Engineering Applications of Artificial Intelligence 50 (2016) 320–330

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Automated generation of feedforward control using feedback linearization of local model networks

Nikolaus Euler-Rolle [a,*], Igor Škrjanc [b], Christoph Hametner [a], Stefan Jakubek [a]

[a] Christian Doppler Laboratory for Model Based Calibration Methodologies, Institute of Mechanics and Mechatronics, Vienna University of Technology, Getreidemarkt 9/E325/A5, 1060 Vienna, Austria

[b] Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 9 November 2015
Received in revised form
22 January 2016
Accepted 28 January 2016

Keywords:
Feedforward control
Feedback linearization
Local model networks
System inversion
Internal dynamics

ABSTRACT

An effective but yet simple approach is introduced to automatically attain a dynamic feedforward control law for non-linear dynamic systems represented by discrete-time local model networks (LMN). In this context, feedback linearization is applied to the generic model structure of LMN and the resulting input transformation is used as model inverse. This general and automated approach for model inversion is applicable even when the overall model complexity may be high. Thus, by representing a non-linear dynamic system by an LMN and applying the proposed feedforward control law generation, a dynamic feedforward control for such a non-linear system can be found automatically with the knowledge of measured input–output data only. However, when feedback linearization is considered, the stability of the internal dynamics plays a key role. This paper analyses the occurring internal dynamics for LMN, which directly result from the chosen model structure in identification, and discusses the effects on the transformed system. Finally, the effectiveness of the proposed data-driven feedforward control is demonstrated by a simulation example as well as by an actual application to the pre-distortion of a microelectromechanical systems (MEMS) loudspeaker with electrostatic actuation.

1. Introduction

The automatic generation of models from measured input and output data is nowadays an established approach in many engineering disciplines (e.g. Sjöberg et al., 1995; Murray-Smith and Johansen, 1997; Norgaard et al., 2000; Nelles, 2001; Ljung, 2010). Commonly, such models are used to simulate the real process for various purposes, such as managing complex traffic networks (McKenney and White, 2013), optimum control of cogeneration heat and power plants (Cerri et al., 2006) or model predictive control in general (Townsend and Irwin, 2001), to name a few. In recent years, significant research efforts have been made to also exploit the structure of non-linear dynamic models in order to facilitate the design of control systems (Hametner et al., 2014; Gao et al., 2002; Deng et al., 2008).

When control tasks are considered, non-linear model structures such as local model networks (LMN) can also be used to determine control laws and their parameters (e.g. Hametner et al., 2013; Hafner et al., 2000). In general, LMN are a well-established multiple-model approach for data-driven modelling of non-linear systems (e.g. Gregorčič and Lightbody, 2007, 2010; Hametner and Jakubek, 2011; Nelles, 2001). This model architecture interpolates between different local models, each valid in a certain operating regime which offers a versatile structure for the identification of non-linear dynamic systems. Each operating regime represents a simple model, e.g. a linear regression model (Murray-Smith and Johansen, 1997), whose parameters are found by identification. Although the complexity of LMN increases with the amount of local linear models to form a sophisticated non-linear model, the model structure still remains generic. This fact is beneficially exploited when automatically generating a dynamic feedforward control law for arbitrarily complex LMN.

To obtain such a dynamic feedforward control law, usually some kind of model inversion has to be performed. Inspired by Silverman (1969), who investigated invertibility for time varying linear systems, Hirschorn (1979) extended the basic principles of system inversion to non-linear systems. Despite numerous research efforts (e.g. Isidori and Byrnes, 1990; Devasia et al., 1996), system inversion still remains a challenging task, which in general requires a thorough analysis and knowledge of the non-linear system under consideration.

However, in the presented approach, by considering the generic model structure of LMN, the application of feedback linearization automatically leads to an output–input relation, which is

* Corresponding author.
E-mail address: nikolaus.euler-rolle@tuwien.ac.at (N. Euler-Rolle).

http://dx.doi.org/10.1016/j.engappai.2016.01.039
0952-1976/

suitable as dynamic feedforward control law for the underlying non-linear process. Thus, a model inverse can directly be found from measured input–output data only.

Originally the concept of feedback linearization has been introduced by Byrnes and Isidori (1984) for the first time. Basically, a non-linear system is linearized exactly by using a non-linear coordinate transformation such that the resulting transformed system consists of an input transformation, linear external dynamics and unobservable internal dynamics. The latter represent a non-linear equivalent to the notion of transmission zeros in linear system theory. A historical perspective of this wide field as well as a detailed review of the feedback linearization technique is given by Isidori (1995) or Slotine and Li (1991). Feedback linearization for discrete-time systems, as it is necessary with LMN, has been addressed for example by Lee et al. (1987), Monaco and Normand-Cyrot (1987) or Grizzle (1986). For continuous-time systems, feedback control using neural networks in combination with feedback linearization approaches has already been applied (He et al., 1998; Chien et al., 2008).

Both, LMN (Murray-Smith and Johansen, 1997; Norgaard et al., 2000; Nelles, 2001; Maass et al., 2009; Novak and Bobal, 2009; Hametner et al., 2014) and the concept of feedback linearization are by themselves well established concepts in academia as well as in industry (e.g. Kotman et al., 2010; Moulin and Chauvin, 2011; Nielsen et al., 2010; Tuan et al., 2013). However, combining both ideas offers the opportunity to provide a substantial tool to dynamically feedforward control any arbitrary non-linear process with knowledge of measured input–output data only. The approach taken in this paper supersedes the need for an in-depth knowledge of the underlying non-linear process as the generic model structure of LMN allows for an automated generation of a feedforward control law. Besides introducing the concept of automatic generation of feedforward control laws, this paper also examines those pitfalls, which are associated with the method, namely the stability of the internal dynamics, respectively, the zero dynamics. According to Isidori (2013), at least "*systems in which the zero dynamics are unstable are still a substantially unexplored and open area of research*". Therefore, in the present contribution an analysis of the more general internal dynamics (as compared to zero dynamics or a minimum phase property) for LMN is given. In addition, an overview of how to choose the architecture of the LMN in order to obtain a model with full relative degree, which is preferable for feedforward control, is given.

Typically, feedforward control is used as an enhancement of common feedback control strategies. In Fig. 1 a so-called two-degrees-of-freedom control scheme is depicted where the design of the feedforward part $\hat{\Sigma}_{\mathrm{ol}}^{-1}$ and the feedback part Σ_{C} is independent of each other. If parameter uncertainties or model errors occur in $\hat{\Sigma}_{\mathrm{ol}}$ and $\hat{\Sigma}_{\mathrm{ol}}^{-1}$, respectively (e.g. due to measurement noise on the identification data), the feedback part will still track the desired trajectory and try to compensate for the inaccuracies. However, only stable plants should be considered in a control scheme including a feedforward part.

Feedforward control of LMN has been considered in the literature before. Karer et al. (2011) applied feedforward control to a dynamic hybrid fuzzy model of a batch reactor with both discrete and continuous states. Therein the partitioning considers the output only and the validity functions are triangular. In the present contribution also the input can be used as a dimension of the partition space, which is an important prerequisite for the partitioning of many non-linearities where off-equilibrium conditions arise (Johansen et al., 2000). In addition, a hierarchical discriminant tree, which is determined from input–output data only (Hametner and Jakubek, 2011; Jakubek and Hametner, 2009), yields the validity functions instead of utilizing fuzzy rules. Nentwig and Mercorelli (2008) proposed an algorithm for a combined analytical/numerical inversion of a static fuzzy neural network applied to a throttle valve control. In contrast, the presented approach in this paper also holds for dynamic LMN and in addition directly incorporates non-linear validity functions of arbitrary shape into the automatic feedforward control law generation. Hagan et al. (2002) propose NARMA-L2 control, which incorporates an approximation of a non-linear autoregressive-moving average (NARMA) model found by non-linear identification. The resulting NARMA-L2 model contains two separate subnetworks such that the next controller input $u(k)$ is not contained inside the non-linearity and can therefore be used to solve for a reference tracking control input. However, a specific model structure is required in the identification. Boukezzoula et al. (2003, 2007) analytically invert a Takagi–Sugeno fuzzy model by feedback linearization for designing a fuzzy controller, although the submodels are inverted locally and an additional criterion has to be considered in each time step to choose from multiple solutions.

In a direct data-driven design approach such as direct inverse control (e.g. Norgaard et al., 2000; Hunt et al., 1992) depicted in Fig. 2(b), the inverse model is identified from input–output data directly. In contrast, in the presented approach outlined in Fig. 2 (a), the inverse is found from an existing plant model. The latter procedure offers the opportunity to exploit the existing and generic model structure of local model networks. By evaluating the relative degree and the resulting internal dynamics, this approach allows a far deeper insight and a methodology to analyse and understand the resulting feedforward control law. In addition, no dedicated identification procedure or special model structure is required.

