theta_K \)
\[ \psi_p \] \( \frac{dy}{d\theta_p} \): Gradient of predicted output with respect to the parameter vector \( \theta_p \)
\[ \lambda \] Forgetting factor
\[ \varphi \] Network input vector; also vector of independent variables of a function
\[ \varphi_c \] Input vector of a controller network
\[ \epsilon \] Noise vector
\[ \sigma \] Width of an operating region
\[ \theta \] Parameter vector
\[ \zeta \] Scheduling vector for an LMN
Einleitung


vorgestellte Fuzzy-Inferenz-System und das von Takagi und Sugeno (1985)
vorgeschlagene Fuzzy-Identifikationsverfahren sind vermutlich die ersten Beispiele
zur stückweisen Modellierung mit unscharfen Übergängen. Heute ist es schwie-
rig zu entscheiden, ob LMN-Netze zur Klasse der Fuzzy-Systeme oder zu den
künstlichen neuronalen Netzen zu rechnen sind. Zur Synthese und zum Trai-
ning der LMN-Netze sind aus der Literatur zahlreiche Formen von Netzen und
Algorithmen bekannt (Johansen und Murray-Smith, 1997). Das Training eines
LMN-Netzes kann in zwei Teilaufgaben aufgeteilt werden:

- Strukturidentifikation und

- Schätzung der lokalen Modelle.

Zu der Strukturidentifikation gehören die Bestimmung der Anzahl lokaler Mo-
delle und die Anpassung der geometrischen Formen und Gewichtungsfunktio-
nen der Arbeitsbereiche. Einige bekannte Trainingsverfahren halten die Anzahl
der Arbeitsbereiche in einem Netz fest, wobei die Parameter der Gewichtungs-
funktionen anhand eines Gradientenverfahrens optimiert werden (Poggio und
Verfahren um die Parameter zu optimieren. Andere Trainingsverfahren ver-
wenden eine adaptive Anzahl lokaler Modelle. Verfahren aus dieser Kategorie
beginnen mit einer einfachen Netzstruktur, z.B. mit einem einzigen lokalen
Modell und erhöhen die Netzkomplexität durch hierarchische Partitionierung
während des Trainingsvorgangs (Sanger, 1991; Nelles, 1997). Andere beginnen
mit einer komplexen Netzstruktur und vereinfachen diese während des Trai-
nings, indem entweder redundante lokale Modelle entfernt oder zusammenge-
fasst werden (Omphundro, 1991). Einige Verfahren (z.B. Murray-Smith und
Hunt, 1995) starten das Verfahren mit einem einfachen Netz und verwenden
den Approximationsfehler um neue lokale Modelle dem Netz hinzuzufügen.
Arbeitsbereiche auf oder verschmelzt diese anhand einer Kombination von lo-
kalem Fehlermaß und Heuristik, um damit eine optimale Struktur zu erhalten.

Eine Vielzahl der LMN-Netze verwendet lineare (oder affine) Modelle. Sol-
che Netze werden als Netze lokaler linearer Modelle (LLM-Netze) bezeichnet.


Identifikationsverfahren für zeitkontinuierliche lineare Systeme verwenden meistens lineare Filterbänke, die als „state-variable filters“ bezeichnet werden, um die Regressionssignale zu erzeugen, die für die Approximation der

den nachteilige Umschalteffekte beobachtet, wenn der Arbeitspunkt zwischen lokalen Modellen schnell wechselt.


Aufgrund der oben geführten Diskussion und unter Berücksichtigung der offenen Probleme auf dem Gebiet des Einsatzes von LLM-Netzen zur Identifikation und Regelung nichtlinearer dynamischer Systeme, werden die folgenden Ziele dieser Arbeit formuliert:


- Zur Untersuchung der praktischen Anwendung von LLM-Netzen sollen die entwickelten Algorithmen in Simulationsstudien mit einander verglichen und an realen Laboranlagen erprobt werden.


Abbildung 1: Zur Gliederung der Dissertation


Kapitel 7 schließt mit einer kurzen Zusammenfassung, einer Diskussion der wichtigsten Ergebnisse und einem Ausblick.
Chapter 1

Introduction

This dissertation addresses the application of local model networks (LMN) in the field of identification and control of nonlinear systems. Local model networks belong to a special class of artificial neural networks and can be used for the approximation of nonlinear mappings. They decompose the space of the input vector $\varphi$ of the nonlinear mapping $f(\varphi)$ into different operating regions in such a way that a relatively simple model for each of these regions can be estimated. These models, which are valid in limited operating regions, are known as local models. A validity function is associated with each operating region. This function can be considered as the weighting function of the local model. The value of the validity function is high if the input vector $\varphi$ lies “well inside” the operating region and decreases with distance between input vector and the centre of the region. This function is also known as blending function as its role is to blend all the local models to give a nonlinear approximation $\tilde{f}$ of the unknown mapping $f$. For a proper approximation it is necessary to adjust the geometry of the operating regions, the form of their blending functions and the estimates of local models. These tasks are performed by learning algorithms.

Research in the field of artificial neural networks can be traced back to the pioneering work by McCulloch and Pitts (1943). The well-known multi-layer perceptron (MLP) networks were proposed by Rosenblatt (1962) and Widrow
(1962). The basic idea of back-propagation algorithm for the training of MLP networks was first coined by Werbos (1974). Subsequently it was published by Rumelhart et al. (1986) and became the most popular learning algorithm in the field of artificial neural networks. A similar learning algorithm was independently derived by Parker (1985). Investigation of the approximation capabilities of multi-layer perceptrons was started in 1987. After the pace-making publication by Hecht-Nielsen (1987), some other research articles on the approximation properties of MLP networks and universal approximation theorem were published by Cybenko (1989), Funahashi (1989) and Hornik and Stinchcombe (1989). Barron (1992) investigated the trade-off between two conflicting requirements concerning the size of the hidden layer. Besides MLP networks, the other popular class of neural networks is known as radial-basis function (RBF) networks. These functions were initially introduced for the solution of multi-variable interpolations. The early work was surveyed by Powell (1985). The use of radial-basis functions for the construction of neural networks was first published by Broomhead and Lowe (1988). Other major contributions to the theory, design and applications of RBF networks are the papers published by Moody and Darken (1989) and Poggio and Girosi (1990). Moody and Darken (1989) proposed a computationally efficient hybrid learning method for RBF networks, which combines self-organised and supervised learning. They declared that their networks learn faster than back-propagation networks and are thus appropriate for real-time use. Poggio and Girosi (1990) used regularisation theory to construct RBF networks for the approximation of continuous mappings. The theory of regularisation networks was generalised to a formulation that included task-dependent clustering and dimensionality reduction. Girosi and Poggio (1990) compared back-propagation-based multi-layer networks with regularisation-based RBF networks and found that the latter are far better than the former in the sense of best approximation properties. Park and Sandberg (1991) proved the universal approximation abilities of the RBF networks. Another major difference between RBF and MLP networks is that individual RBF neurons, on contrary to MLP neurons, make only local contributions to the global approximation realised by the network. For this reason, the RBF networks can be considered as a special class of local
model networks.

The idea of approximation based on local models is not new. The investigations done by some researchers (like Kasavin, 1972; Rajbman et al., 1981; Billings and Voon, 1987) known as piecewise linear approximations or multi-model based identification techniques clearly fall into this category. These approaches use banks of linear models to estimate the behaviour of a system over a wide range of operation. These schemes switch between linear models after detecting a change in operating conditions (Pickhardt, 1995). This switching is sharp leading to undesired transients. Research interest in the soft transitions between local models was intensified after the publication of Zadeh’s (1965) paper on fuzzy sets. This was the time, when piecewise linear modelling took a turn from hard switching to soft transitions. Fuzzy inference system proposed by Mamdani (1974) and fuzzy identification approach published by Takagi and Sugeno (1985) are probably the first examples of piecewise models with soft transitions. Today, it is difficult to decide whether local model networks fall into the category of fuzzy-systems, artificial neural networks or both. For the construction and training of local model networks many approaches have been reported in the literature (Johansen and Murray-Smith, 1997). The task of training a local model network may be split into two subtasks:

- structure identification

- estimation of the local models.

Structure identification comprises the determination of number and geometric shapes of the operating regions and the parameters of the weighting or validity function. Some schemes keep the number of operating regions fixed while the parameters are adjusted by using gradient-based optimisation techniques (Poggio and Girosi, 1990) or maximum-likelihood learning (Jordan and Jacobs, 1994). Other techniques use an adaptive number of local models. Some techniques of this kind start with a simple structure (e.g. single local model) and perform hierarchical partitioning during the learning process and end up with a complex model (Sanger, 1991; Nelles, 1997). Others start with a large number of local models and merge them to a small number using a fine-to-
coarse policy (Omohundro, 1991). Some techniques (e.g. Murray-Smith and Hunt, 1995), starting with a simple network, use the approximation error in order to add new local models to the network, while preventing over-fitting. Ali (1998) and Junge (1999) combined a local error index with some heuristic to split or merge local regions in order to get a parsimonious structure.

As the local models are mostly linear (or affine), estimation techniques from the linear system theory, like recursive least squares (RLS), are applied for the estimation of their parameters. Parameter estimation of local linear models can be classified into two approaches. One approach, which is known as local learning approach, selects the currently active model from the network and estimates its parameters by minimising a local cost function. The other approach minimises a global cost function by estimating the parameters of all the local models simultaneously. Murray-Smith and Johansen (1997) gave a comparison of both approaches.

Research work in the field of identification and control of nonlinear dynamical systems using neural networks was started in late nineteen eighties. (Narendra and Parthasarathy, 1990; Matthews, 1990; Wabgaonkar and Stubberud, 1990; Fernandez et al., 1990; Narendra and Parthasarathy, 1991) seem to be among the early publications in this field. Research activities in this field touched their climax in mid nineties (Junge, 1999). Hunt et al. (1992) surveyed the pioneer approaches in this field. The earlier concepts used multi-layer perceptron (MLP) networks for identification tasks. Research works published by Girosi and Poggio (1990) and Park and Sandberg (1991) on the properties of RBF networks directed the attention of people working on system identification to this class of networks. Another advantage of RBF networks is their linear-in-parameter property of the output layer. MLP and RBF networks have proven themselves good means for nonlinear system identification. The major drawback of these black-box models is that the network parameters have no direct correspondence to the physical system parameters. If the parameters of a linearised model of the system at current point of operation are needed, then the gradient of this neural model has to be calculated. This calculation is, on one hand, computationally expensive and, on the other hand, error-prone due to the presence of small unwanted ripples in the approximations realised.
by RBF and MLP networks. These facts diverted the attention of the research community once again to local model networks (Johansen, 1994; Murray-Smith, 1994; Junge and Unbehauen, 1998a; Nelles, 1997). These new schemes approximate the nonlinear function smoothly by blending local affine models. This smoothness of approximation is necessary if a model-based controller is to be designed, or the model gradients are required for some optimisation purposes. Another major advantage of local model networks is the exploitation of linear design techniques while devising corresponding controller networks (Hunt and Johansen, 1997; Junge and Unbehauen, 1998b).

Major focus of neural identification has been on the estimation of discrete-time nonlinear input-output models (Narendra and Parthasarathy, 1990; Sjöberg, 1995). Such models are described by nonlinear difference equations. A little attention was given to the identification of nonlinear state-space models. Neural identification of continuous-time systems has also been less extensively explored as compared to discrete-time systems. This lack of interest is noticeable not only in the nonlinear case but a comparison of identification techniques for linear discrete-time systems with those of linear continuous-time systems reveals that the area of continuous-time identification is far under-represented. Out of a total 55724 publications in the field of system identification and parameter estimation, Bohn (2000) could find only six publications, which addressed the problem of nonlinear continuous-time state-space identification. The reasons for this under-representation may be one or more of the following: In the age of digital computers the data are available in sampled form, so it is convenient to train a discrete-time model. The other reason is that for the implementation of continuous-time models the system states have to be integrated in continuous-time. Execution of exact integration is not possible using a digital computer. Of course there are algorithms, which perform approximate integration by discretising the time scale but the computational cost is high. A third reason may be that controller implementation with digital computers is discrete. In order to design a discrete-time controller a discrete-time model is required.

Identification techniques for continuous-time linear systems often use a bank of linear filters, also known as state-variable filters, to generate regression
signals required for the approximation of the coefficients of a linear differential equation (Unbehauen and Rao, 1987; Unbehauen and Rao, 1990; Young, 1981). There are some other techniques derived from the original state-variable filters approach, which transform a linear differential equation into an algebraic equation using a low-pass filter operator (Johansson, 1993). Once this transformation is done, the coefficients of the resulting algebraic equation can be determined by using the filtered signals as regressors. Although this kind of transformation is not valid for nonlinear differential equations, there are some examples in the literature, which make use of this technique under some restrictions. Especially people working on local linear model networks employ techniques of this kind to determine the parameters of the local linear models (McLoone et al., 2001). These techniques neglect the overlap factors among different local models and concentrate on one local model at a time.

Most of the neural control schemes proposed in the early and mid nineties are based on MLP or RBF networks (Hunt et al., 1992; Narendra and Mukhopadhyay, 1997; Ng, 1997). Nonlinear training techniques are required to train such neural controllers in an inverse model, internal model or model reference adaptive control setup. The neural networks in such control schemes need an extensive pre-training by simulations before they can be used in real control loops. Local model networks tackle the problem in another way. These schemes utilise, in most cases, two local model networks. One network is responsible for the identification of the plant, while the other behaves as controller (Junge, 1999; Hunt and Johansen, 1997). In an adaptive scheme, the model network is trained at each sampling hit. On the basis of the parameters of currently active local model in the identification network, corresponding control law is calculated. Such schemes can be termed as master-slave schemes, as the model network behaves as a master and the controller network is dependent on the master for parameters and validity functions (Ali, 1998). Even with soft transitions between local models and controllers, these approaches face transient problems if the operating point is jumping quickly from one local model to the other.

Like identification, the attention of the researchers working on neural control was mainly focused on controller design based on input-output models.
Only few publications on application of neural networks, especially local model networks, in state-space control can be found (Bendtsen, 1999).

On the basis of above discussion and keeping the open problems in mind, the major aims and contributions of this dissertation are formulated as below:

- One of the major aims of this work is the development and investigation of various techniques for the identification of local linear state-space models for discrete as well as continuous-time nonlinear systems. This identification problem is formulated as an adaptive observation problem for nonlinear systems, where model parameters and system states have to be estimated on the basis of sampled input-output data of the system. In this work LLM networks are used to solve this problem. As a first step local learning algorithms for LLM networks are derived on the basis of state-variable-filters method known from the identification of continuous-time linear systems. Problems faced by local learning algorithms due to neglecting the overlapping and switching nonlinearities are investigated. As an alternative global learning techniques are proposed. Algorithms based on these techniques define a global cost function and utilise gradients of the whole blended model for the minimisation of this cost function.

- The second major aim is to develop and investigate various nonlinear control techniques using LLM networks. These networks are applied in this work to devise new nonlinear state-feedback and output-feedback control schemes in master-slave configuration. Merits and demerits of these control schemes are discussed. New methods for the application of LLM networks to predictive control of nonlinear systems are developed. Master-slave control schemes are compared with non-master-slave predictive control schemes.

- In order to investigate the practical application of LLM networks the developed algorithms are tested in several simulation studies and laboratory experiments on real plants.

This dissertation is organised as follows (Figure 1.1). Chapter 2 gives some introductory details of local model networks. Starting from the basic notion,
different forms of operating regions and validity functions are presented. The training approaches for local model networks concerning structure optimisation as well as parameter estimation are summarised. This chapter also deals with the application of local model networks to the identification of nonlinear input-output models. The approach discussed in this chapter is based on the nonlinear models of the discrete-time systems and is already known in the literature. In this work, this is included only for the sake of completion and reference. The major emphasis of this dissertation is on the following chapters.

Chapter 3 describes methods for the identification of nonlinear state-space models using local model networks. The presentation is kept so that majority of derivations and algorithms can directly be implemented for discrete- as well as for continuous-time systems. Local training techniques using state-variable filters known from the continuous-time linear system identification are compared with global training techniques, which use the gradients of the global cost function to adjust the network parameters.

![Diagram](image)

**Figure 1.1:** Structure of the dissertation

Chapter 4 deals with input-output model-based control schemes proposed for nonlinear systems. Three new control schemes are proposed in this chapter. The first scheme uses two local model networks to implement a model reference
adaptive control scheme for nonlinear systems. The second scheme realises adaptive feedforward compensation of measurable disturbances for a class of nonlinear systems. The third scheme is a predictive control scheme, which uses a local linear model network to predict the future trajectory of the nonlinear system. In addition to the prediction of the system trajectory, the LMN model delivers system parameters at each point of the predicted trajectory. This information is used to derive the gradients of the objective function in order to optimise the future control sequence.

In Chapter 5 two state-space control techniques are proposed. The first technique uses a master-slave configuration. Each element of the controller (slave) network is designed by using linear quadratic (LQ) optimisation based on the parameters of the corresponding local model in the master network. The second scheme is again a predictive control scheme based on the local state-space models.

Chapter 6 deals with the experimental results including simulation studies and experiments on laboratory plants. Simulation studies are used to discuss typical characteristics of the algorithms presented in this dissertation. Simulation examples are selected such that these characteristics could be elaborated. Laboratory experiments include the identification and control of two laboratory plants, a hydraulic positioning system and a pH neutralisation plant. These plants demonstrate highly nonlinear behaviours due to the presence of a dead zone nonlinearity in the actuator and a nonlinear titration curve respectively.

Chapter 7 concludes this dissertation with a short summary, discussion of important results and an outlook for the future research in this area.
Chapter 2

Local Model Networks

This chapter gives an overview of local model networks. In Section 2.1, after the introduction of the basic structure of the local model network, different approaches for the partitioning of input space, specification of validity functions and selection of local models are described briefly. Section 2.2 deals with the application of local linear model (LLM) networks to the identification of nonlinear dynamical systems in input-output model form. Section 2.3 summarises existing approaches for the identification of network structure and the estimation of local model parameters.

2.1 Basic Structure and Operating Regions

Local model networks are based on the divide-and-conquer strategy. The input space of the complex nonlinear mapping to be approximated is divided into different operating regions, also known as operating regimes. For each of these operating regions a relatively simple, e.g. linear, model is estimated. All these local models blended together give the nonlinear approximation of the unknown mapping. A local model approximation of a nonlinear vector function
\( y = f(\varphi) \) can be given as

\[
\hat{y} = \hat{f}_{LM}(\varphi) = \sum_{j=1}^{L} g_j(\varphi)v_j(\zeta),
\]

(2.1)

where \( g_j \) is a local model\(^1\), \( v_j \) is the validity function associated with the model \( g_j \). \( L \) is the total number of local models in the network. The validity function, sometimes also known as blending function, defines the geometry of the operating region of the local model. The so-called scheduling vector \( \zeta \) may be a selected part of, or identical to, the input vector \( \varphi \). The typical structure of a local model network is drawn in Figure 2.1.

The idea of local modelling approach has emerged from different disciplines, e.g. neural networks, fuzzy logic, statistics and artificial intelligence. Johansen and Murray-Smith (1997) give a summary of various approaches. From the point of view of validity functions, local modelling approaches can be divided into two classes. Some approaches realise sharp switching of local models using step-like validity functions. The other class avoids unwanted switching transients by using soft transitions from one local model to the other. Validity

\(^1\)Generally, the LMN approaches use local models with a single output. In this dissertation local models with multiple outputs are utilised due to the reasons, which will be significant in the next chapter.
functions in approaches of this class are continuous functions with a typical range of $[0, 1]$. Operating regions of neighbouring local models overlap each other in order to realise a smooth transition between the models.