Numerous applications in various branches of the industry benefit from the presented approach as merely adequately measured input–output data are required to identify a model (i.e. an LMN) of almost any arbitrary non-linear dynamic process. To automatically obtain a feedforward control law for such a process, the LMN is represented in discrete-time state space form, which is then transformed into a feedback linearized normal representation. To determine the required feedforward input value for the desired reference trajectory, an input transformation is utilized. Therein the current and past model outputs are replaced by the desired reference values. Besides an illustrative example, an application of the presented approach, a microelectromechanical

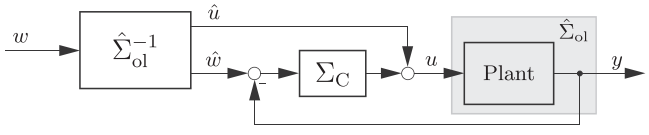

Fig. 1. Two-degrees-of-freedom control scheme.

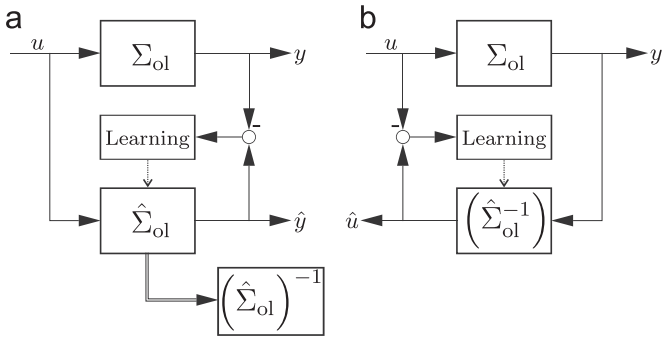

Fig. 2. Comparison of (a) the feedforward control law generation using feedback linearization of a LMN plant model and (b) a direct data-driven control approach.

systems (MEMS) loudspeaker with electrostatic actuation, is considered. In a laboratory setup, data-driven feedforward control is applied for pre-distortion. Experimental validation shows that the non-linear characteristic of the MEMS loudspeaker can be linearized successfully using the presented approach.

Subsequently, in Section 2 the LMN and its state space representation is presented. An overview of feedback linearization in general and the description of the feedforward control law generation is given in Section 3. In Section 4 the resulting internal dynamics are analysed and in Section 5 an illustrative example as well as the experimental application is shown.

2. Local model networks

To apply the feedback linearization technique, a dynamic single input, single output local model network (LMN; Nelles, 2001; Gregorčič and Lightbody, 2007, 2010; Hametner and Jakubek, 2011) is considered. Due to the application of feedforward control, it is assumed to be stable (Hametner et al., 2014). The LMN is given in a (non-minimum realization) state space representation of the form

$$\boldsymbol{x}_i(k+1) = \boldsymbol{A}_i\boldsymbol{x}(k) + \boldsymbol{B}_i u(k) + \boldsymbol{f}_i \tag{1a}$$

$$\boldsymbol{x}(k+1) = \sum_{i\in\mathcal{I}} \Phi_i(\tilde{\boldsymbol{x}}(k))\boldsymbol{x}_i(k+1) \tag{1b}$$

$$\hat{y}(k) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}(k), \tag{1c}$$

For the indices i of the local models an ordered set is defined as

$$\mathcal{I} = (i \in \mathbb{N} \mid 1 \le i \le I) \tag{2}$$

where I denotes the user prescribed number of local linear models. The validity functions $\Phi_i(k)$ represent a weighting of the local state vectors and will be explained later in this section. Further, the architecture of a dynamic local model network in state space representation is depicted in Fig. 3, where q^{-1} represents the time shift operator. A detailed description of the state space notation (1) can be found in Hametner et al. (2013). Subsequently, only a short summary of notation is given. Note that the incorporation of an affine term $\boldsymbol{f}_i$ is necessary to obtain an arbitrary close approximation of the linear time-varying dynamic system, which results from dynamic linearization about trajectories of general non-linear systems (Johansen et al., 2000).

Basically, the state vector definition can be chosen arbitrarily. Although, by choosing the state vector $\boldsymbol{x}(k) \in \mathbb{R}^{(M-1+N)\times 1}$ as

$$\boldsymbol{x}(k) = \begin{bmatrix} u(k-M+1) \\ \vdots \\ u(k-1) \\ \hdashline \hat{y}(k-N+1) \\ \vdots \\ \hat{y}(k) \end{bmatrix}, \tag{3}$$

the same state vector definition holds for each local model. Thus, the treatment and assessment of equations is simplified considerably. In addition it is assumed that all local models possess the same relative degree δ_{local} according to the well-known result for linear systems

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{A}_i^j\boldsymbol{B}_i = 0 \quad \forall j < \delta_{\mathrm{local}} - 1, \quad i \in \mathcal{I} \tag{4a}$$

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{A}_i^{\delta_{\mathrm{local}}-1}\boldsymbol{B}_i \neq 0 \quad \forall i \in \mathcal{I}. \tag{4b}$$

This is fulfilled by the local system matrices $\boldsymbol{A}_i \in \mathbb{R}^{(M-1+N)\times(M-1+N)}$, which contain time shifts of the input (above the dashed line) as well as those of the output

$$\boldsymbol{A}_i = \begin{bmatrix} \underline{\boldsymbol{I}}_{M-1} & \boldsymbol{0}_{M-1\times N} \\ \hdashline \boldsymbol{0}_{N-1\times M-1} & \overline{\boldsymbol{I}}_N \\ \boldsymbol{b}^{(i)\mathrm{T}} & \boldsymbol{a}^{(i)\mathrm{T}} \end{bmatrix}, \tag{5}$$

where abbreviations are defined as

$$\overline{\boldsymbol{I}}_j = \begin{bmatrix} \boldsymbol{0}_{j-1\times 1} & \boldsymbol{I}_{j-1} \end{bmatrix}, \quad \overline{\boldsymbol{I}}_j \in \mathbb{R}^{(j-1)\times j} \tag{6}$$

$$\underline{\boldsymbol{I}}_j = \begin{bmatrix} \begin{bmatrix} \boldsymbol{0}_{j-1\times 1} & \boldsymbol{I}_{j-1} \end{bmatrix} \\ \boldsymbol{0}_{1\times j} \end{bmatrix}, \quad \underline{\boldsymbol{I}}_j \in \mathbb{R}^{j\times j} \tag{7}$$

for an arbitrary size j with $\boldsymbol{I}$ denoting the identity matrix. The last row of (5) contains the parameters $\boldsymbol{b}^{(i)\mathrm{T}} \in \mathbb{R}^{1\times(M-1)}$ of the input and of the autoregressive part $\boldsymbol{a}^{(i)\mathrm{T}} \in \mathbb{R}^{1\times N}$ in the form

$$\boldsymbol{b}^{(i)\mathrm{T}} = \begin{bmatrix} b_M^{(i)} & \cdots & b_3^{(i)} & b_2^{(i)} \end{bmatrix} \tag{8}$$

$$\boldsymbol{a}^{(i)\mathrm{T}} = \begin{bmatrix} a_N^{(i)} & \cdots & a_1^{(i)} \end{bmatrix}. \tag{9}$$

The input vector $\boldsymbol{B}_i \in \mathbb{R}^{(M-1+N)\times 1}$ is defined as

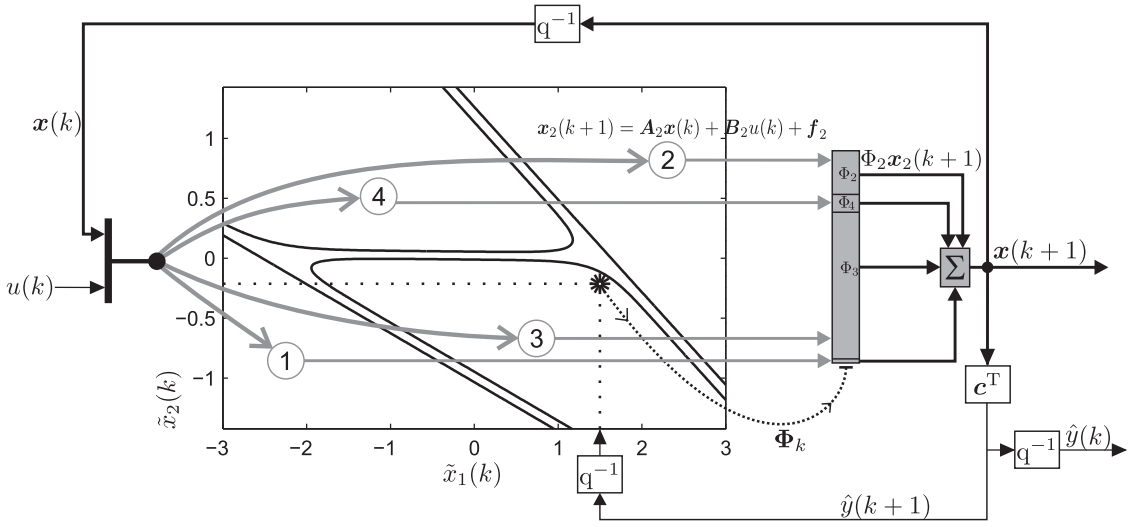

Fig. 3. Architecture of a local model network.

$$\boldsymbol{B}_i = \begin{bmatrix} \mathbf{0}_{M-2\times 1} \\ 1 \\ \hline \mathbf{0}_{N-1\times 1} \\ b_1^{(i)} \end{bmatrix}, \tag{10}$$

and the term $\boldsymbol{f}_i \in \mathbb{R}^{(M-1+N)\times 1}$ in (1) introduces a local affine term

$$\boldsymbol{f}_i = \begin{bmatrix} \mathbf{0}_{M-2+N\times 1} \\ c^{(i)} \end{bmatrix}. \tag{11}$$

The output matrix $\boldsymbol{c}^{\mathrm{T}} \in \mathbb{R}^{1\times(M-1+N)}$ is constant:

$$\boldsymbol{c}^{\mathrm{T}} = \begin{bmatrix} \mathbf{0}_{1\times M-2+N} & 1 \end{bmatrix}. \tag{12}$$

However, this formulation is only one way amongst many others to represent an LMN. Exactly the same input–output behaviour, as it results from (1), can also be described by notation as difference equation

$$\hat{y}(k) = \sum_{i\in\mathcal{I}} \Phi_i(\tilde{\boldsymbol{r}}(k)) \left[\sum_{m\in\mathcal{M}} b_m^{(i)} u(k)\mathrm{q}^{-m} - \sum_{n\in\mathcal{N}} a_n^{(i)} \hat{y}(k)\mathrm{q}^{-n} + c^{(i)} \right], \tag{13}$$

which is especially useful in the identification procedure. Each local parameter in (8)–(11) is related to (13) through the sets of the orders $\mathcal{M}$ of the used time delays of the input and of the feedback output orders $\mathcal{N}$. Thus, the individual entries b_m with index $\{m\in\mathbb{N} \mid 2\le m\le M\}$ in (8), b_1 in (10) and a_n with index $\{n\in\mathbb{N} \mid 1\le n\le N\}$ in (9) are found according to

$$b_m = \begin{cases} b_m^{(i)} & m\in\mathcal{M} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$a_n = \begin{cases} a_n^{(i)} & n\in\mathcal{N} \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Therefore, the resulting state space system order relates to the set according to $N=\max(\mathcal{N})$ and the maximum time shift of the input to $M=\max(\mathcal{M})$. The number of elements of the sets $\mathcal{M}$ and $\mathcal{N}$ is referred to as $|\mathcal{M}|$ and $|\mathcal{N}|$, respectively.

The validity functions $\Phi_i(\tilde{\boldsymbol{r}}(k))$ in (13) and $\Phi_i(\tilde{\boldsymbol{x}}(k))$ in (1) govern the weighted aggregation of the local model outputs. They are constrained to form a partition of unity

$$\sum_{i\in\mathcal{I}} \Phi_i = 1 \tag{16}$$

$$0\le \Phi_i \le 1 \quad \forall i\in\mathcal{I}. \tag{17}$$

The appropriate selection of the shape of the validity functions and its parametrization depends on the problem under investigation. In the application example in Section 5, continuously differentiable sigmoid functions are used as validity functions. They are found by optimization and a hierarchical discriminant tree (Hametner and Jakubek, 2011; Jakubek and Hametner, 2009). However, the applicability of the presented feedforward control is not influenced by the shape of the validity functions. Therefore, concerning the identification task and related topics reference may be given to the literature (Sjöberg et al., 1995; Murray-Smith and Johansen, 1997; Norgaard et al., 2000; Nelles, 2001).

The input vector $\tilde{\boldsymbol{r}}(k)$ of the validity functions $\Phi_i(\tilde{\boldsymbol{r}}(k))$, which spans the so-called partition space, can be chosen differently from the sets $\mathcal{M}$ and $\mathcal{N}$

$$\tilde{\boldsymbol{r}}(k) = \left[\boldsymbol{u}(k-\tilde{\mathcal{M}}) \;\; \hat{\boldsymbol{y}}(k-\tilde{\mathcal{N}}) \right], \quad \tilde{\boldsymbol{r}}(k)\in\mathbb{R}^{1\times\tilde{o}}, \tag{18}$$

although the sets $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{N}}$ are usually subsets of $\mathcal{M}$ and $\mathcal{N}$. They are also user defined. Note that due to the time shifted evaluation of the update equation for $\boldsymbol{x}_i(k+1)$ in (1) as compared to $\hat{y}(k)$ in (13) the validity functions in state space notation are determined using $\tilde{\boldsymbol{x}}(k)=\tilde{\boldsymbol{r}}(k+1)$.

Usually, the partition space is spanned by input and output variables or by output variables only. Choosing either the input or output as the only variable for the partition space would be sufficient if only equilibrium models were considered. However, when strong transient operating conditions occur, the respective second quantity has to be added to the partition space in order to distinguish between near-equilibrium conditions and their off-equilibrium counterparts (Johansen et al., 2000).

When the input appears in $\tilde{\boldsymbol{r}}(k)$ it is reasonable to choose the lowest order of the input used in the partition space equally to the lowest input order in the regressor (i.e. $\min(\tilde{\mathcal{M}})=\min(\mathcal{M})$). Otherwise the system dynamics would change even before the input variable takes effect as a system input.

Additionally, an excitation by the local affine term in combination with the validity functions would be possible, which acts as an additional input $\sum_{i\in\mathcal{I}}\Phi_i(\tilde{\boldsymbol{r}}(k))c^{(i)}$ containing a lower order of the input in $\tilde{\boldsymbol{r}}(k)$ than in the regressor itself.

When the state vectors are blended as described above, (1) can be rewritten as

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(\boldsymbol{\Phi}_k)\boldsymbol{x}(k) + \boldsymbol{B}(\boldsymbol{\Phi}_k)u(k) + \boldsymbol{f}(\boldsymbol{\Phi}_k) \tag{19a}$$

$$\hat{y}(k) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}(k), \tag{19b}$$

where the appearing matrices are found by weighted aggregation from

$$\boldsymbol{A}(\boldsymbol{\Phi}_k) = \sum_{i\in\mathcal{I}} \Phi_i(\tilde{\boldsymbol{x}}(k))\boldsymbol{A}_i \tag{20}$$

$$\boldsymbol{B}(\boldsymbol{\Phi}_k) = \sum_{i\in\mathcal{I}} \Phi_i(\tilde{\boldsymbol{x}}(k))\boldsymbol{B}_i \tag{21}$$

$$\boldsymbol{f}(\boldsymbol{\Phi}_k) = \sum_{i\in\mathcal{I}} \Phi_i(\tilde{\boldsymbol{x}}(k))\boldsymbol{f}_i, \tag{22}$$

using the notation of the membership functions $\boldsymbol{\Phi}_k\in\mathbb{R}^{I\times 1}$ in vector form

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(\tilde{\boldsymbol{x}}(k)) = \begin{bmatrix} \Phi_1(\tilde{\boldsymbol{x}}(k)) \\ \vdots \\ \Phi_i(\tilde{\boldsymbol{x}}(k)) \\ \vdots \\ \Phi_I(\tilde{\boldsymbol{x}}(k)) \end{bmatrix}. \tag{23}$$

When LMN are considered, in general two ways of interpolating between the local models can be pursued: either a weighted aggregation of the local model outputs or a blended combination of model parameters (Gregorčič and Lightbody, 2008). Using output blending, the global output of the model is determined as a linear combination of the local model outputs. Thus, the individual models could also be represented by completely different types of model structures. The drawbacks of output blending are a reduced transparency of the overall model and the requirement that each local model must be stable. As an alternative approach, the parameters of the local models can be blended, if all of them comprise the same structure. Using this approach, the transparency is increased as the overall model structure remains the same as that of the local models. Due to the incorporation of off-equilibrium local models, whose equilibrium points do not lie in the region of their validity (Johansen et al., 2000), the parameter blended LMN is not necessarily unstable if unstable off-equilibrium local models are utilized (Hametner et al., 2014).

3. Feedforward control

The automatic generation of a dynamic feedforward control law is accomplished by feedback linearizing the LMN, which can

be readily achieved due to its generic model structure. In the sequel the feedback linearization technique for discrete-time systems is shortly reviewed in general (adapted from Henson and Seborg, 1997), which is then applied to the non-linear structure of LMN in Section 3.2.