There are different paradigms of dividing the space of the scheduling vector $\zeta$ into operating regimes. Some of these techniques divide it into hyper-elliptical and some in hyper-rectangular and others in entirely irregular geometrical shapes. Figure 2.2 shows examples of rectangular and elliptical partitioning for a scheduling vector of dimension two. Local model networks having hyper-elliptical shaped operating regions are similar to RBF networks if all the local models are constants. Hyper-rectangular shaped local regions have an advantage that a single local region can be split into two regions, which cover exactly the same area and are again rectangular in shape. Validity functions

\begin{figure}[h]
\begin{center}
\includegraphics[width=\textwidth]{figure2.2}
\end{center}
\caption{Rectangular and elliptical operating regions}
\end{figure}

for elliptical local models are normally Gaussian functions. The validity function of the $j$th local model for an $r$ dimensional scheduling vector $\zeta$ can be given as

$$v_j(\zeta) = \exp \left( -\frac{1}{2} \sum_{i=1}^{r} \left( \frac{\zeta_i - c_{ij}}{\sigma_{ij}} \right)^2 \right),$$

(2.2)

where $c_{ij}$ and $\sigma_{ij}$ are the centres and widths of the $j$th validity region in the $i$th input/scheduling dimension. A typical validity function for a rectangular
local model \( j \) is given by the following expression (Junge, 1999)

\[
v_j(k) = \begin{cases} 
1 & a_j(k) \leq 1 - \beta \\
 f_a(a_j(k), \beta), & 1 - \beta < a_j(k) < 1 + \beta \\
0 & a_j(k) \geq 1 + \beta, 
\end{cases} \quad (2.3)
\]

where \( 2\beta \) is the width of the region in which two neighbouring models overlap and \( f_a \) is any continuous function, which causes a smooth transition between 0 and 1. The quantity \( a_j(k) \) is defined as

\[
a_j(k) = \max_{1 \leq i \leq r} \frac{\left| \zeta_i(k) - c_{ij} \right|}{0.5\sigma_{ij}}. \quad (2.4)
\]

Local model networks are modular networks, where each local model is a separate module. The local models \( g_j \) can be static linear or nonlinear functions of \( \varphi \) or dynamical models represented by differential or difference equations (Nelles, 2001). Sbarbaro (1997) implemented a scheme with local Laguerre models. The most popular form of local model networks uses linear (strictly speaking affine) functions of the form

\[
g_j(\varphi) = \mathbf{W}_j^T \varphi + r_j, \quad (2.5)
\]

where matrix \( \mathbf{W}_j \) contains the parameters of the local model for the input vector \( \varphi \), also known as slope parameters, and \( r_j \) is an offset vector. A network composed of such models is called a local linear model (LLM) network, and has the output

\[
\hat{y} = \sum_{j=1}^{r} (\mathbf{W}_j^T \varphi + r_j)v_j(\zeta). \quad (2.6)
\]

Although all the local models represented by equation 2.5 are linear functions of \( \varphi \), the output \( \hat{y} \) of the blended network represented by equation 2.6 is a nonlinear function of \( \varphi \). These networks are capable of approximating nonlinear input-output mappings and will be used throughout this work for nonlinear identification and control purposes.
2.2 Identification of Local Linear Input-Output Models

This section describes how local linear model networks can be utilised for the identification of input-output models of nonlinear dynamical systems. The introduction in previous section considered the LLM networks as a tool for the approximation of nonlinear static functions. In order to identify a nonlinear dynamical system with LLM networks, a certain arrangement of external dynamics is necessary. This arrangement depends on the model, which is chosen for the description of the system.

2.2.1 Model Representation

The choice of a proper model is of crucial importance in system identification. Generally, the ARX (auto-regressive with exogenous inputs) model structure is used to describe a linear system corrupted by additive noise. Billings (1980) gave a comprehensive survey about traditional approaches to nonlinear dynamical system identification, including approaches based on general Volterra series and restricted block-oriented models like Hammerstein and bi-linear models. Descriptions and comparisons of various nonlinear model structures can be found in several publications (e.g. Leontaritis and Billings, 1985a; Leontaritis and Billings, 1985b; Billings and Chen, 1989; Chen and Billings, 1988). Different model structures in connection to neural networks are discussed by Narendra and Parthasarathy (1990). Almost all these models deal with discrete-time systems. The most popular and fairly general model used for the identification is the so-called NARX (nonlinear auto-regressive with exogenous inputs) model described by the nonlinear difference equation

\[ y(k) = f(y(k-1), \ldots, y(k-n_y), u(k-d-1), \ldots, u(k-d-n_u)) + \epsilon(k), \]  

(2.7)

where \( y \) is the vector of \( m \) outputs and \( u \) is the vector of \( r \) inputs of the system, \( \epsilon \) is a noise vector, \( f \) is a nonlinear function, \( k \) is the discrete-time instant, \( d \) is the transmission delay and \( n_u \) and \( n_y \) are the maximum lags of the input and output signals respectively. The deterministic nonlinear mapping \( f \)
of equation (2.7) can be approximated by using a proper nonlinear function approximator. A local linear model approximation of the above function is

\[
\hat{y}(k) = \hat{f}_{LM}(\varphi(k)) = \sum_{j=1}^{L} (W_j^T \varphi(k) + r_j)v_j(\zeta(k)),
\]

where

\[
\varphi(k) = \begin{pmatrix}
    y(k - 1) \\
    \vdots \\
    y(k - n_y) \\
    u(k - d - 1) \\
    \vdots \\
    u(k - d - n_u)
\end{pmatrix}.
\]

The LLM network is trained by using samples of vector \(\varphi(k)\) as inputs and samples of system output \(y\) as targets. This identification setup is also known as series-parallel setup (Narendra and Parthasarathy, 1990). For the simulation of trained network model the input vector \(\varphi(k)\) is redefined as

\[
\varphi(k) = \begin{pmatrix}
    \hat{y}(k - 1) \\
    \vdots \\
    \hat{y}(k - n_y) \\
    u(k - d - 1) \\
    \vdots \\
    u(k - d - n_u)
\end{pmatrix}.
\]

The LLM network with above input vector represents a nonlinear dynamical model of the system.

### 2.2.2 Dynamics of a Local Linear Model

This sub-section deals with the dynamics of an individual local linear model in an LLM network, which has been trained as a NARX model of a dynamical system. This consideration is of particular interest as it provides a basis for the design of local controllers. Suppose that the operating point remains near
to the centre of a local model $j$ for a “long time”. The value of the validity function $v_j$ remains approximately constant at its maximum value of one, while all other validity functions in the network are approximately zero. When the nonlinear model represented by the LLM network is linearised close to the centre of the region $j$, the resulting linear model is given by

$$\hat{y}(k) = w_j^T \varphi(k) + r_j.$$  \hspace{1cm} (2.11)

The $z$-transformation of this expression after substitution of the value of $\varphi(k)$ from equation 2.10 is given as

$$\hat{Y}(z) = \frac{B_j(z^{-1})}{A_j(z^{-1})} z^{-d} U(z) + \frac{r_j}{A_j(z^{-1})},$$ \hspace{1cm} (2.12)

where

$$A_j(z^{-1}) = 1 - \sum_{i=1}^{n_b} w_{ji} z^{-i}$$ \hspace{1cm} (2.13)

and

$$B_j(z^{-1}) = \sum_{i=1}^{n_u} w_{j,i+n_y} z^{-i}.$$ \hspace{1cm} (2.14)

Obviously this approximation is only valid in a small vicinity of the centre of the operating region. As long as data remain in this small neighbourhood, the nonlinear effects of validity functions and interactions between local models can be neglected.

### 2.3 Learning Techniques

A learning algorithm for a local model network has to perform two tasks: identification of the network structure and estimation of local model parameters. Structure identification comprises of the determination of the number and geometrical shapes of the operating regions and the parameters of the validity functions including centres, widths and overlap factors. A short description of various learning techniques for the training of local model networks is given in the following sub-sections.
2.3.1 Structure Identification

The techniques for the optimisation of positions and dimensions of the local regions reported in the literature are summarised in the following.

**Fixed selection:** In this approach the centres are selected randomly from the input data or distributed uniformly on a regular lattice. The widths of the operating regions are calculated according to some thumb rules based on a priori information. If the complexity of the problem is not known then a large number of local models is needed. This problem becomes severe if the dimension of the scheduling vector $\zeta$ is high.

**Self-organising and clustering:** In this approach the centres of the operating regimes are trained in an unsupervised learning fashion. A fixed number of local models organise themselves to cover the space of the scheduling vector (e.g. Moody and Darken, 1989; Sbarbaro, 1992). The disadvantage is that the local models are clustered according to the density of the data, not according to the complexity of the problem.

**Parametric optimisation:** Such approaches are also based on a fixed number of local models. But the parameters of the validity functions are optimised using gradient-based techniques. Examples can be found in (Poggio and Girosi, 1990; Jordan and Jacobs, 1994).

**Coarse-to-fine partitioning:** Such techniques start with a simple structure, e.g. a global linear model, and divide the input space into ever smaller areas. Examples are classification and regression trees (CART) (Breiman et al., 1984; Quinlan, 1993), basis-function trees (Sanger, 1991) and local linear model tree (LOLIMOT) (Nelles, 1997).

**Fine-to-coarse learning:** These techniques start with a large number of local models. During training, local models are merged together to get a simpler structure (Omohundro, 1991). The disadvantage of such approaches is that in the case of a high-dimensional problem a very large number of local models have to be initialised. A very large amount of data is required to train all the local models sufficiently before the decision about model merging is made.
Splitting and merging: These schemes try to adjust the network complexity according to the complexity of the problem. The algorithm tries to partition the input space more frequently, where the unknown mapping seems to be complex. The insignificant partitions are removed in order to limit the complexity of the network. The online adaptive k-tree lattice learning (ONALAL) algorithm presented in (e.g. Junge and Unbehauen, 1998a; Ali, 1998; Junge, 1999) combines a local error index with some heuristic to split or merge the local regions in order to get an optimal structure. This learning algorithm can be started with any initial form of the lattice. It automatically splits a local model into two, if the behaviour of the model is not satisfactory according to a given criterion. On the other hand, if two neighbouring models have nearly identical parameters, then in order to achieve a parsimonious structure these local models are merged together. As the input-output model-based adaptive control schemes presented in Chapter 4 utilise this algorithm for the online identification of the plant, a simplified flow chart of the algorithm is drawn in Figure 2.3.

All these structure identification techniques are computationally expensive. A lot of training is required before a decision about model reduction or refinement can be made. This problem becomes more crucial if the input dimension of the network is large. Therefore, it is always advantageous (or sometimes necessary) to consider the maximum possible a priori information about the system during determining the initial structure of the network.

### 2.3.2 Parameter Estimation

Estimation of local model parameters depends upon the choice of the local models. In the case of local linear models, this estimation is a linear optimisation problem if the validity functions are assumed to be known and fixed. Approaches for the estimation of local model parameters can be divided into two classes: local and global learning techniques. In the following sub-section these techniques are introduced and compared briefly.
Figure 2.3: A simplified flow chart of the ONALAL learning algorithm
2.3.3 Local and Global Learning

Local learning techniques define a local error criterion for each local linear model. Based on the minimisation of this criterion the parameters of each local model are estimated independently. In this approach the effect of other local models on the approximation due to overlaps is neglected. Generally, linear estimation techniques, e.g. method of recursive least squares (RLS), are used in this optimisation problem. Local learning techniques applied by Junge and Unbehauen (1998a) and Nelles (1997) use validity functions for the data weighting in a weighted recursive least squares estimation. Parameter convergence and the quality of approximation deteriorates with increasing overlap factors. Global parameter estimation techniques define a global error index and estimate the parameters of the whole blended model simultaneously. These schemes are capable of achieving better approximation quality as the effect of overlaps are not neglected in the estimation process. Major drawbacks of these techniques are:

- The number of parameters to be updated on each sample hit is large.

- Selection of too large overlap factors or structure mismatch may lead to an ambiguous estimation problem.

The latter means that several parameter settings of local models are possible, which achieve more or less the same approximation quality. Murray-Smith and Johansen (1997) showed that it is possible in such cases that one can end up with a parameter set of local models, which shows a good global behaviour but the parameters of the local models are quite different from those of the locally linearised problem. On one hand, selection of large overlap factors makes the transition behaviour of the model smooth, but on the other hand causes difficulties in the estimation process. A trade-off between the smoothness and the estimation should be made in the choice of overlap factors.
Chapter 3

Identification of Local Linear State-Space Models

A general state-space representation of a nonlinear system can be given as

\[ \begin{align*}
px(t) & = f(x(t), u(t)), \\
y(t) & = h(x(t), u(t)),
\end{align*} \]

(3.1)

where \( u \) is the vector of inputs, \( x \) the vector of state variables and \( y \) is the vector of system outputs. \( f \) and \( h \) are nonlinear vector functions. The operator \( p \) has different interpretations. In case of continuous-time systems, \( p \) stands for the differential operator \( \frac{d}{dt} \) and for discrete-time\(^1 \) systems it is a placeholder for the forward shift operator \( q \). If all the quantities \( u, x \) and \( y \) are measurable, then the nonlinear functions \( f \) and \( h \) can be approximated by any universal function approximator. Unfortunately the vector \( x \) is generally not (completely) accessible. In such cases, in addition to the approximation of nonlinear mappings \( f \) and \( h \), the state vector \( x \) has to be estimated. This problem becomes more complex than the identification problem discussed in the previous chapter. There are some approaches (e.g. Kreisslmeier, 1977; Ljung and

\(^1\)The variable \( t \) is used here to represent continuous and discrete time in order to keep the algorithms applicable to both time domains. But in expressions, where time variable has only discrete interpretation, e.g. recursive update algorithms, \( t \) is replaced by index \( k \) defined as \( t = kT \), \( T \) being the sampling period.
Söderström, 1983; Nazaruddin, 1994) reported in the literature, which address this problem for linear systems. But only a few publications could be found for nonlinear systems (Bohn, 2000). In this chapter the application of local model networks to solve this problem is addressed. The next section deals with the modelling of nonlinear systems in observer canonical form. Section 3.2 answers the question, how the parameters of a continuous-time model can be estimated from sampled data. Sections 3.3 and 3.4 describe different techniques for the estimation of local linear model parameters. Two approaches of parameter estimation methods are considered here. The first approach, which falls into the category of local learning techniques, makes the use of well-known state-variable filter technique in order to generate the regressors for the estimation of the parameters of active linear model. The second approach, which may be considered as a global learning technique, takes the whole blended model into account and adjusts the parameters of this model in order to minimise a global cost function. In Section 3.5 a comparison of these estimation techniques is presented. The chapter is summarised in Section 3.6.

3.1 Modelling in Observer Canonical Form

The nonlinear observer canonical form (NOCF) proposed by Keller (1986) is given as

\[ \begin{align*}
px(t) &= E \mathbf{x}(t) + f(x_1(t)) + g(x_1(t))u(t), \\
y(t) &= c^T \mathbf{x}(t),
\end{align*} \tag{3.2} \]

where

\[ E = \begin{pmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 1 \\
0 & 0 & \cdots & 0 & 0 
\end{pmatrix}, \tag{3.3} \]

and

\[ c^T = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{3.4} \]

This canonical form derives its name from the property that a nonlinear observer can be designed in canonical states as in the linear case by means of
pole-placement (Jelali and Schwarz, 1995). This canonical form exists only for a class of nonlinear systems. Keller (1986) investigated the conditions, which a nonlinear system must satisfy in order to be transferable into this form. A relatively more general form of the above representation is

\[
\begin{align*}
px(t) & = E\dot{x}(t) + f(x_1(t), u(t)), \\
y(t) & = c^T x(t).
\end{align*}
\tag{3.5}
\]

Another major advantage of this form is that the nonlinearity is dependent only on two variables which are physically accessible. This fact on one hand eases the estimation of the nonlinear vector function \( f \), and on the other hand, reduces the curse of dimensionality due to a small number of independent variables of \( f \). The latter feature is of significant importance when neural networks are used to approximate the nonlinearity. A local linear model approximation of the nonlinear vector function \( f(x_1, u) \in \mathbb{R}^n \) involved in equation 3.5 is given as

\[
f_{LM}(x_1, u) = \sum_{j=1}^{I} v_j(x_1, u)(a_j x_1 + b_j u + r_j),
\tag{3.6}
\]

where \( a_j x_1 + b_j u + r_j \) represents a local affine model and \( v_j(x_1, u) \) is the validity function associated with this local model. For the estimation of the system state \( x \), an observer with state \( \hat{x} \) should be designed. If for each local model \( j \) a feedback vector \( k_j \) is considered, then the whole setup of an observer can be expressed by

\[
\begin{align*}
px(t) & = E\dot{\hat{x}}(t) + \sum_{j=1}^{I} v_j(\hat{x}_1, u)(a_j \hat{x}_1 + b_j u + r_j + k_j(y - \hat{x}_1)), \\
& = E\dot{\hat{x}}(t) + f_{LMO}(\hat{x}_1, y, u).
\end{align*}
\tag{3.7}
\]

The function \( f_{LMO} \) can be realised by a local model network. Once the observer network is constructed, its parameters can be estimated in order to minimise the estimation error \( e(t) = y(t) - \hat{x}_1(t) \). The whole adaptive observation setup is shown in Figure 3.1. The objective to determine the proper structure of the model network and the estimation of its parameters and states. In this work, it is supposed that a proper network structure is known from \( a \) priori information or can be determined using the techniques discussed in Chapter 2.
The emphasis of this chapter is on the estimation of parameters $a_j, b_j, r_j$ and $k_j$ for all $L$ local models.

### 3.2 Identification of Continuous-Time Models from Sampled Data

Online identification of discrete-time system parameters is often done using recursive algorithms. Computational and update routines are called on a digital computer at each sampling hit. Parameters and signal covariances are updated recursively. Typical examples of continuous-time parameter estimation methods for linear systems are state-variable filter methods (Unbehauen and Rao, 1987) and continuous-time extended Kalman filters (Gelb, 1974). An exact realisation of these online identification methods requires ideal analog computers. If these methods are to be implemented on a digital computer then the input and output signals of the system must be discretised and discrete-time equivalents of these methods must be derived. There are two possible solutions to this problem: delta-operator model and continuous-discrete identification. In the following subsections, these techniques are described briefly.
3.2.1 Delta-Operator Model

This technique is based on the so-called delta (δ) operator, which is a discrete-time approximation of the continuous-time differential operator d/dt and is defined as (Ninnes and Goodwin, 1991; Young et al., 1991)

\[ \delta x(k) = \frac{x(k+1) - x(k)}{T}, \]  

(3.8)

where \( T \) is the sampling time. It is clear that this approximation is valid for short sampling times only, as

\[ \lim_{T \to 0} \delta x(k) = \lim_{T \to 0} \frac{x(k+1) - x(k)}{T} = \frac{dx(t)}{dt} \bigg|_{t=kT} \]  

(3.9)

The z-domain equivalent of this operator is, sometimes, denoted by \( \gamma \) and is given by

\[ \gamma = \frac{z - 1}{T} \]  

(3.10)

Inverse of this operator \( \gamma^{-1} = \frac{T}{z-1} \) is known as Euler’s forward method for discrete-time integration. The major advantage of this approximation is that only discrete-time implementation of the estimation algorithm is needed. The accuracy of parameter estimation depends on the sampling time. If the sampling time is sufficiently short the parameters and states determined by this setup are good estimates of the real continuous-time counterparts.