3.1. Feedback linearization of discrete-time systems

The input–output linearization problem for a general discrete-time non-linear non-affine system

$$\boldsymbol{x}(k+1) = \boldsymbol{F}(\boldsymbol{x}(k), u(k)) \tag{24a}$$

$$y(k) = h(\boldsymbol{x}(k)) \tag{24b}$$

is considered (Lee et al., 1987; Monaco and Normand-Cyrot, 1987; Grizzle, 1986). The composition of the scalar function $h(\boldsymbol{x}) : \mathbb{R}^{d\times 1} \to \mathbb{R}$ and the vector function $\boldsymbol{F}(\boldsymbol{x}) : \mathbb{R}^{d\times 1} \to \mathbb{R}^{d\times 1}$, with d describing the system dimension, is defined as $h\circ\boldsymbol{F}(\boldsymbol{x}) = h(\boldsymbol{F}(\boldsymbol{x}))$. Higher order compositions are defined recursively: $h\circ\boldsymbol{F}^j(\boldsymbol{x}) = h\circ\boldsymbol{F}^{j-1}(\boldsymbol{F}(\boldsymbol{x}))$, where $h\circ\boldsymbol{F}^0(\boldsymbol{x}) = h(\boldsymbol{x})$. The composition operator plays the same role as does the Lie derivative in the continuous-time case.

The discrete-time system (24) is said to have relative degree δ at the point $(\boldsymbol{x}_0, u_0)$ if

$$\frac{\partial}{\partial u(k)}\Big(h\circ\boldsymbol{F}^j(\boldsymbol{x}(k), u(k))\Big) = 0 \quad \forall j \le \delta - 1 \tag{25}$$

for all $(\boldsymbol{x}, u)$ in a neighborhood of $(\boldsymbol{x}_0, u_0)$ and

$$\frac{\partial}{\partial u(k)}\Big(h\circ\boldsymbol{F}^\delta(\boldsymbol{x}_0, u_0)\Big) \neq 0. \tag{26}$$

Thus, by definition of the relative degree all compositions fulfilling $j \le \delta - 1$ are independent of the current input $u(k)$ and can be written as

$$h\circ\boldsymbol{F}^j(\boldsymbol{x}(k), u(k)) = h\circ\boldsymbol{F}_0^j(\boldsymbol{x}(k)), \quad 1 \le j \le \delta - 1. \tag{27}$$

To represent the system (24) in normal form, a diffeomorphism $[\boldsymbol{\xi}^{\mathrm{T}}(k), \boldsymbol{\eta}^{\mathrm{T}}(k)]^{\mathrm{T}} = \boldsymbol{\Gamma}(\boldsymbol{x}(k))$ defining the new coordinates $\boldsymbol{\xi}(\cdot)$ and $\boldsymbol{\eta}(\cdot)$ is constructed as follows. The $\boldsymbol{\xi}(\cdot)$ coordinates are chosen as

$$\xi_j(k) = h\circ\boldsymbol{F}_0^{j-1}(\boldsymbol{x}(k)), \quad 1 \le j \le \delta. \tag{28}$$

The remaining $d-\delta$ states

$$\eta_j(k) = \Gamma_{\delta+j}(\boldsymbol{x}(k)), \quad 1 \le j \le d - \delta \tag{29}$$

can be chosen arbitrarily such that $\boldsymbol{\Gamma}$ is invertible and

$$\frac{\partial}{\partial u(k)}\big(\Gamma_{\delta+j}\circ\boldsymbol{F}(\boldsymbol{x}(k), u(k))\big) = 0, \quad 1 \le j \le d - \delta \tag{30}$$

holds. As a result, the system in normal form is

$$\begin{aligned}
&\xi_1(k+1) = \xi_2(k)\\
&\xi_2(k+1) = \xi_3(k)\\
&\qquad\vdots\\
&\xi_\delta(k+1) = h\circ\boldsymbol{F}^\delta(\boldsymbol{\Gamma}^{-1}(\boldsymbol{\xi}(k), \boldsymbol{\eta}(k)), u(k)) = v(k)\\
&\boldsymbol{\eta}(k+1) = \boldsymbol{q}(\boldsymbol{\xi}(k), \boldsymbol{\eta}(k))\\
&y(k) = \xi_1(k),
\end{aligned} \tag{31}$$

where $\boldsymbol{q}(\boldsymbol{\xi}(k), \boldsymbol{\eta}(k))$ represents the unobservable $(d-\delta)$-dimensional internal dynamics

$$q_j(\boldsymbol{\xi}(k), \boldsymbol{\eta}(k)) = \Gamma_{\delta+j}\circ\boldsymbol{F}(\boldsymbol{\Gamma}^{-1}(\boldsymbol{\xi}(k), \boldsymbol{\eta}(k))), \quad 1 \le j \le d - \delta, \tag{32}$$

which are independent of $u(\cdot)$ due to (30). By introducing the virtual input $v(k)$, that part of system (31), which is described by the external states $\boldsymbol{\xi}(k)$, can be considered as a chain of δ time-shifts with output $y(k)$ and input $v(k)$. These external dynamics are therefore linear and its virtual input $v(k)$ is found by a non-linear algebraic equation representing the input transformation

$$v(k) = h\circ\boldsymbol{F}^\delta(\boldsymbol{x}(k), u(k)). \tag{33}$$

If the relative degree equals the system order, i.e. $\delta = d$ holds, the original system (24) is said to have full relative degree and no internal dynamics exist. Conditions for the relative degree of LMN are explicitly derived in the next section. As in the continuous-time case, for any feedforward or feedback control law the unobservable internal dynamics need to be asymptotically stable.

3.2. Feedback linearization of local model networks

For the application of feedback linearization, the LMN is considered in state space representation (1) such that the general system (24) is defined as

$$\boldsymbol{F}(\boldsymbol{x}(k), u(k)) = \boldsymbol{A}(\boldsymbol{\Phi}_k)\boldsymbol{x}(k) + \boldsymbol{B}(\boldsymbol{\Phi}_k)u(k) + \boldsymbol{f}(\boldsymbol{\Phi}_k) \tag{34a}$$

$$h(\boldsymbol{x}(k)) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}(k). \tag{34b}$$

To determine the relative degree, (25) and (26) are considered, respectively. First, by evaluating (25) for $j=1$, it is checked whether the relative degree is $\delta > 1$, which is true if

$$\frac{\partial}{\partial u_k}\Big(\Big[\boldsymbol{b}^{\mathrm{T}}(\boldsymbol{\Phi}_k)\ \ \boldsymbol{a}^{\mathrm{T}}(\boldsymbol{\Phi}_k)\Big]\boldsymbol{x}(k) + b_1(\boldsymbol{\Phi}_k)u(k) + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{\Phi}_k)\Big) = 0. \tag{35}$$

Eq. (35) holds, if $b_1(\boldsymbol{\Phi}_k) = 0$ and $\frac{\partial \boldsymbol{\Phi}_k}{\partial u_k} = 0$. The latter condition states that the partitioning at point in time k has to be independent of $u(k)$.

Subsequently, through repeated evaluation of (25) with increasing values of j, the relative degree δ is determined as the lowest integer, which fulfills (26). For an LMN this evaluation can be reformulated as two independent conditions concerning the parameters and validity functions, respectively:

$$b_j(\boldsymbol{\Phi}_k) = 0, \quad \forall \boldsymbol{\Phi}_k \wedge \forall j < \delta \tag{36}$$

$$\frac{\partial \boldsymbol{\Phi}_{k+j}}{\partial u(k)} = 0, \quad 0 \le j \le \delta - 2. \tag{37}$$

Thus, the relative degree δ is determined primarily by the index of the first non-zero numerator parameter b_j (i.e. $b_j = 0,\ \forall j < \delta$) such that $\delta = \min(\mathcal{M})$ holds. In case of output partitioning only (i.e. $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(\hat{\boldsymbol{y}}(k+1-\tilde{\mathcal{N}}))$), the validity functions are independent of $u(\cdot)$ and (37) is irrelevant. When considering input partitioning, usually the partitioning is chosen such that $\tilde{\mathcal{M}} \subseteq \mathcal{M}$. Otherwise the system dynamics could change even before the input variable takes effect as a system input. Using the definitions $\tilde{\boldsymbol{x}}(k) = \tilde{\boldsymbol{r}}(k+1)$ and (18), the second condition (37) is fulfilled for

$$\delta \le \min(\tilde{\mathcal{M}}). \tag{38}$$

In general the relative degree can be formulated as

$$\delta = \min(\tilde{\mathcal{M}} \cup \mathcal{M}). \tag{39}$$

When the state transformation (28) is applied to the structure of the LMN (34) one gets

$$\begin{aligned}
\xi_1(k) &= \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}_k\\
\xi_2(k) &= \boldsymbol{c}^{\mathrm{T}}\big(\boldsymbol{A}(\boldsymbol{\Phi}_k)\boldsymbol{x}_k + \boldsymbol{f}(\boldsymbol{\Phi}_k)\big)\\
\xi_3(k) &= \boldsymbol{c}^{\mathrm{T}}\big(\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{A}(\boldsymbol{\Phi}_k)\boldsymbol{x}_k + \boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{f}(\boldsymbol{\Phi}_k) + \boldsymbol{f}(\boldsymbol{\Phi}_{k+1})\big)\\
&\ \ \vdots\\
\xi_\delta(k) &= \boldsymbol{c}^{\mathrm{T}}\big(\boldsymbol{A}(\boldsymbol{\Phi}_{k+\delta-2})\cdots\boldsymbol{A}(\boldsymbol{\Phi}_k)\boldsymbol{x}_k + \boldsymbol{A}(\boldsymbol{\Phi}_{k+\delta-2})\cdots\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{f}(\boldsymbol{\Phi}_k)\\
&\qquad + \cdots + \boldsymbol{f}(\boldsymbol{\Phi}_{k+\delta-2})\big)
\end{aligned} \tag{40}$$

and the input transformation (33) in original coordinates yields

$$\begin{aligned}
v(k) = \boldsymbol{c}^{\mathrm{T}}\big[&\boldsymbol{A}(\boldsymbol{\Phi}_{k+\delta-1})\cdots\boldsymbol{A}(\boldsymbol{\Phi}_k)\boldsymbol{x}_k + \boldsymbol{A}(\boldsymbol{\Phi}_{k+\delta-1})\cdots\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{B}(\boldsymbol{\Phi}_k)u(k)\\
&+ \boldsymbol{A}(\boldsymbol{\Phi}_{k+\delta-1})\cdots\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{f}(\boldsymbol{\Phi}_k) + \cdots + \boldsymbol{f}(\boldsymbol{\Phi}_{k+\delta-1})\big].
\end{aligned} \tag{41}$$

The remaining $d-\delta$ states (29) of the internal dynamics have to be chosen such that $\boldsymbol{\Gamma}$ is diffeomorph and the internal dynamics are independent of the input, i.e. such that (30) is satisfied. Thus, the internal dynamics can be analyzed by using the affine

diffeomorphism

$$\begin{bmatrix} \boldsymbol{\xi}(k) \\ \hdashline \boldsymbol{\eta}(k) \end{bmatrix} = \boldsymbol{\Gamma}(\boldsymbol{x}(k)) = \boldsymbol{T}(k)\boldsymbol{x}(k) + \boldsymbol{\chi}(k), \tag{42}$$

with the validity function dependent transformation matrix

$$\boldsymbol{T}(k) = \begin{bmatrix} \boldsymbol{T}_1(k) \\ \hdashline \boldsymbol{T}_2(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}^{\mathrm{T}} \\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{A}(\Phi_k) \\ \vdots \\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{A}(\Phi_{k+\delta-2})\cdots\boldsymbol{A}(\Phi_k) \\ \hdashline \boldsymbol{T}_2(k) \end{bmatrix} \tag{43}$$

and $\boldsymbol{\chi}(k)$ containing local affine terms

$$\chi(k) = \begin{bmatrix} 0 \\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}(\Phi_k) \\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{A}(\Phi_{k+1})\boldsymbol{f}(\Phi_k) + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}(\Phi_{k+1}) \\ \vdots \\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{A}(\Phi_{k+\delta-2})\cdots\boldsymbol{A}(\Phi_{k+1})\boldsymbol{f}(\Phi_k) + \cdots \\ + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}(\Phi_{k+\delta-2}) \\ \hdashline \boldsymbol{0} \end{bmatrix} \tag{44}$$

The lower part $\boldsymbol{T}_2(k)$ of the transformation matrix has to be determined such that $\boldsymbol{T}(k)$ is nonsingular. Thus, $\boldsymbol{T}_2(k)$ must be orthogonal to $\boldsymbol{T}_1(k)$. For this purpose, the kernel $\boldsymbol{N}_1^{\mathrm{T}}(k) \in \mathbb{R}^{d-\delta+1\times d}$ of $\boldsymbol{T}_1(k)$

$$\boldsymbol{N}_1^{\mathrm{T}}(k) = \ker(\boldsymbol{T}_1(k)) \tag{45}$$

is used to define

$$\boldsymbol{T}_2(k) = \boldsymbol{V}^{\mathrm{T}}(k)\boldsymbol{N}_1^{\mathrm{T}}(k) \quad \in \mathbb{R}^{d-\delta\times d} \tag{46}$$

with an arbitrary matrix $\boldsymbol{V}^{\mathrm{T}}(k) \in \mathbb{R}^{d-\delta\times d-\delta+1}$. However, when choosing $\boldsymbol{V}^{\mathrm{T}}(k)$ it has to be ensured that $\boldsymbol{T}_2(k)$ has full rank to yield a regular transformation $\boldsymbol{T}(k)$.

Further, according to (30), the input must not influence the internal dynamics directly, which means that

$$\boldsymbol{T}_2(k)\boldsymbol{B}(\boldsymbol{\Phi}_k) \stackrel{!}{=} \boldsymbol{0} \tag{47}$$

is required. Replacing $\boldsymbol{T}_2(k)$ by its definition (46) yields

$$\boldsymbol{V}^{\mathrm{T}}(k)\boldsymbol{N}_1^{\mathrm{T}}(k)\boldsymbol{B}(\boldsymbol{\Phi}_k) = \boldsymbol{0}. \tag{48}$$

From this relation, matrix $\boldsymbol{V}^{\mathrm{T}}(k)$ can be determined

$$\boldsymbol{V}^{\mathrm{T}}(k) = \ker(\boldsymbol{N}_1^{\mathrm{T}}(k)\boldsymbol{B}(\boldsymbol{\Phi}_k)), \tag{49}$$

which in combination with (46) completes the nonsingular diffeomorphism.

The transformed system can be written as:

$$\begin{bmatrix} \tilde{\boldsymbol{\xi}}(k+1) \\ \hdashline \tilde{\boldsymbol{\eta}}(k+1) \end{bmatrix} = \tilde{\boldsymbol{A}}(k) \begin{bmatrix} \tilde{\boldsymbol{\xi}}(k) \\ \hdashline \tilde{\boldsymbol{\eta}}(k) \end{bmatrix} + \tilde{\boldsymbol{B}}(k)u(k) + \tilde{\boldsymbol{f}}(k) \tag{50}$$

with

$$\begin{bmatrix} \tilde{\boldsymbol{\xi}}(k) \\ \hdashline \tilde{\boldsymbol{\eta}}(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\xi}(k) \\ \hdashline \boldsymbol{\eta}(k) \end{bmatrix} - \chi(k) \tag{51}$$

$$\tilde{\boldsymbol{A}}(k) = \begin{bmatrix} \boldsymbol{T}_1(k+1) \\ \hdashline \boldsymbol{T}_2(k+1) \end{bmatrix} \boldsymbol{A}(\Phi_k) \begin{bmatrix} \boldsymbol{T}_1(k) \\ \hdashline \boldsymbol{T}_2(k) \end{bmatrix}^{-1} \tag{52}$$

$$\tilde{\boldsymbol{B}}(k) = \begin{bmatrix} \boldsymbol{T}_1(k+1) \\ \hdashline \boldsymbol{T}_2(k+1) \end{bmatrix} \boldsymbol{B}(\Phi_k) \tag{53}$$

$$\tilde{\boldsymbol{f}}(k) = \begin{bmatrix} \boldsymbol{T}_1(k+1) \\ \hdashline \boldsymbol{T}_2(k+1) \end{bmatrix} \boldsymbol{f}(\Phi_k). \tag{54}$$

Substituting the input transformation (41) for $u(k)$ into (50) yields the system in normal form

$$\begin{bmatrix} \tilde{\boldsymbol{\xi}}(k+1) \\ \hdashline \tilde{\boldsymbol{\eta}}(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_\delta & \boldsymbol{0}_{\delta\times d-\delta} \\ \hdashline \tilde{\boldsymbol{A}}_{\xi\eta}(k) & \tilde{\boldsymbol{A}}_{\eta\eta}(k) \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\xi}}(k) \\ \hdashline \tilde{\boldsymbol{\eta}}(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{\delta-1\times 1} \\ v(k) \\ \hdashline \boldsymbol{0}_{d-\delta\times 1} \end{bmatrix} + \tilde{\boldsymbol{f}}(k). \tag{55}$$

The substitution

$$v(k) = \tilde{\boldsymbol{A}}_{(\delta)}(k) \begin{bmatrix} \tilde{\xi}(k) \\ \hdashline \tilde{\boldsymbol{\eta}}(k) \end{bmatrix} + \tilde{B}_{(\delta)}(k)u(k) + \tilde{f}_\delta(k) \tag{56}$$

represents the input transformation (41) expressed in the new coordinates. The scalar $\tilde{B}_{(\delta)}(k)$ and the matrix $\tilde{\boldsymbol{A}}_{(\delta)}(k)$ denote row number δ of the transformed input vector $\tilde{\boldsymbol{B}}(k)$ and system matrix $\tilde{\boldsymbol{A}}(k)$, respectively.