3.2.2 Continuous-Discrete Setup

This approach is a hybrid approach, in which state estimates and covariances of the estimation algorithm are propagated by using continuous-time models, whereas the parameter update, state and covariance corrections are applied periodically at each sampling hit. Approaches of this category are computationally expensive but achieve better approximation accuracy than delta-operator-based approaches. Examples of this type of algorithms include different filtering algorithms reported by Gelb (1974) and Bohn (2000).
3.3 Local Learning Techniques

The philosophy of a local learning technique is as follows: On the basis of the current input pattern a winner model is determined. Winner is that local model, whose validity function assumes a value greater than that of other models. The parameters of the winner model are updated in order to minimise a local cost function. McLoone et al. (2001) suggested a scheme to estimate the parameters of velocity-based continuous-time local models. They transform the differential equation of each local model to an algebraic equation using low-pass filters. This idea of algebraic transformation suggested by Johansson (1993) is very similar to the approach of state-variable filters (Unbehauen and Rao, 1990; Unbehauen and Rao, 1987; Young, 1981). The question that remains open in such schemes is how to select the linear filters for a specific problem. Kreisselmeier (1977) used the idea of an adaptive observer in order to identify a continuous-time system. This way the linear filters are characterised by the closed-loop observer poles. But the question arises, where to place observer poles. Some kind of optimal filtering may be a possible answer to this question. Section 3.3.1 describes how the local adaptive pole-placement method can be used for the identification of LLM observers. In Section 3.3.2 a local parameter estimation approach based on the recursive prediction error (RPE) method is described. Both of these schemes are local learning techniques, which can be applied to estimate the parameters of the currently active unit in the observer network shown in Figure 3.1.

3.3.1 Local Pole-Placement Method

This method is based on the adaptive observer for linear continuous-time systems suggested by Kreisselmeier (1977). The observer equation for a linear system is given as

\[ p\hat{x}(t) = F\hat{x}(t) + ky(t) + bu(t) \]  

(3.11)
with
\[
F = \begin{pmatrix}
-f_1 & 1 & 0 & \cdots & 0 \\
-f_2 & 0 & 1 & \vdots \\
\vdots & \vdots & \ddots & \ddots & 0 \\
\vdots & \vdots & & 1 \\
-f_n & 0 & \cdots & 0 & 0
\end{pmatrix}.
\] (3.12)

The eigenvalues of matrix \( F \) are the poles of the closed-loop observer. Vector \( k \) is known as observer gain and can be calculated by the relation \( k_i = f_i - a_i \).

The variables \( a_i \) are certain elements of the system matrix
\[
A = \begin{pmatrix}
-a_1 & 1 & 0 & \cdots & 0 \\
-a_2 & 0 & 1 & \vdots \\
\vdots & \vdots & \ddots & \ddots & 0 \\
\vdots & \vdots & & 1 \\
-a_n & 0 & \cdots & 0 & 0
\end{pmatrix}.
\] (3.13)

In order to find an expression for the estimation of the parameters \( k \) and \( b \), define two auxiliary vectors \( \psi_y \) and \( \psi_u \) as
\[
P \psi_y(t) = F^T \psi_y(t) + e_1 y(t), \quad \psi_y(0) = 0; \quad \psi_u(t) = F^T \psi_u(t) + e_1 u(t), \quad \psi_u(0) = 0.
\] (3.14) (3.15)

With a parameter vector \( \theta = (k^T \ b^T)^T \), and an initial value \( \hat{x}(0) = 0 \), the behaviour of the observed state vector can be given as
\[
\hat{x}(t) = \begin{pmatrix} T_1 \psi_y(t) & \cdots & T_n \psi_y(t) & T_1 \psi_u(t) & \cdots & T_n \psi_u(t) \end{pmatrix} \theta,
\] (3.16)

where the matrices \( T_1 \) to \( T_n \) are transformation matrices. Due to the particular form of matrix \( F \), the transformation matrices \( T_1 \) to \( T_n \) can be determined from the following relationship
\[
(pI - F)^{-1} e_i = T_i (pI - F^T)^{-1} e_1, \quad (i = 1, 2, \cdots, n)
\] (3.17)
where $e_i$ is the $i$th column of the unity matrix $I$. Further, it can be shown that
\[
\begin{pmatrix}
\psi_y^T(t) & \psi_u^T(t)
\end{pmatrix} = 
\begin{pmatrix}
c^T 
\end{pmatrix}
\begin{pmatrix}
T_1 \psi_y(t) & \cdots & T_n \psi_y(t) & T_1 \psi_u(t) & \cdots & T_n \psi_u(t)
\end{pmatrix}.
\]
(3.18)
Considering this relationship, the observer output can be given as
\[
\hat{y}(t) = (\psi_y^T(t) \quad \psi_u^T(t)) \theta.
\]
(3.19)
The estimation of the parameter vector $\theta$ is a linear regression problem that can be solved by applying linear optimisation techniques, e.g. RLS.

**Extension for affine systems**

The local models involved in equation 3.6 are not linear but affine. The above scheme must be extended to handle affine systems described by
\[
p x(t) = E x(t) + ax_1(t) + bu(t) + r.
\]
(3.20)
In order to estimate the elements of the offset vector $r$, another state-variable vector
\[
p \psi_r(t) = F^T \psi_r(t) + e_1,
\]
(3.21)
is introduced. The parameter vector is extended to give $\theta = (k^T \quad b^T \quad r^T)^T$ and the following regression is used for parameter estimation
\[
\hat{y}(t) = \psi^T(t) \theta,
\]
(3.22)
where $\psi(t) = (\psi_y^T(t) \quad \psi_u^T(t) \quad \psi_r^T(t))^T$. This scheme faces a problem. For any matrix $F$ with stable eigenvalues in a continuous-time observer canonical form the steady-state value of $\psi_r(t)$ will converge to
\[
\psi_r(\infty) = \begin{pmatrix}
0 & \cdots & 0 & 1/f_n
\end{pmatrix}^T.
\]
(3.23)
Soon after the identification is switched on, the algorithm lacks sufficient excitation for the identification of offset terms $r_1, r_2, \cdots, r_{n-1}$. This way the
estimation of only the last offset term $r_n$ is possible. The parameter values $r_1, r_2, \cdots, r_{n-1}$ remain unidentifiable. For an affine system this is not a problem because the parameters $r_1, r_2, \cdots, r_{n-1}$ in no way effect the system output as long as they remain constant. The effect of these parameters on the approximation error is zero. If the parameter estimator is considered to be an extended observer, where the parameters of the model are the additional states, then it can be shown that the states $r_1, r_2, \cdots, r_{n-1}$ are unobservable. It can be concluded that in case of an affine system one does not have to estimate these terms as long as they remain constant.

Table 3.1 gives the complete algorithm for the estimation of parameters of delta-operator LLM observer using recursive least squares (RLS) method on the basis of local pole-placement.

**Table 3.1: A local recursive least squares algorithm for the parameter estimation of delta-operator LLM observer (local pole-placement method)**

| Filter input and output signals | $p \psi_y(k) = F^{\top} \psi_y(k) + e_1 y(k)$  
|                               | $p \psi_u(k) = F^{\top} \psi_u(k) + e_1 u(k)$  
|                               | $\psi(k) = \begin{pmatrix} \psi_y^{\top}(k) & \psi_u^{\top}(k) & 1/f_n \end{pmatrix}^{-1}$ |

Determine active model $j$ for data $y(k)$ and $u(k)$. Next steps are carried out only for model $j$.

| Parameter vector | $\hat{\theta}_j = \begin{pmatrix} \hat{k}_j^{\top} & \hat{b}_j^{\top} & \hat{r}_j \end{pmatrix}^T$ |

| Covariance update | $P_j(k) = P_j(k - 1) - \frac{P_j(k-1)\psi(k)\psi^\top(k)P_j(k-1)}{1 + \psi^\top(k)P_j(k-1)\psi(k)}$ |

| Parameter adaptation gain | $l(k) = P_j(k)\psi(k)$ |

| Estimation error | $e(k) = y(k) - \psi^\top(k)\hat{\theta}_j(k - 1)$ |

| Parameter update | $\hat{\theta}_j(k) = \hat{\theta}_j(k - 1) + l(k)e(k)$ |

| State estimate | $p \hat{\mathbf{x}}(k) = F \hat{\mathbf{x}}(k)$  
|                  | $\quad + \sum_{i=1}^L (\hat{k}_i y(k) + \hat{b}_i u(k) + \hat{r}_i e_n) v_i(y, u)$ |
3.3.2 Local Recursive Prediction Error Method

In this section a recursive prediction error (RPE) method for the estimation of parameters and states of local affine model (equation 3.20) is derived. The observer equation for this system with $Ex(t) + ax_1(t)$ as $Ax(t)$ has the form

$$p\hat{x}(t) = (A - kc^T)\hat{x}(t) + bu(t) + ky(t) + r.$$  \hspace{1cm} (3.24)

Defining the observation error $e(t) = y(t) - c^T\hat{x}(t)$, a quadratic cost function to be minimised can be given as

$$J = \frac{1}{2}e^2(t).$$  \hspace{1cm} (3.25)

The objective is to adjust the parameters $\theta = \begin{pmatrix} a^T & k^T & b^T & r^T \end{pmatrix}^T$ in order to minimise the cost function. The Gauss-Newton method to solve this problem can be formulated as (Ljung and Söderström, 1983; Bohn, 2000)

$$S(k) = \lambda(k) + \psi^T(k)P(k-1)\psi(k)$$  \hspace{1cm} (3.26)

$$l(k) = P(k-1)\psi(k)/S(k)$$  \hspace{1cm} (3.27)

$$\dot{\theta}(k) = \dot{\theta}(k-1) + l(k)e(k)$$  \hspace{1cm} (3.28)

$$P(k) = (P(k-1) - l(k)S(k)l^T(k))/\lambda(k),$$  \hspace{1cm} (3.29)

where $P$ is the covariance matrix, $\lambda$ is a forgetting factor, $l$ is known as parameter adaptation gain, $S$ is an auxiliary variable and $\psi$ is defined as

$$\psi(k) = -\frac{de(t)}{d\theta} \bigg|_{t=kT} = \left( c^T \frac{d\hat{x}(t)}{d\theta^T} \right)^T \bigg|_{t=kT}. $$  \hspace{1cm} (3.30)

The expression $\frac{d\hat{x}(t)}{d\theta^T}$ is the Jacobian of vector $\hat{x}(t)$ and is also known as parameter sensitivity matrix. In this work this expression will often be denoted by $W(t)$. In order to minimise the cost function given by equation 3.25 an expression for the sensitivity matrix must be found. As the calculation of an exact relationship is usually not possible, an approximation can be achieved using the following procedure. A parameter sensitivity model can be derived by taking the derivative of equation 3.24 with respect to $\theta.$

$$\frac{d(p\hat{x}(t))}{d\theta^T} = \frac{d((A - kc^T)\hat{x}(t))}{d\theta^T} + \frac{db}{d\theta^T}u(t) + \frac{dk}{d\theta^T}y(t) + \frac{dr}{d\theta^T}$$  \hspace{1cm} (3.31)
Chapter 3. Identification of Local Linear State-Space Models

The operator $d/d\theta^T$ is commutable with operator $p$ only if the parameter vector $\theta$ is constant (Eykhoff, 1974). In that case equation 3.31 would give an exact propagation model for the sensitivity matrix. But in the case of online identification, the parameters of the model are time-varying. If it is supposed that the parameter variations are much slower than the signal variations, then an approximation of $p\frac{d\hat{x}(t)}{d\theta^T}$ can be given as

$$p\frac{d\hat{x}(t)}{d\theta^T} = \frac{d((A - kc^T)\hat{x}(t))}{d\theta^T} + \frac{db}{d\theta^T} u(t) + \frac{dk}{d\theta^T} y(t) + \frac{dr}{d\theta^T}. \quad (3.32)$$

By applying the matrix product rule

$$\frac{d(AB)}{dP(r \times s)} = \frac{dA}{dP}(I_s \otimes B) + (I_r \otimes A)\frac{dB}{dP} \quad (3.33)$$

equation 3.32 can be simplified as

$$p\frac{d\hat{x}(t)}{d\theta^T} = (A - kc^T)\frac{d\hat{x}(t)}{d\theta^T} + \frac{d(A - kc^T)}{d\theta^T}(I_{n_p} \otimes \hat{x}(t)) + \frac{db}{d\theta^T} u(t) + \frac{dk}{d\theta^T} y(t) + \frac{dr}{d\theta^T}, \quad (3.34)$$

where $n_p$ is the length of the parameter vector $\theta$. Denoting $\frac{d\hat{x}(t)}{d\theta^T}$ by $W(t)$ the above expression can be written as

$$pW(t) = (A - kc^T)W(t) + M(t), \quad (3.35)$$

where

$$M(t) = \frac{d(A - kc^T)}{d\theta^T}(I_{n_p} \otimes \hat{x}(t)) + \frac{db}{d\theta^T} u(t) + \frac{dk}{d\theta^T} y(t) + \frac{dr}{d\theta^T}$$

$$= \begin{pmatrix} I_{n_1}(t) & I_{n_2}(t) & I_{n_3}(t) & I_n \end{pmatrix}. \quad (3.36)$$

Estimation of offset vector $r$

The problem of estimation of all the elements of the offset vector $r$ in equation 3.24 is now considered for this scheme. If the sensitivity model of the vector $r$ is defined as $W_r = \frac{d\hat{x}}{d\alpha}$, which consists of the last $n$ columns of the matrix $W$, the following propagation equation for this sensitivity matrix can be extracted from equation 3.35

$$pW_r(t) = (A - kc^T)W_r(t) + In. \quad (3.37)$$
If the matrix $F = A - kc^T$ is in continuous-time observer canonical form and has fixed or time-varying but stable eigenvalues, then from any initial value $\mathcal{W}_r(t)$ the sensitivity matrix will exponentially converge to

$$\mathcal{W}_r(\infty) = F^{-1} = \begin{pmatrix} 0^T & f^* \\ I_{n-1} & \end{pmatrix}.$$  (3.38)

It is clear from equation 3.30 that the information contained only in the first row of matrix $\mathcal{W}(t)$ is used for the parameter estimation. All the elements of the first row of $\mathcal{W}_r(\infty)$ are equal to zero except the last element. This means that only the parameter $r_n$ can be estimated by this method. Other elements of vector $r$ have no effect on the observation error in $x_1$. The first $n-1$ elements of vector $r$ are not identifiable in this scheme too. A complete local recursive prediction method for the identification of discrete-time or delta-operator local state-space models is given in Table 3.2.

**A special case**

The eigenvalues of the matrix $A - kc^T$ represent the closed-loop poles of the observer. If instead of a time-varying observer matrix $A - kc^T$ some fixed observer matrix $F$ is selected (independent of model parameters: pole placement) then the number of parameters to be estimated will decrease by $n$. Either the parameter vector $a$ or $k$ has to be estimated and the other can be calculated from the relationship $F = A - kc^T$. This scheme then becomes identical to the method described in the previous sub-section. The propagation of the sensitivity matrix $\mathcal{W}(t)$ is not dependent on the parameters to be estimated. This leads to a linear regression problem, which can be solved by using the recursive least squares (RLS) method (Unbehauen, 1995).

### 3.4 Global Learning Techniques

In this section the identification of the whole blended network is considered. The derivations given in the following sub-sections deal systems, which can be described by the nonlinear observer canonical form expressed by the equations
**Table 3.2:** Local recursive prediction error method for LLM networks

<table>
<thead>
<tr>
<th>Description</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimation error</td>
<td>$e(k) = y(k) - c^T \hat{x}(k)$</td>
</tr>
<tr>
<td>Error gradient</td>
<td>$\psi(k) = \begin{pmatrix} I_{3n} &amp; 0 \ 0^T &amp; e_n^T \end{pmatrix} \mathcal{W}^T(k) c$</td>
</tr>
<tr>
<td>Determine active model $j$ for data $\hat{x}_1(k)$ and $u(k)$</td>
<td>$\hat{\theta}_j = \begin{pmatrix} \hat{a}_j^T \ \hat{k}_j^T \ \hat{b}_j^T \ \hat{r}_j \end{pmatrix}$</td>
</tr>
<tr>
<td>Parameter vector</td>
<td>$\hat{\theta}_j = \begin{pmatrix} \hat{a}_j^T \ \hat{k}_j^T \ \hat{b}_j^T \ \hat{r}_j \end{pmatrix}$</td>
</tr>
<tr>
<td>Parameter adaptation gain</td>
<td>$\lambda(k) = \psi^T(k) P_j(k - 1) \psi(k)$</td>
</tr>
<tr>
<td>Parameter update</td>
<td>$\hat{\theta}_j(k) = \hat{\theta}_j(k - 1) + l(k) e(k)$</td>
</tr>
<tr>
<td>Covariance update</td>
<td>$P_j(k) = (P_j(k - 1) - l(k) S(k) l^T(k)) / \lambda(k)$</td>
</tr>
<tr>
<td>Sensitivity propagation</td>
<td>$p\mathcal{W}(k) = F(k) \mathcal{W}(k) + M(k)$ with $M(k) = \begin{pmatrix} \hat{x}_1(k) &amp; u(k) &amp; e(k) &amp; 1 \end{pmatrix} \otimes I_n$ and $F(k) = E + (\hat{a}_j(k) - \hat{k}_j(k)) c^T$</td>
</tr>
<tr>
<td>State estimate</td>
<td>$p\hat{x}(k) = E\hat{x}(k) + \sum_{i=1}^{L} (\hat{a}_i(k) \hat{x}_1(k) + \hat{k}_i(k) e(k) + \hat{b}_i(k) u(k) + \hat{r}_j(k) e_n) v_i(\hat{x}_1, u)$</td>
</tr>
</tbody>
</table>

3.5 and 3.6. Defining $A = (a_1 \ a_2 \ \cdots \ a_L)$, $B = (b_1 \ b_2 \ \cdots \ b_L)$, $R = (r_1 \ r_2 \ \cdots \ r_L)$ and $v = (v_1 \ v_2 \ \cdots \ v_L)^T$, the global model described by these equations can be rewritten as

$$p x(t) = E x(t) + (A x_1(t) + B u(t) + R)v(x_1(t), u(t)). \quad (3.39)$$

An observer for the above system can be described by

$$p \hat{x}(t) = E \hat{x}(t) + (A \hat{x}_1(t) + B u(t) + K e(t) + R)v(\hat{x}_1(t), u(t)), \quad (3.40)$$
where \( K = (k_1 \ k_2 \ \cdots \ k_L) \). The problem of estimation of the parameter vector

\[
\theta = \text{col} \begin{pmatrix}
A \\
B \\
K \\
R
\end{pmatrix}
\]

(3.41)

is considered in sub-sections 3.4.1 and 3.4.2.