The block diagram of the resulting feedback linearized model structure with affine terms is depicted in Fig. 4. Therein, those blocks, which represent the internal dynamics, are framed by a dashed line and the input transformation is outlined by a grey background. In the normal form system (55), the system matrix ends up to have a block structure. The internal dynamics are described by the trajectory dependent right lower block matrix $\tilde{\boldsymbol{A}}_{\eta\eta}(k) \in \mathbb{R}^{d-\delta\times d-\delta}$ with the external states $\tilde{\boldsymbol{\xi}}(k)$ acting as excitation with $\tilde{\boldsymbol{A}}_{\xi\eta}(k)\tilde{\boldsymbol{\xi}}(k) \in \mathbb{R}^{d-\delta\times 1}$ as the left lower block matrix. In return, the resulting internal states $\tilde{\boldsymbol{\eta}}(k)$ are used in the input transformation to evaluate the relation between the physical input $u(k)$ and the virtual input $v(k)$ to the external dynamics. For a detailed analysis of the internal dynamics' structure and $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ see Section 4.1.

3.3. Feedforward control law generation

Due to its generic model structure, the general normal form representation (31) of an LMN is readily and automatically available by applying the system transformation described in the previous section. For such a normal system, the implementation of dynamic feedforward control is straightforward.

If the transformed input (33) is chosen such that $v(k) = w(k+\delta)$, where $w(\cdot)$ describes the desired output trajectory, exact tracking $y(k+\delta) = w(k+\delta)$ can be achieved. The physical input $u(k)$ can be found in the original coordinates by solving

$$w(k+\delta) - h \circ \boldsymbol{F}^{\delta}(\boldsymbol{w}(k), u(k)) = 0, \tag{57}$$

where $\boldsymbol{w}(\cdot)$ denotes the state vector $\boldsymbol{x}(\cdot)$ with the values of $y(\cdot)$ being replaced by the reference values $w(\cdot)$. In particular, (41) is used for an LMN. Besides the replacement of the actual output values within the state vector $\boldsymbol{x}(\cdot)$ with the reference values $\boldsymbol{w}(\cdot)$, also the partitioning is found from the reference instead of the actual output only (i.e. $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(\boldsymbol{w}(k+1-\tilde{\mathcal{N}}), \boldsymbol{u}(k+1-\tilde{\mathcal{M}}))$). Thus, the feedforward control law

$$\begin{aligned} w(k+\delta) - \boldsymbol{c}^{\mathrm{T}}\big[&\boldsymbol{A}(\boldsymbol{\Phi}_{k+\delta-1})\cdots\boldsymbol{A}(\boldsymbol{\Phi}_k)\boldsymbol{w}_k + \boldsymbol{A}(\boldsymbol{\Phi}_{k+\delta-1})\cdots\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{B}(\boldsymbol{\Phi}_k)u(k) \\ &+ \boldsymbol{A}(\boldsymbol{\Phi}_{k+\delta-1})\cdots\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{f}(\boldsymbol{\Phi}_k) + \cdots + \boldsymbol{f}(\boldsymbol{\Phi}_{k+\delta-1})\big] = 0 \end{aligned} \tag{58}$$

does not use any actual values $\hat{y}(\cdot)$ of the output.

Alternatively, the feedforward control law can be obtained by considering the input transformation in the transformed coordinates $\left[\tilde{\boldsymbol{\xi}}^{\mathrm{T}}(k)\ \tilde{\boldsymbol{\eta}}^{\mathrm{T}}(k)\right]$. Nevertheless, in either case the stability of the

internal dynamics has to be checked if they exist, i.e. if the system has less than full relative degree. Therefore a system transformation is necessary anyhow.

In the subsequent section, a system classification concerning the internal dynamics as well as its implication on the choice of model structure for the parameter estimation is given.

4. System analysis

The applicability of an LMN in feedforward control is decisively determined by the relative degree and the internal dynamics, respectively. While the stability of the internal dynamics depends on the actual values of the estimated parameters only, the relative degree is already predefined as given in (39) by the choice of the model structure before the actual parameter identification takes place. When the relative degree equals the system order, the system is said to have full relative degree. In this case there are no internal dynamics and feedback linearization is straightforward.

When the structure of an LMN is considered, in particular the choice of input partitioning as well as the orders of the input used in the regressor are relevant for the relative degree and thus for the internal dynamics. In Table 1 the implications of these choices are shown in an overview. In the simplest case, the partitioning considers the output only ($\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(\hat{\boldsymbol{y}}(k+1-\tilde{\mathcal{N}}))$, i.e. $\tilde{\mathcal{M}} = \varnothing$) and a local model structure with no numerator can be induced by the choice of the regressor using $|\mathcal{M}| = 1$. In this case, the LMN will have full relative degree and thus no internal dynamics. This also holds for input partitioning (i.e. $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(\hat{\boldsymbol{y}}(k+1-\tilde{\mathcal{N}}), u(k+1-\tilde{\mathcal{M}}))$), as long as the chosen order matches that of the regressor, i.e. $\tilde{\mathcal{M}} = \mathcal{M}$ with $|\mathcal{M}| = 1$. It is reasonable to choose the lowest order of the input used in the partition space equally to the lowest input order in the regressor

$$\min(\tilde{\mathcal{M}}) = \min(\mathcal{M}), \tag{59}$$

which is denoted as "causal choice" in Table 1. Otherwise the system dynamics would change even before the input variable takes effect as a system input. As soon as multiple input orders are used in the regressor, the stability of the internal dynamics has to be checked independent of the partitioning under consideration.

When only output partitioning is used, the current input $u(k)$ appears in the feedforward control law (58) or (41) only once and an explicit solution is possible. In case of input partitioning, particularly when (59) holds, (58) or (41), respectively, is an implicit equation. The unknown current input $u(k)$ appears within the product containing the input vector in the equation directly, as well as within the membership function $\boldsymbol{\Phi}_{k+\delta-1}$ as a partition variable. Thus, an explicit solution is not possible. The slightly more complex task of evaluating the implicit feedforward control law is achieved numerically using the Newton–Raphson method for example.

4.1. Internal dynamics

The stability assessment of the internal dynamics for non-linear systems is generally very hard to tackle even in the single input, single output case (Isidori, 2013). Although the internal dynamics of the normal form system (55), which can be rewritten as

$$\tilde{\boldsymbol{\eta}}(k+1) = \tilde{\boldsymbol{A}}_{\eta\eta}(k)\tilde{\boldsymbol{\eta}}(k) + \tilde{\boldsymbol{A}}_{\xi\eta}(k)\tilde{\boldsymbol{\xi}}(k), \tag{60}$$

have a neat structure, their behaviour depends on the desired trajectory $\boldsymbol{w}(k)$ in two ways. On the one hand, it excites the internal dynamics through the external states, which contain time shifts of the desired trajectory only, cf. e.g. (55). On the other hand, the behaviour of the internal dynamics, which is determined by $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ as the right lower block of $\tilde{\boldsymbol{A}}(k)$, varies highly non-linearly with the trajectory in the following way. The actual system matrices of the original system (34) depend on the membership functions $\boldsymbol{\Phi}_k$. In the feedback linearization, the transformation matrix $\boldsymbol{T}(k)$ and its affine term $\boldsymbol{\chi}(k)$ incorporate various membership function values $\boldsymbol{\Phi}_k$ to $\boldsymbol{\Phi}_{k+\delta-2}$, which are found by evaluation of the partitioning along the past and future desired trajectory. These values are used in the matrix $\boldsymbol{T}_1(k)$ directly as well as in the kernel approach to find $\boldsymbol{T}_2(k)$. It is the resulting transformation matrix $\boldsymbol{T}(k)$ that finally determines the transformed normal form system (55) and together with the affine term $\boldsymbol{\chi}(k)$ accounts for the sophisticated non-linear behaviour. Thus, each matrix itself depends on the trajectory $\boldsymbol{w}(k)$, whereas the resulting system matrix $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ of the internal dynamics depends on products of such matrices as well as their kernel.

Due to this highly non-linear and trajectory dependent model structure, usual stability analysis methods for the system dynamics of LMN (Hametner et al., 2014) are not applicable to the internal dynamics. To assess the global stability of an LMN, these methods try to formulate a suitable Lyapunov function, which eventually leads to linear matrix inequalities, which consider constant local system matrices only (Tanaka and Sugeno, 1992). Applying the same methodology to assess the asymptotic stability of the internal dynamics would require for example the common

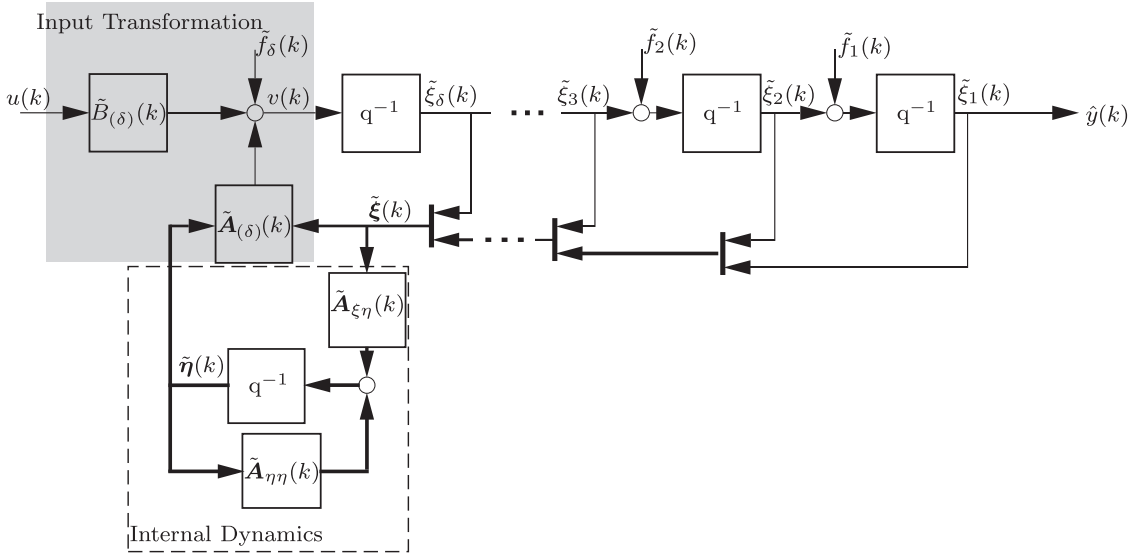

Fig. 4. Structure of a feedback linearized LMN for $\delta \geq 2$ with affine terms in the external dynamics.

Table 1
Classification of systems.

	Partitioning	
	$\tilde{\mathcal{M}}=\varnothing$	$\tilde{\mathcal{M}}\neq\varnothing$
Loc. Num. Poly.		
$\lvert\mathcal{M}\rvert=1$	No internal dynamics	No internal dynamics for causal choice of partitioning
$\lvert\mathcal{M}\rvert>1$	Check stability of internal dynamics	Check stability of internal dynamics

quadratic Lyapunov function

$$V(k)=\tilde{\boldsymbol{\eta}}^{\mathrm{T}}(k)\boldsymbol{P}\tilde{\boldsymbol{\eta}}(k), \tag{61}$$

using a constant positive definite matrix $\boldsymbol{P}$, to decrease strictly monotonically over time. However, in contrast to the stability assessment of LMN, the system matrix $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ of the internal dynamics changes with the trajectory at each point in time so that an untractable number of LMIs would result. As a physically inspired supplement, the strongly related concept of passivity theory (e.g. Slotine and Li, 1991) could also be applied to assess whether the internal dynamics represent a passive system. In that case, the stability of the internal dynamics would be guaranteed for any trajectory.

Further, the eigenvalues of the right lower block $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ can be considered at each point in time. Each eigenvalue represents the momentary location of the internal dynamics' poles. In this way, an eigenvalue scatter plot can be generated for any trajectory. It reveals if a pole lies outside of the unit circle at any point in time of the simulated trajectory.

However, for the considered system structure, stability cannot be characterized by the location of the eigenvalues of $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ alone. Considering the transformation $\tilde{\boldsymbol{\eta}}(k)=\boldsymbol{E}(k)\boldsymbol{\nu}(k)$, where $\boldsymbol{E}(k)$ is the momentary eigenvector basis of $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ and $\boldsymbol{\Lambda}(k)$ a diagonal matrix with the eigenvalues of $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ as its diagonal elements, (60) transforms to

$$\boldsymbol{\nu}(k+1)=\boldsymbol{E}^{-1}(k+1)\boldsymbol{E}(k)\boldsymbol{\Lambda}(k)\boldsymbol{\nu}(k) \tag{62}$$

with the excitation $\tilde{\boldsymbol{\xi}}(k)$ omitted. In (62) it becomes obvious that the resulting transformed system would be decoupled with its poles equal to the eigenvalues $\boldsymbol{\Lambda}(k)$ only if the product $\boldsymbol{E}^{-1}(k+1)\boldsymbol{E}(k)$ was equal to the identity matrix. In other words, the faster the eigenvector basis changes, the stronger is the coupling of the states $\boldsymbol{\nu}(k)$ as well as the deviation from the momentary eigenvalues $\boldsymbol{\Lambda}(k)$. In particular, by considering the spectral norm of the matrix product $\boldsymbol{E}^{-1}(k+1)\boldsymbol{E}(k)\boldsymbol{\Lambda}(k)$ as a measure of the dilation or contraction of $\boldsymbol{\nu}$ over time, the following inequality holds for exponential stability:

$$\|\boldsymbol{E}^{-1}(k+1)\boldsymbol{E}(k)\boldsymbol{\Lambda}(k)\|_2\leq\|\boldsymbol{E}^{-1}(k+1)\boldsymbol{E}(k)\|_2\|\boldsymbol{\Lambda}(k)\|_2<1. \tag{63}$$

This criterion in conjunction with the eigenvalues itself could serve as another starting point in the assessment of stability.

When operating points are considered, $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ as well as its eigenvector basis $\boldsymbol{E}(k)$ remains constant as the desired reference trajectory remains constant also. Thus, for assessing whether an operating point has stable internal dynamics, the consideration of the eigenvalues of $\tilde{\boldsymbol{A}}_{\eta\eta}(k)$ is sufficient. Additionally, to ensure that the system can transition from one operating point to another in a stable way, the $\boldsymbol{\eta}$-trajectory (see an exemplary phase plane plot in Fig. 5) has to remain bounded such that the resulting feedforward control input signal remains bounded, or even more restrictive adheres to input signal constraints.

Altogether, for the practical application of feedforward control in the presence of internal dynamics, only momentary stability assertions related to operating points can be made. Although the assessment of a stationary operating point is directly possible, a global stability proof is still the topic of ongoing research. When dynamic feedforward control is intended, the application to a model with full relative degree is beneficial.

5. Results

In this section an illustrative example shows state trajectories as well as the internal dynamics of an LMN with two local models. In addition, experimental results for the pre-distortion of a microelectromechanical systems (MEMS) loudspeaker demonstrate the successful application to a non-linear physical process.

5.1. Illustrative example

A synthetic LMN with two local models and a model structure using $\mathcal{M}=\{3, 4, 5\}$ and $\mathcal{N}=\{1, 2, 3\}$ is used to illustrate the described approach. The system matrix, the input vector and the local affine term, as described in Section 2, contain the following values:

$$\boldsymbol{A}_1=\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 0.09 & -0.3 & 0.5 & 0 & 0.4 & -1.3 & 1.8\end{bmatrix} \tag{64a}$$

$$\boldsymbol{A}_2=\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 0.72 & -1.44 & 0.8 & 0 & 0.45 & -1.77 & 2.3\end{bmatrix} \tag{64b}$$

$$\boldsymbol{B}_1^{\mathrm{T}}=\boldsymbol{B}_2^{\mathrm{T}}=[0\ 0\ 0\ 1\ 0\ 0\ 0] \tag{65}$$

$$\boldsymbol{f}_1^{\mathrm{T}}=[0\ 0\ 0\ 0\ 0\ 0\ 0.5] \tag{66a}$$

$$\boldsymbol{f}_2^{\mathrm{T}}=[0\ 0\ 0\ 0\ 0\ 0\ -0.5]. \tag{66b}$$

The partitioning considers the output only, thus $\tilde{\mathcal{N}}=\{1\}$ and $\tilde{\mathcal{M}}=\varnothing$ holds. In particular, in this illustrative example the validity functions have been chosen arbitrarily and parameterized manually. They are defined as

$$\Phi_1(k)=\frac{1+\tanh\left(1.5\ \hat{y}(k-1)\right)}{2} \tag{67a}$$

$$\Phi_2(k)=1-\Phi_1(k) \tag{67b}$$

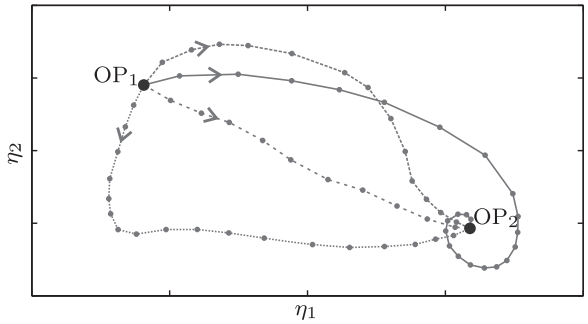

Fig. 5. Phase plane of an operating point (OP) transition with exemplary trajectories of the internal states.