### 3.4.1 RPE Method for Discrete-Time and Delta-Operator Models

A global recursive prediction error (RPE) method for the estimation of network parameters for discrete-time or delta-operator models is proposed in this sub-section. Rewrite the observer equation 3.40 for discrete time instants \( t = kT \) as

\[
p\hat{x}(k) = E\hat{x}(k) + f_{LM}(\theta, \hat{x}_1(k), u(k), e(k), v(k)).
\]

(3.42)

The sensitivity model for this observer is given by differentiating its equation with respect to \( \theta \) and replacing the operators \( p \) and \( d/d\theta^T \)

\[
p \frac{d\hat{x}(k)}{d\theta^T} = \frac{d(E\hat{x}(k))}{d\theta^T} + \frac{df_{LM}}{d\theta^T},
\]

(3.43)

where

\[
\frac{df_{LM}}{d\theta^T} = \frac{\partial f_{LM}}{\partial \hat{x}_1} \frac{d\hat{x}_1}{d\theta^T} + \frac{\partial f_{LM}}{\partial e} \frac{de}{d\theta^T}
\]

\[
+ \frac{\partial f_{LM}}{\partial v} \frac{dv}{d\theta^T} + \frac{\partial f_{LM}}{\partial \theta^T}.
\]

(3.44)

In order to simplify the above expression, the following relationships have to be considered:

\[
\frac{d(E\hat{x}(k))}{d\theta^T} = E \frac{d\hat{x}(k)}{d\theta^T}
\]

(3.45)

\[
\frac{\partial f_{LM}}{\partial \hat{x}_1} = Av
\]

(3.46)

\[
\frac{\partial f_{LM}}{\partial e} = Kv
\]

(3.47)
\[
\frac{\partial f_{LM}}{\partial v^T} = A\hat{x}_1 + Bu + Ke + R \tag{3.48}
\]
\[
\frac{d\hat{x}_1}{d\theta^T} = c^T \frac{d\hat{x}}{d\theta^T} \tag{3.49}
\]
\[
\frac{de}{d\theta^T} = -c^T \frac{d\hat{x}}{d\theta^T} \tag{3.50}
\]
\[
\frac{dv}{d\theta^T} = \frac{\partial v}{\partial \hat{x}^T} \frac{d\hat{x}}{d\theta^T} + \frac{\partial v}{\partial \theta^T} = \begin{pmatrix} \frac{\partial v}{\partial \hat{x}_1} & 0 & \cdots & 0 \end{pmatrix} \frac{d\hat{x}}{d\theta^T} = \frac{\partial v}{\partial \hat{x}_1} c^T \frac{d\hat{x}}{d\theta^T} \tag{3.51}
\]
\[
\frac{\partial f_{LM}}{\partial \theta^T} = v^T \otimes \begin{pmatrix} I_n & \hat{x}_1 & I_n \hat{x}_1 & \hat{x}_1 u & \hat{x}_1 I_n \end{pmatrix} = v^T \otimes \begin{pmatrix} \hat{x}_1 & u & e & 1 \end{pmatrix} \otimes I_n. \tag{3.52}
\]

Now substitute these values in equation 3.44
\[
\frac{df_{LM}}{d\theta^T} = (Av - Kv + (A\hat{x}_1 + Bu + Ke + R) \frac{\partial v}{\partial \hat{x}_1} )c^T \frac{d\hat{x}}{d\theta^T} + v^T \otimes \begin{pmatrix} \hat{x}_1 & u & e & 1 \end{pmatrix} \otimes I_n. \tag{3.53}
\]

The complete model for the propagation of the sensitivity matrix \( W(k) = \frac{d\hat{x}(k)}{d\theta^T} \) can be rewritten as
\[
pW(k) = FW(k) + M(k), \tag{3.54}
\]
where
\[
F = E + (Av - Kv + (A\hat{x}_1 + Bu + Ke + R) \frac{\partial v}{\partial \hat{x}_1} )c^T \tag{3.55}
\]
and
\[
M(k) = v^T \otimes \begin{pmatrix} \hat{x}_1(k) & u(k) & e(k) & 1 \end{pmatrix} \otimes I_n. \tag{3.56}
\]

The complete estimation algorithm for discrete-time or delta-operator models is given in Table 3.3.
Table 3.3: Global recursive prediction error method

<table>
<thead>
<tr>
<th>Parameter vector</th>
<th>$\hat{\Theta} = \text{col}(\hat{\Theta})$, with $\hat{\Theta} = (\hat{A}^T \hat{B}^T \hat{K}^T \hat{R})^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimation error</td>
<td>$e(k) = y(k) - c^T \hat{x}(k)$</td>
</tr>
<tr>
<td>Error gradient</td>
<td>$\psi(k) = \mathbf{W}^I(k)c$</td>
</tr>
<tr>
<td>Determine validity vector $v(k)$ and $\dot{v}(k) = \left. \frac{\partial u(t)}{\partial \hat{x}_1(t)} \right</td>
<td>_{t=kT}$ for current data.</td>
</tr>
<tr>
<td>Parameter adaptation gain</td>
<td>$S(k) = \lambda(k) + \psi^I(k)P(k-1)\psi(k)$</td>
</tr>
<tr>
<td>Parameter update</td>
<td>$\hat{\theta}(k) = \hat{\theta}(k-1) + l(k)e(k)$</td>
</tr>
<tr>
<td>Covariance update</td>
<td>$P(k) = (P(k-1) - l(k)S(k)l^I(k))/\lambda(k)$</td>
</tr>
<tr>
<td>State estimate</td>
<td>$p\hat{x}(k) = E\hat{x}(k) + f_{LM}(\hat{\theta}(k), \hat{x}_1(k), u(k), e(k), v(k))$</td>
</tr>
<tr>
<td>Sensitivity model</td>
<td>$p\mathbf{W}(k) = F(k)\mathbf{W}(k) + M(k)$</td>
</tr>
<tr>
<td>with $M(k) = v^I \otimes (\hat{x}_1(k) \ u(k) \ e(k) \ 1) \otimes I_n$</td>
<td></td>
</tr>
<tr>
<td>and $F(k) = E + (\hat{A} - \hat{K})v(k)c^T$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$+ (\hat{A}\hat{x}_1(k) + \hat{B}u(k) + \hat{K}e(k))$</td>
</tr>
<tr>
<td></td>
<td>$+ \hat{R}\dot{v}(k)c^T$</td>
</tr>
</tbody>
</table>

3.4.2 Continuous-Discrete Method

In this section a global continuous-discrete method for the estimation of local model parameters is introduced. This method is similar to the adaptive extended Kalman filter (Gelb, 1974; Bohn, 2000). The estimates of system states are propagated using a continuous-time model

$$\dot{\hat{x}}(t) = E\hat{x}(t) + (A\hat{x}_1(t) + Bu(t) + R)v(\hat{x}_1(t), u(t)).$$ (3.57)

As the measured system output $y(k)$ is available only at sample hits, a correction of the states can be applied on each of them by

$$\hat{x}(k^+) = \hat{x}(k^-) + Ke(k)v(\hat{x}_1(k), u(k)), $$ (3.58)
where superscripts $-\,$ and $+\,$ represent the instants just before and just after
the correction is applied. Define the following parameter vectors:

$$
\theta = (\theta_p^T \theta_K^T)^T \quad (3.59)
$$

$$
\theta_p = \text{col}((A^T B^T R^T)^T) \quad (3.59)
$$

$$
\theta_K = \text{col}(K).
$$

Rewrite the equation 3.57 in the following form

$$
\dot{x}(t) = E \dot{x}(t) + f_{LM1}(\theta_p, \dot{x}_1, u, v), \quad (3.60)
$$

where $f_{LM1}$ is the mapping realised by a local model network. Differentiate
the above equation with respect to $\theta_p$ to derive a sensitivity propagation model
for $\frac{d\dot{x}(t)}{d\theta_p^T}$.

$$
\frac{d\dot{x}(t)}{d\theta_p^T} = \frac{d(E \dot{x}(t))}{d\theta_p^T} + \frac{df_{LM1}}{d\theta_p^T}, \quad (3.61)
$$

where

$$
\frac{df_{LM1}}{d\theta_p^T} = \frac{\partial f_{LM1}}{\partial \dot{x}_1} \frac{d\dot{x}_1}{d\theta_p^T} + \frac{\partial f_{LM1}}{\partial v^T} \frac{dv}{d\theta_p^T} + \frac{\partial f_{LM1}}{\partial \theta_p^T}.
$$

Substituting these values in equation 3.62 yields

$$
\frac{df_{LM1}}{d\theta_p^T} = (Av + (A\dot{x}_1 + Bu + R) \frac{\partial v}{\partial \dot{x}_1}) c^T \frac{d\dot{x}}{d\theta_p^T} + v^T \otimes (\dot{x}_1 \ u \ 1) \otimes I_n. \quad (3.63)
$$

The model for the propagation of $\mathcal{W}_p(t) = \frac{d\dot{x}(t)}{d\theta_p^T}$ can be written as

$$
\dot{\mathcal{W}}_p(t) = F \mathcal{W}_p(t) + M(t), \quad (3.64)
$$

where

$$
F = E + (Av + (A\dot{x}_1 + Bu + R) \frac{\partial v}{\partial \dot{x}_1}) c^T \quad (3.65)
$$

and

$$
M(t) = v^T \otimes (\dot{x}_1(t) \ u(t) \ 1) \otimes I_n. \quad (3.66)
$$
Table 3.4: Global recursive prediction error method for a continuous-discrete setup

<table>
<thead>
<tr>
<th>Parameter vector</th>
<th>[ \theta = \begin{pmatrix} \theta_p \ \theta_K \end{pmatrix}, \quad \text{with} \quad \theta_p = \text{col}((A^T B^T R^T)^T) \quad \theta_K = \text{col}(K) ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>State propagation</td>
<td>[ \hat{x}(t) = E\hat{x}(t) + (\hat{A}\hat{x}_1 + \hat{B}u(t) + \hat{R})v(\hat{x}_1, u) ]</td>
</tr>
</tbody>
</table>
| \( T(k-1)^+ \leq t \leq T(k^-) \) | \[ \begin{align*} \dot{\mathbf{W}}_p(t) &= F(t)W_p(t) + M(t) \\
\mathbf{W}_K(t) &= F(t)W_K(t) \\
\text{with} \quad M(t) &= v(t)^T \otimes (\hat{x}_1(t) u(t) 1) \otimes I_n \\
F(t) &= E + \hat{A}v(t)c^T \\
&\quad + (\hat{A}\hat{x}_1(t) + \hat{B}u(t) + \hat{R}) \frac{\partial v(t)}{\partial \hat{x}_1(t)} c^T \end{align*} \] |
| Sensitivity propagation | \[ \text{Determine validity vector} \quad v(k) \quad \text{and} \quad \dot{v}(k) = \frac{\partial v(t)}{\partial \hat{x}_1(t)} \bigg|_{t=kT} \quad \text{at sample hit.} \] |
| Estimation error | \[ e(k) = y(k) - c^T \hat{x}(k^-) \] |
| Error gradient | \[ \psi(k) = \begin{pmatrix} \mathbf{W}_p^T(k^-) \\ \mathbf{W}_K^T(k^-) \end{pmatrix} c \] |
| Parameter adaptation gain | \[ S(k) = \lambda(k) + \psi^T(k)P(k-1)\psi(k) \] |
| Parameter update | \[ l(k) = P(k-1)\psi(k)/S(k) \] |
| Covariance update | \[ P(k) = (P(k-1) - l(k)S(k)l^T(k))/\lambda(k) \] |
| State correction | \[ \hat{x}(k^+) = \hat{x}(k^-) + Kv(k)e(k) \] |
| Sensitivity correction | \[ \begin{align*} \mathbf{W}_p(k^+) &= \mathbf{W}_p(k^-) \\
&\quad - \hat{K}(v(k) - e(k^-)\dot{v}(k))c^T\mathbf{W}_p(k^-) \\
\mathbf{W}_K(k^+) &= \mathbf{W}_K(k^-) \\
&\quad - \hat{K}(v(k) - e(k)\dot{v}(k))c^T\mathbf{W}_K(k^-) \\
&\quad + v(k)^T \otimes I_n e(k) \end{align*} \] |
Similarly, an expression for the propagation of $\mathbf{W}_K(t) = \frac{d\mathbf{x}(t)}{d\theta_K}$ can be given as

$$\dot{\mathbf{W}}_K(t) = F\mathbf{W}_K(t).$$ (3.67)

In addition to the calculation of the sensitivity propagation equations, expressions for discrete-time sensitivity corrections have to be derived. Differentiating equation 3.58 with respect to $\theta_p$ and $\theta_K$ respectively:

$$\mathbf{W}_p(k^+) = \mathbf{W}_p(k^-) - K(v - e \frac{\partial v}{\partial \mathbf{x}_1})c^T \mathbf{W}_p(k^-)$$ (3.68)

$$\mathbf{W}_K(k^+) = \mathbf{W}_K(k^-) - K(v - e \frac{\partial v}{\partial \mathbf{x}_1})c^T \mathbf{W}_K(k^-) + v^T \otimes I_{n\epsilon}(k).$$ (3.69)

The complete algorithm is summarised in Table 3.4.

It should be mentioned that the estimation of the offset parameters of local models, which is not possible by local learning techniques becomes possible in case of global learning methods. The reason is that the global estimation methods consider the validity functions of local models to generate regression signals (see equation 3.56, 3.66). As a consequence these regression signals do not go to steady state and lose excitation. But it should be noted that the absolute values of the first $n - 1$ offset terms of local models have no significance. The terms $r_1...r_{n-1}$ of one local model in an LLM network can be fixed to zero as a reference. This allows to estimate other $(L - 1)n + 1$ offset terms of the network.

### 3.5 Local Versus Global Learning

This section deals with the comparison between local and global learning techniques proposed in previous sections. One obvious advantage of local techniques is the small number of parameters to be updated in each recursive update step. In global learning this number seems to be large and dependent on the number of local models in the network. This fact makes the global algorithms expensive as far as computer memory is concerned. If data are not wandering quickly among the local models, then one can see that the gradients related to most of the local model parameters remain almost zero. The
learning algorithm is realised in such a way that only the parameters with nonzero sensitivities are updated. This way the computational cost of global learning algorithms does not rise with the increasing number of local models in the network.

In order to compare estimation and convergence properties of both types of techniques the following simulation example is considered. The continuous-time system to be identified is described as

\[
\begin{pmatrix}
\dot{x}_1(t) \\
\dot{x}_2(t)
\end{pmatrix} = \begin{pmatrix}
x_2(t) \\
0
\end{pmatrix} + \sum_{i=1}^{2} (-a_i x_1(t) + b_i u(t)) v_i(t),
\]

(3.70)

where

\[
a_1 = \begin{pmatrix}
1 & 2 \\
8 & 0
\end{pmatrix} \quad a_2 = \begin{pmatrix}
3 \\
2
\end{pmatrix}
\]

(3.71)

and

\[
b_1 = \begin{pmatrix}
4 \\
2
\end{pmatrix} \quad b_2 = \begin{pmatrix}
1 \\
0.5
\end{pmatrix}
\]

(3.72)

The validity functions \(v_1\) and \(v_2\) are described as

\[
v_1(t) = \left(1 + \left|\frac{x_1(t) + 2}{2}\right|^8\right)^{-1} \quad v_2(t) = \left(1 + \left|\frac{x_1(t) - 2}{2}\right|^8\right)^{-1}
\]

(3.73)

It is supposed that the validity functions and the order of the local models are known. The parameters \(a_1, a_2, b_1, b_2\) are unknown and have to be estimated. Local and global estimation techniques are applied to estimate the parameters in a delta-operator model form. The estimation results are drawn in figures 3.2 and 3.3.
Figure 3.2: Parameter estimation with local learning

Figure 3.3: Parameter estimation with global learning
Chapter 3. Identification of Local Linear State-Space Models

The estimates delivered by the global learning algorithm converge to their steady-state values. These steady-state parameter values are very close to the original continuous-time parameters. This deviation is dependent on the sampling period and becomes zero as sampling period approaches to zero. Whereas the estimates delivered by local learning algorithm do not converge. The reason is that the system input signal is selected in such a way that the scheduling variable $x_1(t)$ is fluctuating frequently between the activation regions of both models. This fluctuation causes a frequent switching of the identification between these local models. Due to neglecting the overlap effects the information contained by the vector $\psi$ is invalid for both models. In order to demonstrate this effect, in another simulation experiment the input signal is chosen in such a way that model switching is less frequent. The results of local learning with this input signal are drawn in Figure 3.4.

![Figure 3.4: Parameter estimation with local learning (slow switching)](image)

The grey-shaded time span in this figure is the period, when the local model 1 is active and the parameters of this model are estimated. During the time span with white background, the parameters of the local model 2 are estimated. From the results it is clear that as long as the data lie in a single model, parameter estimates of this model tend to move towards real parameter...
values shown by dashed lines. But 'just before' and 'just after' the switching these estimates are destroyed.

3.6 Chapter Summary

In this chapter various schemes for the estimation of local linear state-space models are introduced. In Section 3.1 an observer canonical form for non-linear systems is presented. Section 3.2 deals with problem of estimation of continuous-time models from sampled data. Techniques based on delta-operator model and continuous-discrete setup are discussed. In Section 3.3 two local learning algorithms for the estimation of local linear state-space models are presented. These algorithms determine the most active model at current operating conditions and update its parameters recursively neglecting the non-linearity caused by overlap factors. The algorithm given in Sub-section 3.3.1 is based on the state-variable filter method. Predefined state variable filters are used to generate regression signals and a recursive least squares algorithm (RLS) is applied for the parameter estimation. This method is an extension of the procedure proposed by Kreisselmeier (1977) for affine systems. The recursive prediction error method, given in Sub-section 3.3.2, may be considered as a generalisation of the state-variable filter method, where the filters are no more constant but adaptive and the prediction error is minimised by using the recursive Gauss-Newton method. Section 3.4 deals with global learning methods for blended state-space models. These methods minimise a global cost function in order to estimate the parameters of the whole blended model simultaneously. Sub-section 3.4.1 deals with a method based on the prediction error minimisation for discrete-time or delta-operator models. Sub-section 3.4.2 describes a global method for the identification of continuous-time local linear state-space models in continuous-discrete setup. In Section 3.5 global and local learning techniques are compared. With the help of a simulation example, it is shown that the local learning techniques, which are reported to be preferable for the estimation of local input-output models (Murray-Smith and Johansen, 1997), do not seem to be the better choice in case of state-space and continuous-time models.
Chapter 4

Controller Design Based on Input-Output Models

One way to control a nonlinear system is to linearise it around an operating point, approximate the dynamics using a linear model and design a controller based on this linear model. This approach has attracted the attention of researchers for a long time, because it provides quite satisfactory results in the normal range of operation of a system. But the dynamical systems, from a control point of view, are becoming more and more complex as a direct consequence of the technological development. The complexity of such systems exhibits itself in the form of strongly coupled nonlinear subsystems and in wide operating ranges. Adaptive control based on the linear model of such a system around some operating point often fails to provide a satisfactory control solution. Nonlinear control schemes are necessary for such cases.

Neural networks provide a basis for the approximation of nonlinearities and implementation of learning control schemes. Approaches reported in the literature to tackle the nonlinear control problem with neural networks include internal model control, direct or indirect inverse control, feedback linearisation, model reference adaptive control, neural predictive control (Hunt et al., 1992; Narendra and Mukhopadhyay, 1997; Ng, 1997). All these schemes use neural networks as black-box approximators. Static or dynamic gradient methods
are used to adjust the parameters of these controller networks. The major
drawback of such a control scheme is that the controller network needs an
extensive pre-training in simulations before it can be used in the real control
loop.

Local model networks tackle the problem in another way. The input space
of the nonlinear function involved in NARX representation of a dynamical
system is partitioned into several regions in such a way that the dynamics
of the system may be approximated by locally valid linear models within
some predefined error bounds. This property of local model networks provides
valuable information about the local behaviour of the system, which can be
utilised in model-based controller design (Ali, 1998). Most LMN-based control
schemes use two networks. One network is responsible for the identification
of the plant, while the other behaves as controller (Junge, 1999; Hunt and Jo-
hansen, 1997). Corresponding to each local model in the model network there
is a local controller in the controller network. Local controllers are dependent
on their model network counterparts for parameter update and validity func-
tions. Such schemes are known as master-slave schemes, the model network
being the master and the controller network being the slave. The major advan-
tage of master-slave techniques is that each local controller can be designed on
the basis of local model parameters using the techniques known from the linear
system theory, i.e. pole-placement, LQ-regulator design etc. The behaviour
of each local model given by equation 2.12 is not linear but affine. The offset
term $r_j$ in this equation is operating-point-dependent and time-varying. For
the design of local controllers in addition to considering polynomials $A_j$ and
$B_j$ this offset should be compensated. Considering $r_j$ not as a parameter but
as an external disturbance to the system, two compensation approaches are
often used. In the first approach a direct compensation is devised by subtracting
$r_j/B_j(z^{-1})$ from the input of the system. The basic requirement for this
type of compensation is the availability of an accurate model. Besides, the
polynomial $B_j(z^{-1})$ must be a Hurwitz polynomial. The other scheme, which
is robust against model inaccuracies and does not call for a stable inverse of
the local model is the inclusion of an integrator in the control loop. The ap-
lication of any of the above techniques allows to consider the local model as
a pure linear system for further controller design purposes.

In this chapter three new control schemes for nonlinear systems on the basis of LLM networks are proposed. Section 4.1 deals with a model reference control scheme using local linear model networks. In Section 4.2, a strategy for adaptive feedforward cancellation of measurable disturbances for a class of nonlinear systems is proposed. Both schemes are based on a master-slave strategy. In Section 4.3, a model predictive control scheme for nonlinear systems using local linear models is described. Though this predictive control setup utilises the local linear properties of LLM networks, but it does not fall into the category of master-slave schemes. In Section 4.4, the chapter is summarised, and merits and demerits of master-slave and adaptive predictive control schemes are discussed.

4.1 Model Reference Adaptive Control

The idea behind this technique is to design a nonlinear controller network for a nonlinear system such that the overall closed-loop system behaves like a given linear model. Figure 4.1 depicts the underlying structure of this master-slave local linear model reference control system. The LLM network is trained by the learning algorithm to estimate the NARX model of the plant. The Δ-block, which precedes the LLM network, is responsible for the generation of regressors needed by the NARX model. On the basis of currently available estimates of LLM network parameters \( p(k) \), the adaptation algorithm determines the parameters \( p_c(k) \) of the controller network. The validity vector \( v(k) \) available from the LLM network is used by the controller network to activate local controllers. In order to determine the parameters of a local controller \( j \), consider the transfer function of the local linear model given in equation 2.12 without offset.

\[
\frac{Y(z)}{U(z)} = \frac{B(z^{-1})}{A(z^{-1})}z^{-d},
\]

where \( A(z^{-1}) \) and \( B(z^{-1}) \) are polynomials in \( z^{-1} \) with real coefficients and \( d \) is the discrete transmission delay, which is selected in a way that the absolute coefficient in polynomial \( B \) is nonzero. Furthermore, the polynomials \( A \) and \( B \)
are assumed to be relatively prime. The subscript $j$ is removed in this equation for the sake of simplicity. The principle of the model reference adaptive control is to design a controller in such a way that the closed-loop output follows the output of the reference model excited with the same command input $w$ as long as the plant is operated in this local region of operation. The control law for a linear model reference controller (Åström and Wittenmark, 1995) is given by

$$U(z) = \frac{T(z^{-1})W(z) - S(z^{-1})Y(z)}{R(z^{-1})I(z^{-1})}, \quad (4.2)$$

where $R(z^{-1}), S(z^{-1})$ and $T(z^{-1})$ are controller polynomials. The polynomial $I(z^{-1}) = (1 - z^{-1})^{f_I}$ enables the controller to have a conditional integral behaviour. The exponent $f_I$ is set equal to one if an additional integrator is desired in the closed loop. The closed-loop transfer function of the system
becomes
\[
\frac{Y(z)}{W(z)} = \frac{T(z^{-1})B(z^{-1})}{R(z^{-1})I(z^{-1})A(z^{-1}) + S(z^{-1})z^{-d}}.
\]
(4.3)

The reference model can be described by the pulse transfer function \(G_m(z)\) given by
\[
G_m(z) = \frac{Y_m(z)}{W(z)} = \frac{B_m(z^{-1})}{A_m(z^{-1})}z^{-d_m},
\]
(4.4)

where \(d_m > d\).

In order to achieve that the closed-loop transfer function is identical to that of the reference model, the controller polynomials should satisfy the following conditions
\[
T(z^{-1}) = B_m(z^{-1})z^{-(d_m - d)},
\]
(4.5)
\[
R(z^{-1}) = R'(z^{-1})B(z^{-1})
\]
(4.6)

and
\[
R'(z^{-1})I(z^{-1})A(z^{-1}) + S(z^{-1})z^{-d} = A_m(z^{-1}).
\]
(4.7)

The solution of equation 4.7 completes the controller design. In these design equations the plant zeros are cancelled directly by the controller poles. If the system has zeros outside or on the unit circle then the control signal may become unstable and as a consequence the whole system may lose stability. In order to avoid this situation, a modified controller structure is introduced (Hahn, 1983; Nöth, 1982). A corrective filter given by equation 4.8 is connected parallel to the plant and its parameters are selected in such a way that the resulting extended plant, comprising both the given plant and the corrective filter, has a numerator polynomial, which possesses stable zeros.

\[
Y_k(z) = \frac{B_k(z^{-1})}{A_k(z^{-1})}z^{-d_k}U_d(z)
\]
(4.8)

For \(d_k = d\), the output \(Y_e(z)\) of the extended plant is given by
\[
Y_e(z) = \frac{B(z^{-1})A_k(z^{-1}) + A(z^{-1})B_k(z^{-1})}{A(z^{-1})I(z^{-1})A_k(z^{-1})}z^{-d}U(z).
\]
(4.9)

The extended plant has an order higher than the original plant. The design problem that remains is to determine the numerator and the denominator
polynomials of the corrective filter. This problem may be considered as a zero-placement problem. Any stable polynomial

\[ P(z^{-1}) = (1 - z^{-1})A(z^{-1})B_k'(z^{-1}) + B(z^{-1})A_k(z^{-1}) \]  

(4.10)
can be assigned to the numerator of the extended plant, where

\[ B_k(z^{-1}) = (1 - z^{-1})^{(1 - f_r)}B_k'(z^{-1}) \]  

(4.11)
ensures the steady state value of \( y_k \) to be zero. The stable polynomial \( P(z^{-1}) \) can be obtained, for example, by the process of spectral factorisation applied to the self-reciprocal polynomial \( A(z^{-1})A(z)B(z^{-1})B(z) \). The Diophantine equation 4.10 should be solved to determine the unknown polynomials \( B_k'(z^{-1}) \) and \( A_k(z^{-1}) \). The corrective filter thus achieves its objective; it makes the numerator polynomial of the extended plant, comprising both the plant and the corrective filter itself, a Hurwitz polynomial without making any contribution to the steady state output of the extended plant. The controller component without integrator has a single output \( u_d \), and three inputs, the command signal \( w \), the measured plant output \( y \), and the output \( y_k \) of the corrective filter. The relationship for the signal \( u_d \) can be written as

\[ R(z^{-1})U_d(z) = T(z^{-1})W(z) - S_s(z^{-1})Y(z) - S_k(z^{-1})Y_k(z). \]  

(4.12)
Considering the transfer function of the corrective filter, the above expression can be simplified as

\[ (A_k(z^{-1})R(z^{-1}) + S_k(z^{-1})B_k(z^{-1})z^{-d_k})U_d(z) = A_k(z^{-1})T(z^{-1})W(z) - A_k(z^{-1})S_s(z^{-1})Y(z), \]  

(4.13)
where \( W(z) \), \( Y(z) \) and \( Y_k(z) \) are the \( z \)-transforms of the signals \( w(k) \), \( y(k) \) and \( y_k(k) \) respectively. Equation 4.13 represents the control law without integrator. Integrated control signal \( U(z) \) is given as

\[ U(z) = \frac{U_d(z)}{(1 - z^{-1})^{(1 - f_r)}}. \]  

(4.14)
\( R(z^{-1}) \), \( S_s(z^{-1}) \), \( S_k(z^{-1}) \) and \( T(z^{-1}) \) in equation 4.13 are controller polynomials, which should be designed so that the closed-loop transfer function becomes
equal to $G_m(z)$. This condition is satisfied if the following four equations hold

\[ R(z^{-1}) = R_s(z^{-1})B(z^{-1}) + R_k(z^{-1})B_k(z^{-1}), \quad (4.15) \]

\[ A_m(z^{-1}) = I(z^{-1})R_s(z^{-1})A(z^{-1}) + S_s(z^{-1})z^{-d}, \quad (4.16) \]

\[ A_m(z^{-1}) = R_k(z^{-1})A_k(z^{-1}) + S_k(z^{-1})z^{-d} \quad (4.17) \]

and

\[ T_j(z^{-1}) = B_m(z^{-1})z^{-(d_m-d)}. \quad (4.18) \]

Equations 4.15 to 4.18 are solved to determine the required controller polynomials. The relationship given by equation 4.13 can be reformulated in time domain as

\[ u_d(k) = p_c^T \varphi_c(k), \quad (4.19) \]

where the vector $\varphi_c(k)$ consists of delayed values of $w$, $y$ and $u_d$, and the parameter vector $p_c$ can be calculated from the polynomial coefficients and represents the parameters of a local linear controller in the controller network. The output of the controller network is calculated by blending outputs of all the local controllers as $\sum_{j=1}^L u_d_j(k)$. The additional integrator in the control loop is not a part of the local controllers, as the parameters of this part are always the same. Therefore, there is no sense to pack it as a part of each local controller.

### 4.2 Nonlinear Adaptive Feedforward Cancellation

Tracking of reference or command signals and rejection of disturbances are the major aims when devising a control system. Feedback controllers can successfully suppress the effects of disturbances in steady state if the model of the disturbance is included in the controller (internal model control)(Keuchel, 1988). For example, if a plant is exposed to step-like disturbances, then the inclusion of an integrator in the closed loop can achieve a control without steady-state deviation. But if the disturbance is a sinusoid of a certain frequency then the controller must contain these oscillatory dynamics in order to reject the
disturbance effects. When the disturbances are of varying frequency then a continuous online tuning of the controller is required. The above scheme cannot offer a good solution especially in transient phase, when the disturbances are of varying nature or non-deterministic. If these non-deterministic disturbances are measurable, for example electric load on a power plant, pressure fluctuations on the compressed air supply of a turbo-generator set, variations in sunshine in case of a solar plant, then it is advantageous to compensate such disturbances in a feedforward control. This problem of direct compensation of measurable disturbances is well solved for linear systems, where the effect of disturbance is multiplied with the inverse model of the plant and then subtracted from the control signal (D’Azzo and Houpis, 1966; Unbehauen, 1987). Unfortunately this scheme does not work for nonlinear systems. The following approach is an effort to solve this problem for a certain class of nonlinear systems. This approach uses an LLM network to model the system dynamics and to predict the effect of measurable disturbances. Another local linear feedforward cancellation network (LLF network) is used to implement the nonlinear feedforward cancellation based on this prediction. A local linear controller (LLC) network is included for tracking purposes. This scheme is sketched in Figure 4.2 and is applicable if the nonlinear system can be expressed as a linear combination of two nonlinear sub-systems independently driven by control and disturbance signals (Ali et al., 2001).

Before designing the LLF network, consider a linear discrete-time dynamical system with a single control input $u$ and single output $y$ but $l$ measurable disturbances $z_i$. The system behaviour is described in $z$-domain by the following equation

$$Y(z) = \frac{B(z^{-1})}{A(z^{-1})} z^{-d} U(z) + \sum_{i=1}^{l} \frac{C_i(z^{-1})}{D_i(z^{-1})} z^{-d_i} Z_i(z). \quad (4.20)$$

A control law consisting of the following two components can be proposed for this system

$$U(z) = \hat{U}_b(z) - U_f(z). \quad (4.21)$$

Here $\hat{U}_b(z)$ is a the control effort calculated by the feedback controller, which is responsible for the tracking of a time-varying reference signal and suppresses
the effect of un-modelled dynamics, whereas $U_f(z)$ is the feedforward cancellation of $l$ measurable disturbance signals $Z_i(z)$. If the polynomial $B(z^{-1})$ is a Hurwitz polynomial, then $U_f(z)$ can be calculated as

$$U_f(z) = \frac{A(z^{-1})}{B(z^{-1})} \sum_{i=1}^{l} \frac{C_i(z^{-1})}{D_i(z^{-1})} z^{d-d_i} Z_i(z) \quad \text{for} \quad d \leq d_i, \forall i. \quad (4.22)$$

If the polynomial $B(z^{-1})$ has zeros outside or on the unit circle then the control signal of equation 4.22 may become unstable and as a consequence the whole system may lose stability. In order to avoid this problem a corrective filter, as described in the previous section, is connected parallel to the plant.

Such a disturbance cancellation can not be applied to general nonlinear systems because the principle of superposition is not valid for such systems. But if the system output can be described as the sum of outputs of two nonlinear systems then a scheme can be suggested to cancel the effect of the nonlinear disturbance system. This means that the system should be described or can
be reasonably approximated in the following form

\[ y(k) = y_u(k) + y_z(k) + \epsilon(k), \]  \hspace{1cm} (4.23)

where

\[ y_u(k) = f_u(y_u(k-1), \ldots, y_u(k-n_{y_u}), u(k-d), \ldots, u(k-n_u-d)) \]  \hspace{1cm} (4.24)

and

\[ y_z(k) = f_z(y_z(k-1), \ldots, y_z(k-n_{y_z}), z(k-d_z), \ldots, z(k-n_z-d_z)). \]  \hspace{1cm} (4.25)

Signals \( y_u \) and \( y_z \) are not measurable. The idea is to estimate or predict \( y_z \) using a nonlinear model and cancel its effect in a feedforward fashion. The control system shown in Figure 4.2 consists of three major blocks: identification, feedback controller and adaptive feedforward cancellation (AFC) block.

The **identification block** is composed of an LLM network preceded by a \( \Delta \)-block responsible for the generation of regression vector for the network. As the signals \( y_u(k) \) and \( y_z(k) \) are not assumed to be measurable, the whole system (equations 4.23, 4.24, 4.25) is described as the following single NARX model

\[ y(k) = f(y(k-1), \ldots, y(k-n^*_y), u(k-d), \ldots, u(k-n^*_u-d), \]  \hspace{1cm} (4.26)

\[ z(k-d_z), \ldots, z(k-n^*_z-d_z)) + \epsilon(k), \]

where \( n^*_y = n_{y_u} + n_{y_z}; \) \( n^*_u = n_{u} + n_{y_z}; \) \( n^*_z = n_{z} + n_{y_u}. \) The approximation of the function \( f \) is achieved by using the LLM network. The model of the \( j \)th unit in the hidden layer of the LLM network can be described by

\[ A^*_j(z^{-1})\hat{Y}(z) = B^*_j(z^{-1})z^{-d}U(z) + C^*_j(z^{-1})z^{-d_z}Z(z) + r_j, \]  \hspace{1cm} (4.27)

where

\[ A^*_j(z^{-1}) = A_j(z^{-1})D_j(z^{-1}) \]

\[ B^*_j(z^{-1}) = B_j(z^{-1})D_j(z^{-1}) \]  \hspace{1cm} (4.28)

\[ C^*_j(z^{-1}) = C_j(z^{-1})A_j(z^{-1}). \]
The minimal realisation of the transfer function \( \frac{B_j(z^{-1})}{A_j(z^{-1})} \) can be achieved by cancelling the common roots of polynomials \( A_j^* \) and \( B_j^* \), which, in practice, due to numerical reasons, is almost never the case. The controller block consists of a local linear controller (LLC) network preceded by a \( \Delta \)-block. The number of local controllers in this network is equal to the number of LLM models in the model network. The parameters of each linear controller can be updated using adaptive pole placement or linear quadratic optimisation techniques discussed in (Junge, 1999).

The adaptive feedforward cancellation (AFC) block consists of two components: prediction and cancellation. The prediction component delivers a \( d \)-step-ahead prediction of the disturbance effect \( y_z \) using the LLM network model of the plant, while the cancellation block is responsible for the generation of the proper control signal \( u_f \), which would nullify the disturbance. These components are described briefly in the following.

The Prediction Component: As it is assumed that the system output \( y \) is composed of two components \( y_u \) and \( y_z \). It is plausible to use the model network LLM for the prediction of signal \( y_z \). For this purpose the function approximated by the LLM network can be written as

\[
\hat{y}_z(k + d) = \hat{f}(\hat{y}_z(k + d - 1), \ldots, \hat{y}_z(k + d - n_y^*), 0, \ldots, 0, z(k + d - d_z), \ldots, z(k + d - n_z^* - d_z)).
\]  

(4.29)

The Cancellation Component: As discussed above for linear systems, if the prediction \( \hat{y}_z(k + d) \) of the disturbance effect is available, then it can be multiplied with the inverse model of the extended plant to achieve the cancellation signal \( u_f(k) \). Since the nonlinear model is composed of local linear units, this property of the network can be exploited to design a network composed of local linear cancellation units corresponding to each linear unit in the LLM network. The transfer function of the cancellation unit \( j \) in LLF network under the consideration of a local corrective filter is given as

\[
G_j^f(z) = \frac{A_j(z^{-1})A_{kj}(z^{-1})}{B_j(z^{-1})A_{kj}(z^{-1}) + A_j(z^{-1})B_{kj}(z^{-1})}.
\]  

(4.30)

The input signal of LLF network preceded by \( \Delta \)-block is the prediction \( \hat{y}_z(k+d) \) of the effect of disturbance, which is available from the prediction component.
4.3 Nonlinear Model Predictive Control

In this section a model predictive control scheme for nonlinear systems using LLM networks is proposed. The model predictive control is based on a simple problem formulation, which is well suited to deal with nonlinearities and constraints. Assuming that a discrete-time mathematical model is used, the basic idea is to determine the control action \( u(k) \) at time \( t = kT \), by using the sampling time \( T \) and solving a finite-horizon optimisation problem over a time interval of \( t \in [kT, (k + N)T] \). Applying the receding-horizon principle for the next time instant \( t = (k + 1)T \), a new control \( u(k + 1) \) is found by solving a new optimisation problem for the next time interval \( t \in [(k + 1)T, (k + N + 1)T] \). In mathematical terms this method can be described using the time-invariant nonlinear model of the plant

\[
\hat{y}(k + 1) = \hat{f}([y(k), ..., y(k - n_y + 1), u(k), ..., u(k - n_u + 1)]) \tag{4.31}
\]

delivering the one-step prediction \( \hat{y}(k + 1) \). In this equation \( \hat{f} \) is an approximation of the nonlinear function \( f \) given in equation 2.7. The noise vector \( \epsilon \) and the transmission delay \( d \) are omitted in equation 4.31. Furthermore, the input and output vectors \( u(k), y(k) \) must satisfy the boundary constraints

\[
\hat{y}(k^* + 1) \in \mathcal{Y}, \quad u(k^*) \in \mathcal{U}, \quad \forall k^* \geq k, \tag{4.32}
\]

where \( \mathcal{Y} \) and \( \mathcal{U} \) are compact sets of \( \mathbb{R}^m \) and \( \mathbb{R}^r \) respectively. To be able to describe the desired control behaviour, a quadratic objective function to be minimised is defined as

\[
J(k) = \frac{1}{2} \sum_{i=1}^{N} \left( e^T(k + i)Qe(k + i) + u^T(k + i - 1)Ru^*(k + i - 1) \right), \tag{4.33}
\]

where the matrices \( Q \) and \( R \) are both symmetric and positive definite, and the vector \( e(k + i) = w(k + i) - \hat{y}(k + i) \) represents the deviation between the desired \( w(k + i) \) and predicted \( \hat{y}(k + i) \) output vector at time \( t = (k + i)T \). There are several possibilities to define \( u^* \) such as successive incremental changes \( u^*(k) = u(k) - u(k - 1) \) or deviation of control \( u^*(k) = u(k) - u_0 \) from a steady state value \( u_0 \). In this work the second version is chosen. The weighting
matrices $Q$ and $R$ can be used to influence the speed of control and to set the significance of the $r$ input and $m$ output signals for the optimisation.

It should be mentioned that the objective function in equation 4.33 solely represents one out of many possible objective functions used for predictive control. For example, considering the case of a fixed set-point $\{w(k^*) = w(k^* + 1) \forall k^* \geq k\}$, and attempting to provide an analytical stability proof for a nonlinear predictive control, it might be necessary to append the objective function in equation 4.33 with a terminal cost term $e^T(k + N)Le(k + N)$ and to append the constraints in equation 4.32 with a terminal state condition $e(k + N) \in \Omega$ (Chen and Allgöwer, 1998). These additional terms have been omitted in this work to simplify the following calculations.

Since the predictive control strategy is based on the receding-horizon principle, the optimisation problem must be solved repeatedly after each sampling instant. This property allows to consider the control law as a nonlinear function

$$
\begin{align*}
\mathbf{u}(k) &= \gamma[y(k - 1), ..., y(k - n_y),
\mathbf{u}(k - 1), ..., \mathbf{u}(k - n_u + 1),
\mathbf{w}(k), ..., \mathbf{w}(k + N - 1)],
\end{align*}
$$

which performs the optimisation task

$$
\min_{\mathbf{u}(k), ..., \mathbf{u}(k+N-1)} J(k)
$$

and returns only the first element of the solution $[\mathbf{u}(k), ..., \mathbf{u}(k + N - 1)] \in \mathbb{R}^{r \times N}$ for control. Although the elements $[\mathbf{u}(k + 1), ..., \mathbf{u}(k + N - 1)]$ resulting from the solution are usually discarded for control purposes, they can be effectively used to initiate the optimisation procedure taking place at time $t = (k + 1)T$. This is of significant importance if the optimisation is performed using an iterative method.

The primary requirements of a predictive control scheme are a good prediction of the future trajectory of the system and the calculation of gradients of objective function with respect to the proposed control sequence. A local linear model network as a nonlinear model of the plant seems to be a good candidate for fulfilling both requirements. In addition to delivering a good
prediction of system behaviour, this network also provides a set of parameters of the plant linearised at each operating point on the predicted trajectory. These parameters are used to calculate the gradients of the objective function directly rather than approximating them with other numerically intensive differentiation techniques. In this approach an LLM network is applied for this purpose, which delivers an estimate \( \hat{f} \) of the unknown plant according to the NARX-model description of equation 2.7.

Define a sequence of suggested controller outputs

\[
\bar{u}(k) = \left( u^T(k) \ldots u^T(k + N - 1) \right)^T. \tag{4.36}
\]

In order to estimate the future response of the plant to this sequence, the LLM network output can be calculated recursively to get

\[
\bar{y}(k) = \left( \hat{y}^T(k + 1) \ldots \hat{y}^T(k + N) \right)^T. \tag{4.37}
\]

Each element \( \hat{y}(l) \) of the above vector is given by the following relation

\[
\hat{y}(l) = \hat{f}[\varphi(l)], \tag{4.38}
\]

where

\[
\varphi(l) = \left[ \hat{y}^T(l - 1) \ldots \hat{y}^T(l - n_y)u^T(l - 1) \ldots u^T(l - n_u) \right]^T \tag{4.39}
\]

and

\[
\bar{y}(i) = \begin{cases} y(i) & i \leq k \\ \hat{y}(i) & \text{otherwise} \end{cases} \tag{4.40}
\]

It is clear from the nature of local linear model networks that several local linear models may be activated successively during this prediction process. Figure 4.3 demonstrates this phenomenon for a first-order SISO system. The points A and B correspond to the lower and upper boundaries of the prediction horizon, passing through the local linear models \( L_7, L_8, L_{10}, L_4, L_5 \) and \( L_6 \). In addition to predicting the system output at instant \( l \), the network also returns a parameter matrix

\[
P(l) = \left( A_1(l) \ A_2(l) \ldots A_{n_y}(l) \ B_1(l) \ B_2(l) \ldots B_{n_u}(l) \right), \tag{4.41}
\]
which contains the parameters of the active local linear model

\[ \hat{y}(l) = \sum_{i=1}^{n_u} A_i(l)\hat{y}(l-i) + \sum_{i=1}^{n_u} B_i(l)u(l-i), \]  

(4.42)

where \( A_i \in \mathbb{R}^{m \times m} \) and \( B_i \in \mathbb{R}^{m \times r} \).

This expression can be simplified using the active input pattern vector \( \varphi(l) \) and the parameter matrix \( P(l) \), such that

\[ \hat{y}(l) = P(l)\varphi(l). \]  

(4.43)

For calculating the control deviation, the desired reference signal sequence is assumed to be known over the prediction horizon

\[ \bar{w}(k) = \left( w^T(k+1) \quad \ldots \quad w^T(k+N) \right)^T. \]  

(4.44)

This allows to define an error sequence

\[ \bar{e}(k) = \left( e^T(k+1) \quad \ldots \quad e^T(k+N) \right)^T, \]  

(4.45)

where \( e(k+i) = w(k+i) - \hat{y}(k+i) \). Using this notation the objective function in equation 4.33 can be rewritten using \( u^*(k) = \bar{u}(k) - \bar{u}_0 \) as

\[ J(k) = \frac{1}{2}[\bar{e}^T(k)\bar{Q}\bar{e}(k) + \bar{u}^{*T}(k)\bar{R}\bar{u}^*(k)]. \]  

(4.46)
where
\[ Q = I_N \otimes Q \quad R = I_N \otimes R. \]  
(4.47)

To solve the optimisation task described in equation 4.35 many numerical methods have been suggested in the literature (Himmelblau, 1972). These methods usually make use of a calculated gradient to approach the minimum iteratively. In this work the 'steepest descent' method is applied to update the solution. The \( \nu \)th iteration of the method can be given as
\[ \bar{u}(k)^{(\nu)} = \bar{u}(k)^{(\nu-1)} - \text{diag}(\lambda(k)^{(\nu-1)}) \left. \frac{\partial J(k)}{\partial \bar{u}(k)} \right|_{\bar{u}(k)^{(\nu-1)}}, \]  
(4.48)

where \( \lambda(k)^{(\nu-1)} \in \mathbb{R}^{rN} \) is a weighting vector used to control the optimisation speed. An adaptive strategy is used to determine the weighting vector. A general treatment of such adaptive methods can be found in (Haykin, 1994). The gradient of the objective function with respect to \( \bar{u}(k) \) can be given as
\[ \frac{\partial J(k)}{\partial \bar{u}(k)} = \frac{\partial \bar{e}^T(k)}{\partial \bar{u}(k)} \bar{Q} \bar{e}(k) + \frac{\partial \bar{u}^* T(k)}{\partial \bar{u}(k)} \bar{R} \bar{u}^*(k). \]  
(4.49)

The signal vectors \( \bar{e}(k) \) and \( \bar{u}^*(k) \) are determined using equations 4.45 and 4.36. The gradients of these vectors can be given as
\[ \frac{\partial \bar{u}^* T(k)}{\partial \bar{u}(k)} = I_{rN} \]  
(4.50)

\[ \frac{\partial \bar{e}^T(k)}{\partial \bar{u}(k)} = - \frac{\partial \bar{y}^T(k)}{\partial \bar{u}(k)} = -G^T(k). \]  
(4.51)

The lower block triangular matrix \( \{G_{ij}\} \) is the Jacobian of \( \bar{y}(k) \) with respect to \( \bar{u}(k) \). The non-zero block elements of this matrix can be calculated recursively using parameter matrices \( A_i \) and \( B_i \) according to the following scheme:

for \( i = 1 : N \),

for \( j = 1 : i \),

if \( j > (i - n_u) \) then
\[ G_{ij}(k) = B_{i-j+1}(k+i) \]
if \( j < i \) then
\[ G_{ij}(k) = G_{ij}(k) + \sum_{l=1}^{\min(n_y,i-1)} A_l(k+i)G_{i-l,j}(k) \]
end
end
This scheme for the recursive calculation of the Jacobian of $\tilde{y}(k)$ with respect to $\tilde{u}(k)$ exploits the local linear property of the LLM network. This calculation is far simpler than the numerical differentiation techniques applied on nonlinear models like RBF and MLP networks (Halldorsson et al., 2002).

4.4 Chapter Summary and Discussion

In this chapter three new control schemes based on LLM networks are proposed. Model reference adaptive control and adaptive feedforward cancellation proposed in Sections 4.1 and 4.2 respectively fall into the category of master-slave schemes. Master-slave schemes use two networks, one as identification model (LLM network) and the other as controller. There is a one-to-one correspondence between the components of both networks. Such kind of schemes face transition problems with fast moving operating point. This problem is described in the following.

Suppose that at a time instant $k$ the $j$th local model in the LLM network is active. The LLM network will supply the current validity vector $v(k)$ to the controller network. As the controller network behaves like a slave network in terms of the validity function $v(k)$, the controller output signal $u^{(j)}(k)$ will be calculated based on the output of controller $j$. It may happen that the new $u^{(j)}(k)$ drives the nonlinear plant to such an operating region, which is represented by another local linear model $i$. In that case a situation is created where the control signal was calculated for one 'system' and is now applied to control another 'system'. This will badly affect the quality of control at and after a transition point, if the difference in the dynamics of both models is considerably large. To alleviate the unwanted consequences of such a situation, some sort of cautiousness will have to be incorporated in the calculation of control signal at transition points. The approach taken here is to do an iteration by calculating the new activity of the model network based on the next input pattern $\varphi(k+1)$, which is predictable due to the currently measured system output $y(k)$ and calculated control signal $u^{(j)}(k)$. If according to the predicted validity vector $v(k+1)$ a local linear model $i$ other than the current model $j$ becomes active, then a weighted average of the two control
signals $u^{(j)}(k)$ and $u^{(i)}(k)$ is calculated based on the Euclidean distances of the two operating points from the boundary separating the two regions. Simulation studies have shown that this kind of improvement reduces the transition problems of master-slave schemes (Ashfaq, 1999).

There are two different possibilities to manage the state of the local controllers in the network. Either the local controllers may share common states or each local controller has its own states. In the second case each local controller can be considered as a dynamical block. In the simulation studies this choice proved to be a better alternative (Ali et al., 2000).

The predictive control scheme based on LLM networks does not suffer from such transition problems. The reason is that, in predictive control, future control strategy is devised keeping these inter-model transitions in mind. The basic requirement for the success of predictive control is that all the local models, which are used in a prediction horizon, must deliver local estimates of system parameters within some error bounds. Adaptive predictive control scheme started with zero information about the system may face a problem, which is described in the following.

Suppose that a nonlinear system is being operated in a near vicinity of the operating point $P_1$. The objective is to shift the system operation to a new point $P_2$ in a limited time horizon. The adaptive predictive control scheme uses the model of the plant to predict the system behaviour with respect to the proposed control sequence in the prediction horizon. If this predicted trajectory enters to a region, which has not been previously visited by the online estimation/training algorithm, then the information (predicted behaviour and parameters) provided by the model is inaccurate or even contradicts the real system. Gradients calculated on the basis of this contradictory model may lead to the fact that the optimisation process can not converge. To avoid this problem, a rough estimate of system parameters in the whole range of operation should be known, before an adaptive predictive control can be switched on.
Chapter 5

Controller Design Based on State-Space Models

State-feedback control is a relatively modern control approach. In this approach instead of the output of a system its internal states are fed back to generate the control signal. The major hindrance in the state feedback is that generally the state vector is not completely measurable. On the basis of input-output measurements the states are to be reconstructed with the help of an observer. The control performance in such schemes is strongly dependent on the quality of state observation. If the observation error is too large, then the control quality of state-feedback controllers is generally inferior to that of output feedback controllers. In Chapter 3 methods for online identification and state observation for nonlinear systems are described. These methods estimate the parameters of local linear state-space models and reconstruct system states simultaneously. The controller design techniques considered in this chapter deal discrete-time systems, which can be described as

\[
\begin{align*}
    x(k+1) &= f(x(k), u(k)) \\
    y(k) &= h(x(k))
\end{align*}
\]  

(5.1)

where \( u \in \mathbb{R}^r \) is vector of \( r \) system inputs also known as control vector, \( y \in \mathbb{R}^m \) is the vector of \( m \) system outputs, and \( x \in \mathbb{R}^n \) is the vector of \( n \) system states.
\( f \) and \( h \) are nonlinear vector functions. These functions are approximated by LLM networks as described in Chapter 3\(^1\). These approximations \( \hat{f} \) and \( \hat{h} \) are realised by blending local linear (affine) models

\[
\begin{pmatrix}
\hat{x}(k + 1) \\
\hat{y}(k)
\end{pmatrix} =
\begin{pmatrix}
\hat{f}(\hat{x}(k), u(k)) \\
\hat{h}(\hat{x}(k))
\end{pmatrix} =
\sum_{j=1}^{L} v_j(\hat{x}, u)
\begin{pmatrix}
A_j \hat{x}(k) + B_j u(k) + r_{xj} \\
C_j \hat{x}(k) + r_{yj}
\end{pmatrix}.
\]

(5.2)

The control schemes described in this chapter are based on this model. The first technique, which is described in Section 5.1, is a master-slave technique using two networks, one for modelling/state-estimation, the other for controller implementation. Each local controller is designed on the basis of the linear quadratic (LQ) optimal design procedure. The second technique, which is described in Section 5.2, is a predictive control technique based on the model given in equation 5.2. In predictive control a quadratic objective function defined over a finite time horizon is minimised.

### 5.1 Local Linear-Quadratic Optimal Control

As discussed in previous chapter, if an integrator is included in the control loop or the offsets of the affine local models are compensated in a feedforward way then each local model can be considered as a linear model described by the discrete-time state-space model

\[
x(k + 1) = Ax(k) + Bu(k) \\
y(k) = Cx(k).
\]

(5.3) \hspace{1cm} (5.4)

The linear quadratic (LQ) optimal control problem for this 'system' with an infinite time horizon can be formulated as the minimisation of the following objective function (Anderson and Moore, 1989)

\[
J = \frac{1}{2} \sum_{k=0}^{\infty} (x^T(k)Qx(k) + u^T(k)Ru(k)).
\]

(5.5)

\(^1\)In Chapter 3, methods for identification of systems, which can be described by nonlinear observer canonical form, are treated. In this chapter a relatively general form for multi-input-multi-output (MIMO) systems is considered for controller design.
Chapter 5. Controller Design Based on State-Space Models

Matrices $Q$ and $R$ are symmetric, positive semidefinite matrices. The control law, which minimises $J$, can be given as

$$u(k) = -Kx(k),$$

(5.6)

where the optimal regulator gain matrix $K$ is calculated as

$$K = (R + B^TPB)^{-1}B^TPA.$$  

(5.7)

Matrix $P$ in the above equation is the solution of the discrete-time algebraic Riccati equation

$$Q + A^T(P - PB(R + B^TPB)^{-1}B^TP) = P.$$  

(5.8)

Note that the above design technique is based on two assumptions: the discrete-time system is time-invariant, and an infinite time horizon is considered. The controller designed this way is known as linear quadratic (LQ) optimal regulator, which derives the system state $x$ to the origin with a minimal cost $J$.

In order to make the output vector $y$ of the closed-loop system capable of tracking some reference vector $w$, the control structure have to be changed. This problem is known as tracking problem. There are different structures of optimal tracking or servo systems reported in the literature (Anderson and Moore, 1989; Unbehauen, 1995). In this section the solution with an integrated feedback of tracking error (Unbehauen, 1995) is considered.

Define the tracking error vector

$$e(k) = w(k) - y(k)$$

(5.9)

and introduce

$$p(k + 1) = p(k) + e(k)$$

(5.10)

as additional states of the system. The system with an extended state vector can be described as

$$\begin{pmatrix} x(k+1) \\ p(k+1) \end{pmatrix} = \begin{pmatrix} A & 0 \\ -C & I \end{pmatrix} \begin{pmatrix} x(k) \\ p(k) \end{pmatrix} + \begin{pmatrix} Bu(k) \\ w(k) \end{pmatrix}.$$  

(5.11)
If the LQ controller design is applied to this extended system, then in case of a tracking system, nonzero steady-state values of state vector $\mathbf{x}$ and control vector $\mathbf{u}$ will lead to an unbounded value of the objective function $J$. To avoid this problem, define the incremental extended state vector as

$$
\tilde{\mathbf{x}}(k) = \begin{pmatrix}
\mathbf{x}(k+1) - \mathbf{x}(k) \\
\mathbf{p}(k+1) - \mathbf{p}(k)
\end{pmatrix} = \begin{pmatrix}
\mathbf{x}(k+1) - \mathbf{x}(k) \\
\mathbf{e}(k)
\end{pmatrix}. \quad (5.12)
$$

Accordingly, incremental control signal can be defined as

$$
\tilde{\mathbf{u}}(k) = \mathbf{u}(k+1) - \mathbf{u}(k). \quad (5.13)
$$

The incremental extended system with $\tilde{\mathbf{x}}$ as its state vector and $\tilde{\mathbf{u}}$ as its input vector described by

$$
\tilde{\mathbf{x}}(k+1) = \tilde{\mathbf{A}}\tilde{\mathbf{x}}(k) + \tilde{\mathbf{B}}\tilde{\mathbf{u}} \quad (5.14)
$$

with

$$
\tilde{\mathbf{A}} = \begin{pmatrix}
\mathbf{A} & 0 \\
-C & \mathbf{I}
\end{pmatrix} \quad (5.15)
$$

and

$$
\tilde{\mathbf{B}} = \begin{pmatrix}
\mathbf{B} \\
\mathbf{0}
\end{pmatrix} \quad (5.16)
$$

can be used to design an LQ regulator. This regulator will drive the incremental extended system to a zero steady-state

$$
\lim_{k \to \infty} \tilde{\mathbf{u}}(k) = \mathbf{0} \quad (5.17)
$$

$$
\lim_{k \to \infty} \tilde{\mathbf{x}}(k) = \mathbf{0}. \quad (5.18)
$$

The state $\mathbf{x}$ and control $\mathbf{u}$ of the original system may attain nonzero steady-state values. As the tracking error $\mathbf{e}$ is considered as the part of the state vector $\tilde{\mathbf{x}}$ of the extended incremental system, its steady-state value becomes zero and the objective of tracking control is achieved in steady state.

In order to calculate the controller gain matrix $\mathbf{K}$ according to equation 5.7, the solution $\mathbf{P}$ of the steady-state Riccati equation 5.8 is required. There are many methods for solving this algebraic equation proposed in the literature (Unbehauen, 1995; Anderson, 1978). Some of these methods, starting with a
rough estimate of the solution, use iterative techniques to reach the correct solution (Kleinman, 1968; Hewer, 1971). The other class of algorithms may be named as one-shot algorithms. These algorithms are non-iterative and consider the solution as generalised eigenproblem (Vaughan, 1970; Arnold and Laub, 1984). Algorithms of this category possess better numerical characteristics than iterative methods. In adaptive control, the solution of the Riccati equation is required at each sampling instant. Due to the fact that the new solution $P(k)$ is not very far away from the old solution $P(k-1)$, an iterative solution may be economical, as a fairly good guess of the solution $P(k-1)$ is available for initialisation.

### 5.2 Nonlinear Model Predictive Control

In this section a state-space model predictive control scheme for nonlinear systems using LLM networks is described. This scheme is the state-space equivalent to the approach described in Section 4.3 and is based on the system described by equation 5.1, modelled as equation 5.2 minimising the objective function

$$J(k) = \frac{1}{2} \sum_{i=k+1}^{k+N} \left[ x^T(i)Qx(i) + u^*^T(i-1)Ru^*(i-1) \right]. \quad (5.19)$$

Vector $u^*$ is defined in Section 4.3. $NT$ is the length of prediction horizon, $T$ being the sampling period. Define a sequence of suggested future control signals

$$\bar{u}(k) = \begin{pmatrix} u^T(k) & \ldots & u^T(k+N-1) \end{pmatrix}^T \quad (5.20)$$

and a sequence of predicted system states over the horizon

$$\bar{x}(k) = \begin{pmatrix} \hat{x}^T(k+1) & \ldots & \hat{x}^T(k+N) \end{pmatrix}^T \quad (5.21)$$

The state predictions $\hat{x}(l+1)$ in above sequence are estimated by

$$\hat{x}(l+1) = \hat{f}(\hat{x}(l),u(l)) \quad \text{(for } l = k \ldots k+N-1). \quad (5.22)$$

As $\hat{f}$ is a local linear model approximation of $f$, for each instant $l$ in the prediction horizon, in addition to the prediction of $\hat{x}(l+1)$, the following
linearised parameters are also available

\[
\begin{align*}
\hat{A}(l) &= \left. \frac{\partial \hat{f}}{\partial \hat{x}^T} \right|_{\hat{x}(l), u(l)} \\
\hat{B}(l) &= \left. \frac{\partial \hat{f}}{\partial u^T} \right|_{\hat{x}(l), u(l)}.
\end{align*}
\]

(5.23) (5.24)

Considering the sequence vectors defined in equations 5.20 and 5.21, the objective function can be redefined as

\[
J(k) = \frac{1}{2} [\bar{x}^T(k)Q\bar{x}(k) + \bar{u}^*^T(k)\bar{R}\bar{u}^*(k)],
\]

(5.25)

where

\[
\bar{Q} = I_N \otimes Q \quad \bar{R} = I_N \otimes R.
\]

(5.26)

The calculation of gradients for the minimisation of this function similar to the method described in Section 4.3 requires

\[
\frac{\partial \bar{x}^T(k)}{\partial \bar{u}(k)} = G^T(k).
\]

(5.27)

The calculation of \(G(k)\) can be performed using \(\hat{A}(l)\) and \(\hat{B}(l)\) according to the following schema:

for \(i = 1 : N\),

\[
G_{ii}(k) = \hat{B}(k+i)
\]

for \(j = 1 : i - 1\),

\[
G_{ij}(k) = \hat{A}(k+i)G_{i-1,j}(k)
\]

end

end

This scheme represents a predictive regulator for nonlinear systems. If the objective of control is that the system output vector \(y\) follows the reference vector \(w\), then the objective function have to be altered. One possibility is to define

\[
J(k) = \frac{1}{2} \sum_{i=k+1}^{k+N} \left[ x_e^T(i)Qx_e(i) + u^T(i-1)Ru^*(i-1) \right]
\]

(5.28)
with
\[ x_e(i) = w(i) - \hat{h}(\hat{x}(i)). \]  
(5.29)

The Jacobian matrix \( G^{(e)}(k) \) required for above objective function can be determined as

for \( i = 1 : N \),
\[ G_{ii}(k) = \hat{C}(k + i)\hat{B}(k + i) \]
\[ G_{ii}^{(e)}(k) = \hat{C}(k + i)G_{ii}(k) \]
for \( j = 1 : i - 1 \),
\[ G_{ij}(k) = \hat{A}(k + i)G_{i-1,j}(k) \]
\[ G_{ij}^{(e)}(k) = \hat{C}(k + i)G_{ij}(k) \]
end
end,

where
\[ \hat{C}(l) = \frac{\partial \hat{h}}{\partial \hat{x}^T} \bigg|_{\hat{x}(l), u(l)} \]  
(5.30)

are the parameters delivered by an LLM network used for the approximation of the nonlinear vector function \( h \). The objective function of equation 5.28 is equivalent to the objective function for input-output model based predictive control and imposes only \( m \) constraints on \( x \). The only differences between this scheme and that proposed in Section 4.3 are the method of calculating gradients and the model used for the prediction. Anderson and Moore (1989) describe some other forms of objective functions, which put further constraints on \( x \), which can also be applied within this general framework.

### 5.3 Chapter Summary and Discussion

In this chapter two control schemes based on local linear state-space models are proposed. The first scheme, which is described in Section 5.1, designs local LQ optimal controllers on the basis of the parameters delivered by local linear state-space models in the network. The suppositions - time-invariant linear models and infinite time horizon - made during the design procedure do not
remain valid if the operating point is moving fast. In such a situation this scheme faces transition problems like other master-slave techniques.

The second approach discussed in Section 5.2 is local linear state-space model-based predictive control scheme for nonlinear systems. This scheme minimises a quadratic objective function defined over a finite horizon. The local interpretability of LLM network parameters as ‘parameters of the linearised system’ is exploited by this scheme for the calculation of gradients of the objective function, which are required for nonlinear optimisation. This calculation of gradients is computationally less expensive than calculation methods, which utilise numerical approximations of derivatives.
Chapter 6

Simulation and Experimental Studies

In previous chapters different schemes for the identification and control of nonlinear systems using local linear model (LLM) networks are described. In this chapter these schemes are tested in simulation studies and laboratory experiments on real plants.

Section 6.1 documents the results of simulation studies. The emphasis of these studies is on the demonstration of certain characteristics of the proposed algorithms and on the comparison of various approaches. In order not to confuse the objectives with other side-effects the examples selected for these purposes are simple. The results of laboratory experiments on real plants are presented in Section 6.2. These experiments aim at demonstrating the practical application of the proposed methods. The plants selected for these experiments demonstrate strongly nonlinear behaviours. A hydraulic position system is selected to test the model reference adaptive control scheme. Major source of nonlinearity in this system is the dead zone characteristic of the actuator valve. The nonlinear behaviour of the pH neutralisation plant selected to test state-space identification and control schemes is characterised mainly by the titration curve. The chapter closes with discussion of results and comparison of proposed schemes.
6.1 Simulation Studies

This section is divided into four sub-sections. The first sub-section deals with the state-space identification techniques presented in Chapter 3. In Sub-section 6.1.2 the properties of the model reference controller described in Section 4.1 are studied. The application of adaptive feedforward cancellation (AFC) of measurable disturbances is presented in Sub-section 6.1.3. The state feedback controllers developed in Chapter 5 are tested with the help of a simulation example are shown in Sub-section 6.1.4.

6.1.1 Identification of Local Linear State-Space Models

As mentioned above, this sub-section presents simulation results of local linear state-space identification. Two simulation examples are considered in this sub-section. The first example is a nonlinear continuous-time system. The second example describes a nonlinear discrete-time system. It is supposed that the structure of all LLM networks used to model these systems is known, while the unknown parameters of these networks have to be estimated. Only global learning techniques are applied here, as the local learning techniques proved to be unsuccessful in estimating the parameters of the example presented in Section 3.5.

Example 1: A Continuous-Time System

The nonlinear continuous-time system to be identified is described by the equations

\[
\begin{pmatrix}
\dot{x}_1(t) \\
\dot{x}_2(t)
\end{pmatrix} = \begin{pmatrix}
x_2(t) \\
0
\end{pmatrix} + \sum_{i=1}^{2} \begin{pmatrix}
ap_1 \\
ap_2
\end{pmatrix}_i x_1(t) + \begin{pmatrix}
b_1 \\
b_2
\end{pmatrix}_i u(t) + \begin{pmatrix}
r_1 \\
r_2
\end{pmatrix}_i v_i(x_1) \tag{6.1}
\]

and

\[
y(t) = x_1(t), \tag{6.2}
\]
where $x_1$ and $x_2$ represent the states, and $u$ and $y$ stand for the input and output signals of the system respectively. The numerical values of the parameters are given in Table 6.1.

**Table 6.1:** Numerical values of the parameters of example 1

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$r_1$</th>
<th>$r_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>-3</td>
<td>-2</td>
</tr>
</tbody>
</table>

Figure 6.1 shows a graphical representation of functions $v_1(x_1)$ and $v_2(x_1)$, which are described by the following expressions

$$v_1(x_1) = \left(1 + \left|\frac{x_1 + 1}{2}\right|^8\right)^{-1}$$

$$v_2(x_1) = \left(1 + \left|\frac{x_1 - 3}{2}\right|^8\right)^{-1}.$$  \hspace{1cm} (6.3)

![Graphical representation of functions $v_1(x_1)$ and $v_2(x_1)$](image)

**Figure 6.1:** Validity function $v_1(x_1)$ and $v_2(x_1)$

It is supposed that the structure of the LLM network, which can approximate the behaviour of this system according to the scheme described in Section 3.1 is known. The objective is to demonstrate that the parameters estimated by global learning techniques proposed in Section 3.4 converge to their real values. The estimation results by applying these learning techniques in delta-operator model setup and continuous-discrete model setup are drawn in
Figures 6.2 and 6.3 respectively. A signal with uniformly distributed random amplitude with a pulse width of one sampling period was used as input signal to the system. Simulation was carried out with a sampling period of 0.01.

As discussed in Section 3.4, the estimation of all offset parameters in a network is neither possible nor necessary. In case of a global learning procedure and for observer canonical form, the estimated values of the offset terms $r_1, \ldots, r_{n-1}$ of one local linear model can be anchored to zero, independent of their real values. This way the offset values $r_1, \ldots, r_{n-1}$ of all other $L - 1$ local models can be estimated. The last term $r_n$ in the offset vectors of all models is always identifiable. In this example the parameter $r_1$ of local model 1 is fixed at zero and is excluded from the set of parameters to be estimated. The other parameter estimates for local model 1 are drawn in the upper diagrams of both figures. The lower diagrams of these figures show the parameter estimates for local model 2.

A look at all these four diagrams shows that all parameter estimates converge (close) to their real values. The behaviour of estimates delivered by continuous-discrete setup is smoother than by delta-operator model setup. Another difference is the steady-state error. The delta-operator model parameters do not converge to their real values, but are very close to them. This deviation is due to the approximation of the differential operator $\frac{d}{dt}$ by the delta ($\delta$) operator and is dependent on the sampling period. The merits of the continuous-time setup are, on other hand, coupled with high computational cost required for the inter-sample integration of states and sensitivity matrices.
Figure 6.2: Parameter estimation by the global learning method in delta-operator model setup (dashed lines: actual values; solid lines: estimates)
**Figure 6.3:** Parameter estimation by the global learning method in continuous-discrete setup (dashed lines: actual values; solid lines: estimates)
Chapter 6. Simulation and Experimental Studies

Example 2: A Discrete-Time System

This simulation example is used to demonstrate the parameter estimation for discrete-time local linear state-space models. The nonlinear system to be identified is described by

\[
\begin{pmatrix}
  x_1(k + 1) \\
  x_2(k + 1)
\end{pmatrix}
= \begin{pmatrix}
  x_2(k) \\
  0
\end{pmatrix}
+ \sum_{i=1}^{2} \left[ -\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} x_1(k) + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u(k) + \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \right] v_i(x_1) \tag{6.4}
\]

and

\[ y(k) = x_1(k). \tag{6.5} \]

The numerical values of the parameters of above system are given in Table 6.2. The validity functions \( v_1 \) and \( v_2 \) are described by equation 6.3 given in Example 1. Again it is supposed that the structure of the LLM network used for the identification of this system is known. The unknown parameters of the local models in the LLM network are estimated using the discrete-time global learning algorithm given in Table 3.3. The estimation results are drawn in Figure 6.4. It is clear from this figure that after a training of the network with about 1700 samples of data, its parameters are converged to their real values.

*Table 6.2: Numerical values of the parameters of example 2*

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( r_1 )</th>
<th>( r_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i = 1 )</td>
<td>-1</td>
<td>0.21</td>
<td>0.7</td>
<td>-0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( i = 2 )</td>
<td>-0.8</td>
<td>0.12</td>
<td>1</td>
<td>-0.5</td>
<td>0.2</td>
<td>-0.09</td>
</tr>
</tbody>
</table>
**Figure 6.4:** Parameter estimation of a discrete-time system by the global learning method (dashed lines: actual values; solid lines: estimates)
6.1.2 Model Reference Control

In this section the model reference control scheme, which was proposed in Section 4.1 is tested with the help of a simulation example. The system to be controlled is described by the nonlinear difference equation

\[ y(k) = \frac{1}{2}(f(u(k - 1) + f(y(k - 1))), \]  

(6.6)

where the nonlinear function \( f \) is defined as

\[ f(x) = \begin{cases} 
3x & \text{if } |x| < 0.25, \\
\frac{x + 2\text{sgn}(x)}{3} & \text{otherwise.}
\end{cases} \]  

(6.7)

The system behaviour can be approximated by nine local linear models with sharp model switching. The operating regions of the system are shown in Figure 6.5. In order to address local models two index variables \( i \) and \( j \) are defined. The objective of this simulation is not to demonstrate the parameter estimation but to investigate the characteristics of the model reference control in master-slave configuration. For this purpose sharp switching of local models is selected such that the transition problems may be demonstrated. An LLM network with known structure is used to identify the system. Before using it in a control scheme, the LLM network is trained. In the first experiment the LLM network-based model reference controller, as derived in Section 4.1. is used to control the system. The z-transform of the reference model with output \( y_m \) to be followed by the system is given as

\[ Y_m(z) = \frac{0.21z^{-1}}{1 - z^{-1} + 0.21z^{-2}}W(z). \]  

(6.8)

The control results are drawn in Figure 6.6. Although the model network and the controller network are fully trained, the control performance (Figure 6.6) is not acceptable. As long as the operating point remains in a local model, the system output follows the reference model. But, when the operating point moves from one model to the other, large transients are observed in the system output. Figure 6.7 demonstrate the information about model switching. Any jump in the variable \( i \) or \( j \) shows a transition from one local model to the other.
A simultaneous consideration of Figures 6.6 and 6.7 reveals that switching between models has direct effect on the control performance. In order to lessen these transients, the modifications proposed in Section 4.4 are applied to this controller. The control results with this modified controller are drawn in Figure 6.8 and demonstrate that these modifications have considerably improved the performance of the master-slave control schemes.
Figure 6.6: Master-slave model reference control of the example system ($y_m$: output of the reference model; $y$: system output; $u$: control signal)

Figure 6.7: Indices $(i, j)$ of the local models switched during model reference control
Figure 6.8: Modified master-slave model reference control of the example system ($y_m$: output of the reference model; $y$: system output)

### 6.1.3 Adaptive Feedforward Cancellation

This scheme is proposed in Section 4.2. The simulation example taken to test this scheme is a multiple-input-single-output system described by the nonlinear difference equation

$$y(k) = y_u(k) + y_z(k),$$  \hspace{1cm} (6.9)

where

$$y_z(k) = \frac{1}{4}(\tanh(2.5z(k-1) + \sin(\pi y_z(k-1)))$$ \hspace{1cm} (6.10)

and

$$y_u(k) = 0.7f(u(k-1)) + 0.3f(y_u(k-1))$$ \hspace{1cm} (6.11)

with

$$f(x) = \begin{cases} 
2x & \text{if } |x| < 0.25, \\
\frac{(2x+0.5\text{sgn}(x))}{3} & \text{otherwise.}
\end{cases}$$ \hspace{1cm} (6.12)

In the above equations $u$ is the control signal, $z$ is the measurable disturbance and $y$ is the output of the system. The measurable disturbance signal $z$ is drawn in Figure 6.9.
Figure 6.9: The measurable disturbance signal $z$

An LLM network with rectangular local regions is used for the online identification of the system and a local controller network as feedback controller. The parameters of each linear component in this controller network are updated online using LQ optimisation. The control results without feedforward cancellation are drawn in Figure 6.10. Obviously, the feedback controller alone is absolutely unable to suppress the effect of the rapidly changing disturbance,

Figure 6.10: Reference and output signals $(w, y)$ without AFC
though an integrator is included in the feedback path. The feedforward cancellation is designed online using the scheme described in Section 4.2. The simulation results with the proposed adaptive feedforward cancellation (AFC) are drawn in Figure 6.11. After an adaptation phase of about 80 samples this scheme clearly suppresses the effect of measurable disturbances. In addition to the requirement of an accurate model another disadvantage of such kind of direct cancellation is the excessive perturbation in the control signal $u$, which may be a problem for some industrial actuators.

![Figure 6.11: Reference and output signals ($w,y$) in case of AFC](image)

### 6.1.4 State-Feedback Control

Two state-feedback control schemes based on LLM networks are proposed in Chapter 5. In order to test and compare them in simulations, the discrete-time system described by equations 6.4, 6.5 and 6.3 in Section 6.1.1 is taken as the plant to be controlled. An LLM-based adaptive observer with the global learning algorithm is used for the identification of the system model and for the estimation of its states. On the basis of this estimated model, both control laws - the LLM-based LQ control and the LLM-based predictive control - are calculated online. The control results of both controllers are drawn in Figures 6.12 and 6.13.
Figure 6.12: Adaptive LQ control of a nonlinear discrete-time system using state-space LLM networks ($w$: reference signal; $y$: system output; $u$: control signal)

Comparison of these results shows that in the beginning of the experiment the LQ controller performs better than the predictive controller. But with the passage of time, as the network estimates become more accurate the performance of the predictive controller improves. The predictive controller is more sensitive to the model inaccuracies, and for a good performance it calls for a good model. The LQ controller, on the contrary, is relatively robust against model inaccuracies.
Figure 6.13: Adaptive predictive control of a nonlinear discrete-time system using state-space LLM networks ($w$: reference signal; $y$: system output; $u$: control signal)

Though the system described by equations 6.4, 6.5 and 6.3 has two operating regions with different dynamics, but this difference is not too large to disturb a master-slave controller. In order to demonstrate the effect of master-slave transients, another simulation experiment is performed. The steady-state gain of the local model 2 is multiplied by 4, so that the ratio of steady state-gains of local models becomes large. In this case, the identification is switched off and the model is considered to be fully known to the controller algorithms. The system states are estimated by using fixed (non-adaptive) LLM observers.
The results of both control schemes are drawn in Figures 6.14 and 6.15. In this situation, where the LLM-based predictive controller provides excellent results, the LLM-based LQ control scheme absolutely fails on transition points, though the model is fully known.

**Figure 6.14:** LQ control of a nonlinear discrete-time system with large parameter differences using state-space LLM networks ($w$: reference signal; $y$: system output)

**Figure 6.15:** Predictive control of a nonlinear discrete-time system with large parameter differences using state-space LLM networks ($w$: reference signal; $y$: system output)
6.2 Laboratory Experiments

This section deals with the application of proposed algorithms to real plants. Two laboratory-scale plants are chosen for this purpose, which demonstrate strongly nonlinear behaviour. The behaviour of the first plant, a hydraulic positioning system, is characterised by the presence of a dead zone in the actuator characteristics, direction-dependent driving force and a variable damping. The model reference adaptive control scheme using LLM networks, which is described in Section 4.1, is applied to control this system. The second plant is a laboratory-scale pH neutralisation plant with nonlinear dynamics characterised by the titration curve. This plant is used to test the state-space identification algorithms presented in Chapter 3. The fact that the steady-state gain and time constants of this plant are strongly dependent on the operating point, its control is a difficult problem. It is argued in previous chapters that nonlinear predictive control schemes are suitable for strongly nonlinear systems. The LLM network-based predictive control schemes described in Sections 4.3 and 5.2 are applied to control this system. The results are compared with the LLM-based LQ control scheme.

6.2.1 Adaptive Control of a Hydraulic Positioning System

The hydraulic positioning system consists of three major parts, an actuator, a hydraulic differential cylinder and a spring-mass-damper assembly. A simplified schematic diagram of the whole arrangement is shown in Figure 6.16 (Du, 1989). The working cylinder moves a mass $M$ riding a saddle, which is tied rigidly to its piston. The mass $M$ experiences an opposite force $F_R$ caused by a spring and a damper. The control input $u$ of the system is the driving current of the actuator, whose magnitude determines the position of the slider in the valve, which in its turn is responsible for the pressure difference in the chambers of the working cylinder isolated from each other by the piston gasket. The position $y$ of the saddle, which is the output of the system, changes according to the pressure difference $p_1 - p_2$. A pressure limiting valve is used to keep the system pressure $p_0$ at a constant value. The cylinder possesses
a single sided piston, which causes different magnitudes of impact areas ($A_1$ and $A_2$) in the two working directions. The force and the positioning speed are, therefore, also dependent on the working direction. The actuator may either be a 4/3-way proportional valve or a servo valve. Two of the four ports of the valve ($A$ and $B$) are connected to the hydraulic cylinder, while the other two are connected to a constant pressure source ($p_0$) and an oil collection vessel ($T$) terminating the return path, respectively. This plant shows a nonlinear behaviour with an unstable inverse. Major sources of nonlinearities are the direction dependent damping of the damper as well as the dead zone in the transfer characteristics of the 4/3-way proportional valve (Junge, 1999). Coulomb and viscous frictions, small hysteresis in valves and oil leakage are other sources of nonlinearity.

![Diagram](image)

**Figure 6.16:** Simplified schematic diagram of the hydraulic positioning system

The model reference adaptive control scheme based on LLM networks as proposed in Chapter 4 is applied to the plant for both valve configurations.
The signals are sampled with a period of 100ms. A first-order linear system is taken as the reference model to be followed by the plant. An LLM network with rectangular operating regions is selected for the online identification of the plant. The order of the local models is set to 3. The network is initialised with a single model. The signal $u(k-1)$ and $y(k-1)$ are included in the scheduling vector $\zeta$. The ONALAL learning technique (Ali, 1998; Junge, 1999) is applied to tune the structure of the network and the parameters of the local models.

The control results are drawn in Figures 6.17 and 6.18. It is clear from Figure 6.17 that, in case of a servo valve, after a tuning period of 20 seconds, this control scheme forces the plant to follow the reference model.

![Graph](image)

**Figure 6.17:** LLM network-based model reference adaptive control of the hydraulic positioning system with servo valve. $y_m$: output of the reference model; $y$: controlled position; $u$: control signal.
The control performance for the system with 4/3-way proportional valve drawn in Figure 6.18 is not as good as with servo valve. This is due to the presence of a large dead-zone nonlinearity of this valve. In such cases, where the system behaviour is characterised by a strong input nonlinearity, the control schemes based on the direct compensation of this nonlinearity (Knöhl, 2001) achieve much better results than the schemes, which consider general nonlinear models.

![Graph (a)](image1)

![Graph (b)](image2)

**Figure 6.18**: LLM network-based model reference adaptive control of the hydraulic positioning system with 4/3-way proportional valve. (a) First 100 seconds. (b) after the networks are trained for 400 seconds. $y_m$: output of the reference model; $y$: controlled position
6.2.2 Identification and Control of a pH-Normalisation Plant

The schematic diagram of the laboratory-scale pH neutralisation plant is drawn in Figure 6.19. This plant consists of a five litre tank, which is continuously stirred by an electric stirrer. The tank has three inlets, each of which is connected to a reciprocating pump. These pumps can work with a maximum frequency of 180 cycles/min, and can pump a maximum of 23 litres of fluid per hour. The stroke volume of each pump can be adjusted manually. The pump frequency is dependent on the input voltage, which can be selected in an interval of [-10V, 10V]. This way the whole range of input signal is quantised into 180 steps. In order to measure the pH value of the fluid flowing out of the tank, a pH sensor is placed near to the outlet of the tank. The reliable measuring range of this sensor is between a pH value of 4 to 12.

![Schematic diagram of the pH neutralisation plant](image)

**Figure 6.19**: Schematic diagram of the pH neutralisation plant

In the following experiments two inlets of the plant are used to pump a 0.1 normal solution of acetic acid (CH₃COOH) at a fixed rate of 7.56 l/h. The third inlet is connected to inject a 0.3 normal solution of sodium hydroxide (NaOH). The flow rate of this alkaline solution is varied by the control signal $u$. The stroke volume of this pump is adjusted so that the control signal ranging
from -10V to 10V causes a flow in a range of 0-4 l/h. This plant demonstrates a strongly nonlinear behaviour. The major source of nonlinearity is the titration curve shown in Figure 6.20. Some modelling approaches (e.g. Knohl, 2001; Gerksic and Juricic, 1999) reported in the literature use a Wiener model to describe the behaviour of this plant. This model considers a static nonlinearity at the output of the model. The rest of the plant dynamics are approximated by a linear model. According to the author’s opinion this approach is not justified. Of course, the major source of nonlinearity is the titration curve. The experimental studies have shown that not only the system gain but also some other parameters, like time constants and damping ratio etc., are dependent on the pH value. The system behaviour is very fast and under-damped (very close to the stability limit) at pH values between 7 and 8, whereas at pH extremities, the system seems to be over-damped and has large transmission delays. In addition to the operating-point-dependent dynamics, amplitude quantisation of input signal makes this system really hard for identification and control. There is no analytical model so far, which can describe the dynamics of this system in the whole operating range with a reasonable accuracy.
Identification

For the identification of this plant in continuous-time state-space representation, a second-order nonlinear observer canonical form introduced in Section 3.1 is selected. The estimated state \( \hat{x}_1 \), which represents the output of the model, i.e. the estimated pH value, is selected as a scheduling variable. An LLM network with 6 local models is used as an adaptive observer according to Figure 3.1. The validity functions of these six local models are distributed over

![Validity functions for the LLM network used for the identification of the pH neutralisation plant](image)

**Figure 6.21**: Validity functions for the LLM network used for the identification of the pH neutralisation plant

the concerning range of operation as shown in Figure 6.21. This network is trained online for 5000 seconds with a sampling period of 1 sec\(^1\). The training data, i.e. input and output signals of the system, are drawn in Figure 6.22. The global recursive prediction error method given in Table 3.3 is applied to estimate the parameters of the LLM network in a delta-operator model setup. After training, the behaviour of the LLM network is validated. The validation results are drawn in Figures 6.23(a) and 6.23(b). For validation purposes, the experiment is started in observer mode. After a time \( t_s \), when the model output \( \hat{y} \) follows the system output \( y \), the observer feedback is switched off. After the time instant \( t_s \), the LLM network is running parallel to the real system. Figures 6.23(a) and 6.23(b) show that after switching the observer feedback off, the trajectory predicted by the LLM network follows (or remains close to)

\(^1\)Although the system dynamics are slow such that a sampling period between 4 to 10 seconds can be selected, but for the estimation of continuous-time models, short sampling periods are advantageous.
Figure 6.22: Input and output signals of the plant used for identification

the model output for a finite time of about 50 to 100 seconds. It diverges away slowly from the system trajectory. Such a model is useful for finite-time predictions as required by predictive control schemes. It is not appropriate for long-term forecasting.

In order to estimate a pure continuous-time model of the system, the same training data are used to train an LLM network in a continuous-discrete model setup using the algorithm given in Table 3.4. The same network structure is used in this case. Although, in the training phase the estimation error is small enough, this model fails to follow system trajectory in the validation experiment. A network with a single scheduling dimension, the pH value, and having six local models with validity functions shown in Figure 6.21 can not approximate continuous-time model of the system. In another try the input signal $u$ is also included in the scheduling vector. The operating regions of all the local models are split on the $u = 0$ plane to give 12 two-dimensional
local regions. In this structure different local models are activated for \( u < 0 \) and \( u > 0 \). The new network is trained with the same training data. The validation results for this network are drawn in Figures 6.23(c) and 6.23(d). A comparison of validation results for both models shows that both models are able to predict the system output reasonably over a finite time horizon.

![Diagrams](image)

**Figure 6.23:** Validation of pH neutralisation plant models: Figures (a) and (b) show validation results for the delta-operator model on two different trajectories, Figures (c) and (d) show validation results of the continuous-time model.
Adaptive Control

Like identification, the control of this plant is also hard due to strongly operating-point-dependent dynamics. This property of the plant makes it a good candidate for testing nonlinear control techniques. In this work, the following control techniques are applied to control the pH value of the fluid flowing out of the tank:

- LLM network-based predictive control in input-output representation (described in Section 4.3)

- LLM network-based predictive control in state-space representation (described in Section 5.2)

- LLM network-based local LQ-control in state-space representation (described in Section 5.1)

For the sake of comparison all the experiments are started with same configurations of the LLM network. The network described in previous section with six local models is selected to identify the system in discrete-time state-space or input-output representation. All the local models are initialised with rough estimates of the system parameters before starting the experiment with a sampling period of 8 secs. The control results are drawn in Figures 6.24, 6.25 and 6.26.

From these control results it can be concluded that for systems with strong nonlinearities, nonlinear predictive control techniques perform better than the master-slave techniques like local LQ controllers. The performance of the predictive controller in input-output representation seems to be better than its state-space counterpart. The reason for this fact is the efficiency of the identification algorithm. In the identification process of input-output models, the measured system signals are used as regressors in order to estimate the parameter. In state-space identification, these regressors are estimated by using sensitivity models of the system. The initial behaviour of adaptive state-space control techniques is generally inferior to that of adaptive techniques in input-output representation.
Figure 6.24: LLM network-based predictive control in input-output representation (w: reference signal; y: system output)

LLM network-based predictive control techniques successfully follow the abruptly changing reference signal. Whereas, local LQ control fails to handle such abrupt changes in the reference signal. In order to alleviate this problem, the reference signal is smoothed by a low-pass filter to generate a command signal for the local LQ controller network.
Figure 6.25: LLM network-based predictive control in state-space representation ($w$: reference signal; $y$: system output)

On a change in reference signal, the predictive control schemes have a clear strategy for the calculation of future values of control signal $u$. The local LQ scheme must oscillate the control signal to and fro to bring the system output at the desired value. This phenomenon is a typical characteristic of master-slave schemes, when control strategy devised at time instant $k$ becomes invalid at time instant $k + 1$ due to large discrepancies in the dynamics of models active at instants $k$ and $k + 1$. 
Figure 6.26: LLM network-based local LQ-control in state-space representation (w: reference signal; y: system output)

6.3 Chapter Summary and Discussion

In this chapter identification and control algorithms developed in previous chapters have been applied to various simulation examples (Section 6.1) and laboratory plants (Section 6.2). Sub-section 6.1.1 deals with simulation results concerning identification of two nonlinear systems in discrete as well as continuous-time state-space description. The example systems are simple and it is supposed that the structure of the LLM network required for the identification of these systems is known. The simulation results show that the parameter estimates delivered by the learning algorithms presented in Tables 3.3 and 3.4 converge to their real values. Continuous-discrete identification setup delivers better estimates than the delta-operator model-based identification. But this
improvement is coupled with a higher computational cost. The online identification of a real pH neutralisation plant is discussed in Sub-section 6.2.2. It is supposed that the neutralisation plant can be modelled in a continuous-time nonlinear observer canonical form. The LLM network-based models estimated by applying the global estimation algorithms are tested in so-called validation experiments. The validation results show that these models can be used for a good prediction of the system behaviour over a finite time horizon. They are not capable of a long-term (e.g. infinite horizon) forecasting of the system behaviour.

In order to test the proposed control schemes various simulation examples are used. Sub-section 6.1.2 deals with the properties of the LLM-based model reference control scheme. This control scheme is a typical example of the master-slave controller-scheduling approaches. With the help of a simulation example it is discussed, which kind of problems such a control scheme may face and how these problems are alleviated. This control scheme with improvements proposed in Section 4.4 is applied to control a hydraulic positioning system. This system demonstrates a nonlinear behaviour with unstable inverse. Major sources of nonlinearities are the direction-dependent damping and the dead zone characteristics of the actuator. The results are presented in Sub-section 6.2.1. Adaptive feedforward cancellation (AFC) for a class of nonlinear systems proposed in Section 4.2 is tested using a simulation example. The results discussed in Sub-section 6.1.3 show that if a high-frequency disturbance to a system is measurable then it is advantageous to use a direct cancellation of this disturbance. The disadvantage of such a scheme is the excessive perturbation in the control signal. Simulation results for the LLM-based adaptive state feedback control schemes are presented in Sub-section 6.1.4. Both control schemes - the master-slave LQ control and the predictive control - perform well as long as the difference in the dynamics of different local models is not large or the operating point does not jump abruptly. But if the difference is large or the operating point moves rapidly, then the master-slave scheme fails to deliver acceptable control results. The performance of the predictive control scheme, however, is not affected by these facts. The experiments performed to control the pH neutralisation plant (Sub-section 6.2.2) strengthen
this statement. This plant has strongly operating-point-dependent dynamics. The master-slave LQ controller applied to this plant has no chance to control it, when the command input jumps from a less sensitive operating region to a more sensitive one. If the command input is filtered by a low-pass filter then the master-slave LQ controller is able to control the plant, but the quality is far worse than that of the LLM-based predictive controller.
Chapter 7

Conclusions

Local model networks have become popular in the control community during last few years. The idea behind these networks is to divide the operating range of a nonlinear system in many small operating regions so that a relatively simple model can be determined for each of these regions, which can approximate the system behaviour with a sufficient accuracy. Most networks of this class use locally valid linear models and are known as LLM networks. These networks have been applied to identify and control nonlinear systems. Most of the identification techniques developed so far are limited to nonlinear systems in input-output description. The existing control schemes based on the LLM networks can be categorised under master-slave strategies. Methods for the identification of local linear state-space models along with new LLM-based master-slave and non-master-slave control schemes for nonlinear systems are developed in this dissertation. The proposed identification methods and control schemes are investigated in simulation studies and experiments on real plants.

In Chapter 3, different techniques for parameter estimation of local linear state-space models for continuous as well as discrete-time systems are proposed, and corresponding learning algorithms are developed. These techniques can be divided into two classes: local learning techniques and global learning techniques. Local learning algorithms determine the most active local linear
model at current operating point and update its parameters neglecting the non-
linearity caused by model switching and overlapping effects. These algorithms
are based on the well-known state-variable-filters method and the parameter
estimation is performed by using recursive least squares (RLS) or recursive pre-
diction error (RPE) algorithms. Global learning algorithms presented in this
chapter minimise a global cost function in order to estimate the parameters of
the whole blended model simultaneously using RPE method. With the help
of simulation studies, it is shown that the local learning techniques, which are
reported to be preferable for the estimation of local linear input-output models
(Murray-Smith and Johansen, 1997), fail to converge in the case of state-space
models.

Five new adaptive control schemes on the basis of LLM networks are de-
veloped in Chapters 4 and 5. Model reference adaptive control (MRAC),
adaptive feedforward cancellation (AFC) and LLM-based predictive control
schemes proposed in Chapter 4 are based on input-output models of the sys-
tem. In MRAC a network of local linear controllers is designed for a nonlinear
system such that the closed-loop system behaves like a given reference model.
The AFC scheme realises an adaptive feedforward compensation of measurable
disturbances for a class of nonlinear systems using LLM networks. LLM-based
predictive control scheme optimises a future control sequence in order to min-
imise an objective function. The primary requirements of a predictive control
scheme are a good prediction of the future trajectory of the system and the cal-
culation of gradients of the objective function subject to the proposed control
sequence. An LLM network as a nonlinear model of the system fulfills both
requirements. In addition to delivering a good prediction of system behaviour,
this network also provides a set of parameters of the system linearised at each
operating point on the predicted trajectory. These parameters are used to
calculate the gradients of the objective function directly rather than approxi-
mating them with other numerically intensive differentiation techniques. The
state-feedback control schemes described in Chapter 5 include LLM-based LQ
control and predictive control.

Model reference adaptive control (MRAC), adaptive feedforward cancel-
lation (AFC) and LLM-based LQ control schemes fall into the category of
master-slave schemes. These schemes have advantages and disadvantages. They use two networks, one as identification model and the other as controller. There is a one-to-one correspondence between the components of both networks. The advantage of these schemes is that the parameters of each local controller component can be determined individually by applying linear design techniques. The major drawbacks of these schemes become obvious when they are put in practice to control strongly nonlinear systems with fast moving operating points. In such cases undesired transients are observed even with soft transitions between local controllers.

Results of various simulation studies and laboratory experiments documented in Chapter 6 show satisfactory performance of the proposed identification algorithms. Parameters estimated by global learning algorithms converge to their real values, and the experimental validation of the models of the real plants is successful. In the case of identification of continuous-time local linear state-space models, the parameter estimates delivered by continuous-discrete identification setup are more accurate than those estimated for delta-operator models. But the former is computationally more expensive than the latter. The master-slave control schemes perform well as long as the operating point is moving slowly or the differences in the dynamics of local models are not large. These schemes face transition problems with fast moving operating point. The predictive control schemes based on LLM networks do not suffer from such transition problems. The reason is that, in predictive control, future control strategy is devised keeping these inter-model transitions in mind. The basic requirement for the success of predictive control is that all the local models, which are used in a prediction horizon, must deliver local estimates of system parameters within some error bounds.

Master-slave control configuration is an outcome of the desire to utilise linear design techniques in nonlinear control. In case of a linear controller design a transfer function or a time-invariant state-space model of the system is used. The major assumption for the design is that the system is fully described by the given model and its parameters are not going to change in the future. This assumption leads to determine an infinite horizon controller, which has fixed parameters. The desired control performance is guaranteed as long as the as-
assumption remains valid. In a master-slave scheme for nonlinear systems this assumption remains no longer valid if the operating point is moving among local models and the control performance is deteriorated. This deterioration is dependent, on one hand, on the discrepancies between the dynamics of the local models, which successively become active, and on the other hand, on the rate of change of the operating point. LLM-based predictive control schemes perform better than master-slave schemes, but are computationally more expensive. Furthermore, the predictive control schemes are more sensitive to model uncertainties and require a good model.

During the design of LLM-based control schemes, the global stability of the closed-loop system has not been considered in this work. The future research may be focused on the global design procedures for the local controllers such that the stability of the closed-loop system can be guaranteed.
Zusammenfassung


Kapitel 3 befasst sich mit der Entwicklung verschiedener Parameterschätzungsverfahren für zeitkontinuierliche sowie für zeitdiskrete lokale lineare Zustandsraummodelle. Hierfür wurden entsprechende Lernalgorithmen entwickelt. Diese Verfahren lassen sich in lokale Lernverfahren und globale Lernverfahren einteilen. Lokale Lernverfahren aktualisieren die Parameter des am aktuellen Arbeitspunkt aktiven lokalen Modells und vernachlässigen dabei das durch Modellumschaltung und Überlappung verursachte nichtlineare Verhalten. Die-
ZUSAMMENFASSUNG


Eine „master-slave“-Regelungsstruktur ist aus dem Wunsch entstanden.

Bibliography


Powell, M. J. D. (1985). Radial basis functions for multivariable interpolation: 
A review. In: *IMA Conference on Algorithms for the Approximation of 


Rajbman, N. S., A. A. Dorofeyuk and A. D. Kasavin (1981). Identification of 
nonlinear process by piecewise approximation. In: *Trends and Progress in 


Gaussian networks. In: *Proceedings of IFAC Symposium on Intelligent 
Components and Instruments for Control Applications*. Malaga, Spain. 
pp. 301 – 305.

to Modelling and Control* (T. A. Johansen and R. Murray-Smith, Eds.). 


Lebenslauf

Persönliche Daten
Name                Abid Ali
geboren am          25.07.1967 in Haroonabad, Pakistan
Familienstand       verheiratet

Schulausbildung
03.1972 - 02.1977   Govt. Primary School Haroonabad
03.1977 - 03.1982   Govt. High School Haroonabad
04.1982 - 04.1984   Govt. R. I. College Haroonabad

Hochschulstudium
05.1985 - 01.1991   Studium der Elektrotechnik an der
                    University of Engineering and Technology
                    Lahore (Pakistan)
                    Abschluss: B.Sc.(Electrical Engineering.)
10.1994 - 02.1998   Studium der Elektrotechnik an der
                    Ruhr-Universität Bochum
                    Abschluss: Diplom

Berufserfahrung
1991 - 1992         Field Service Engineer bei Siemens Pakistan
                    Ltd, Lahore.
1992 - 1993         Junior Engineer bei Pakistan Water And
                    Power Development Authority (WAPDA),
                    Lahore
Seit März 1998      wissenschaftlicher Mitarbeiter am Lehrstuhl
                    für Automatisierungstechnik und Prozess-
                    informatik, Ruhr-Universität Bochum

Bochum, im Mai 2003