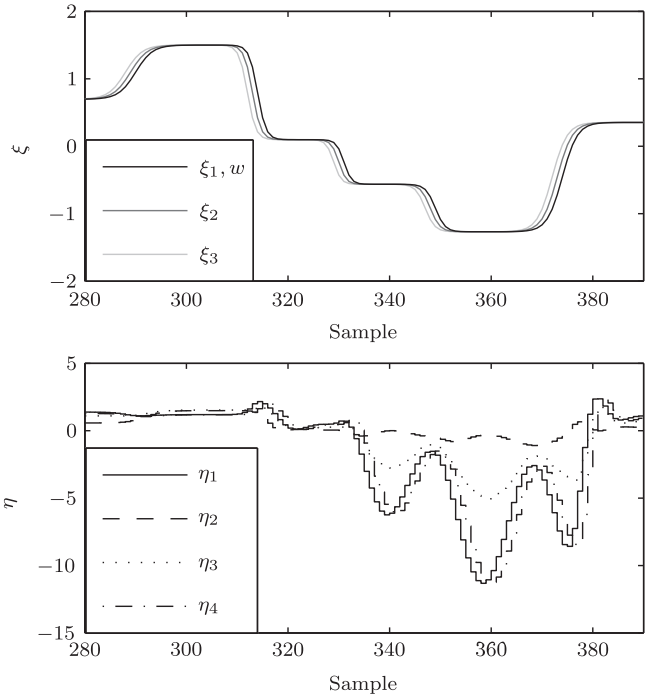

Fig. 6. Upper panel: trajectory of the external states ξ_i and lower panel: trajectory of the internal states η_i.

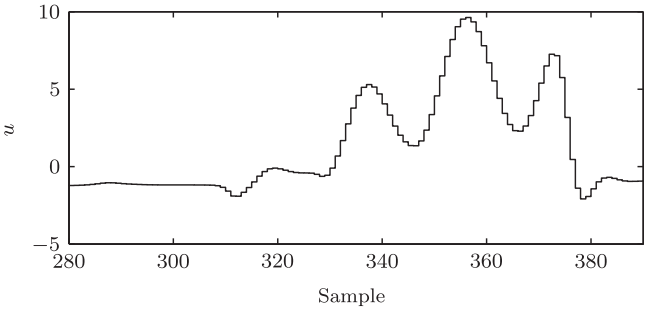

Fig. 7. Feedforward control input signal $u(k)$.

The relative degree according to (39) yields $\delta=3$ and the system order of the non-minimum realization state space model is seven, thus there are three external and four internal states in the transformed system.

Feedforward control is used to track a desired reference trajectory, which is comprised of low-pass filtered steps. In Fig. 6 the states of the transformed system are shown. In the upper panel, the external states $\boldsymbol{\xi}$ represent time shifts of the output, wherein ξ_1 is equivalent to the tracked trajectory $w(k)$. The internal states $\boldsymbol{\eta}(k)$ can be observed in the lower panel. In Fig. 7 the feedforward input signal $u(k)$ is depicted. As it is obvious from Fig. 4, the input transformation used for calculating $u(k)$ uses both, the external and the internal states. Thus, the oscillations in $\boldsymbol{\eta}(k)$, which are induced by the internal dynamics, directly influence the required input signal to track the desired trajectory. Unstable internal dynamics would therefore lead to an unbounded input signal.

5.2. Experimental results

To demonstrate the application of the proposed dynamic feedforward control using LMN, a non-linear MEMS loudspeaker with electrostatic actuation (Tumpold et al., 2014, 2015) is considered without a feedback control part. The MEMS speaker consists of two circular shaped electrodes with a two micrometer air gap between them. The top electrode represents the back-plate or stator and consists of a 600 nm polysilicon (poly)conductor coated with a 140 nm silicon nitride (SiNi) insulation layer. The bottom electrode is represented by a 330 nm polysilicon layer. Because of the high intrinsic tensile pre-stress of the back-plate of about 1 GPa compared to the intrinsic tensile pre-stress of the membrane with about 43 MPa, the back-plate can be assumed as stiff and the membrane as flexible. Non-linearities in the MEMS speaker are caused by large mechanical deformation, the multi-layered structure and the electrostatic force itself. By applying a voltage in the range from 0 to 10 V_{pp} as input signal, the membrane is displaced non-linearly and the resulting displacement (0–0.35 μm_{pp}), which is measured by a laser vibrometer, is considered as the output. Since oscillations in nanometer scale are recorded, the vibrometer sensor head and the MEMS speaker have been arranged on an actively air damped vibration insulated table, which is located on a decoupled foundation (Tumpold et al., 2015). With this setup, low frequency vibrations from the environment could have been avoided.

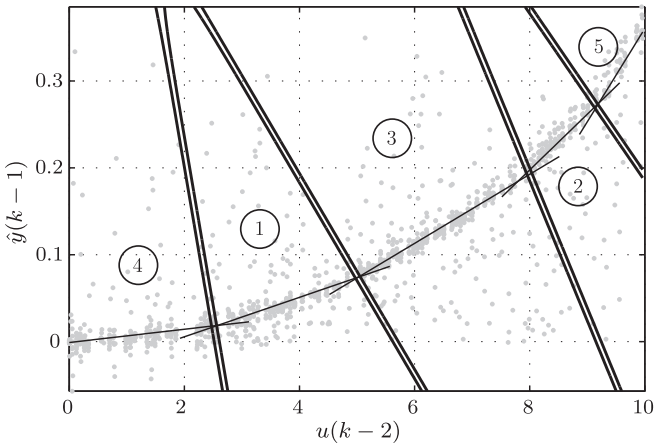

Fig. 8. Contour plot of validity functions (thick solid lines), the identification data (grey dots) and the local equilibria lines (thin solid lines).

Using an amplitude-modulated pseudo-random binary sequence as excitation, input–output data have been measured at 12.5 MHz and downsampled to a sampling frequency of 500 kHz, which is high enough to avoid aliasing effects in the considered frequency range of the human auditory system. A LMN with five local models and a model structure with sets $\mathcal{M}=\{2\}$ and $\mathcal{N}=\{1,2\}$ were identified. Therein, the shape and width of the validity functions (sigmoid functions) has been prescribed, but the local model parameters as well as the partitioning parameters of the hierarchical discriminant tree have been subject to simultaneous optimization using non-linear least squares (Hametner and Jakubek, 2013). The partitioning considers the input as well as the output with $\tilde{\mathcal{M}}=\{2\}$ and $\tilde{\mathcal{N}}=\{1\}$. Thus, according to the system classification in Table 1, no internal dynamics appear.

In Fig. 8 the identification data are shown and local models are represented by contour lines of their validity functions. As each local model is linear, a local equilibrium line can be found within each region of validity. These lines are also visible in Fig. 8. The measurement data clearly show that there appear strong transients, which require the output as well as the input in the partition space.

Fig. 9 illustrates the non-linear membrane displacement of the MEMS speaker for a 1 kHz triangular input signal with 10 V_{pp} and no feedforward control. Finally, the resulting feedforward control law reads as follows:

$$w(k+2)-\boldsymbol{c}^{\mathrm{T}}\left[\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{A}(\boldsymbol{\Phi}_k)\boldsymbol{w}_k+\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{B}(\boldsymbol{\Phi}_k)u(k)+\boldsymbol{A}(\boldsymbol{\Phi}_{k+1})\boldsymbol{f}(\boldsymbol{\Phi}_k)+\boldsymbol{f}(\boldsymbol{\Phi}_{k+1})\right]=0. \tag{68}$$

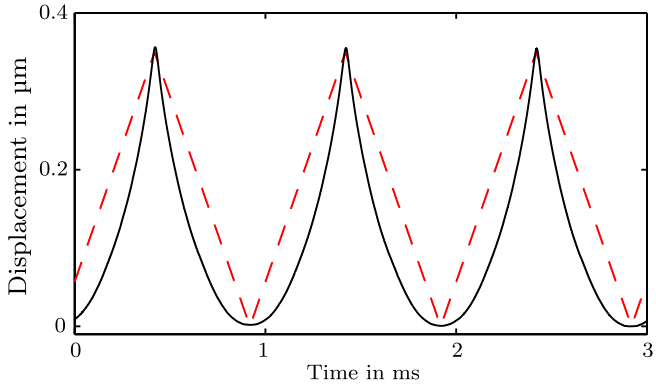

Fig. 9. Response to a 1 kHz/10 V_{pp} triangular input signal with the reference shown as dashed line and the measured output as solid line.

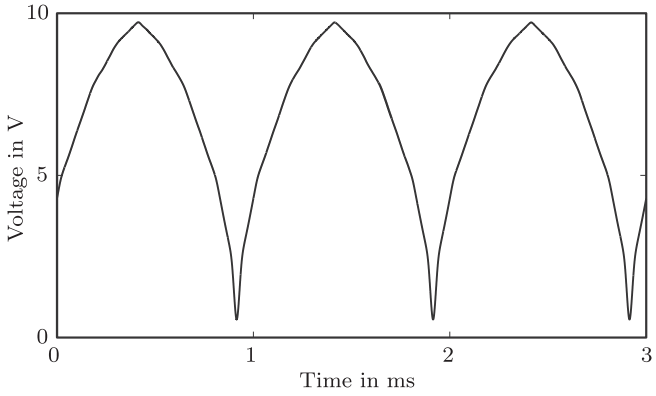

Fig. 10. Pre-distorted input signal for a 1 kHz/0.35 µm triangular reference.

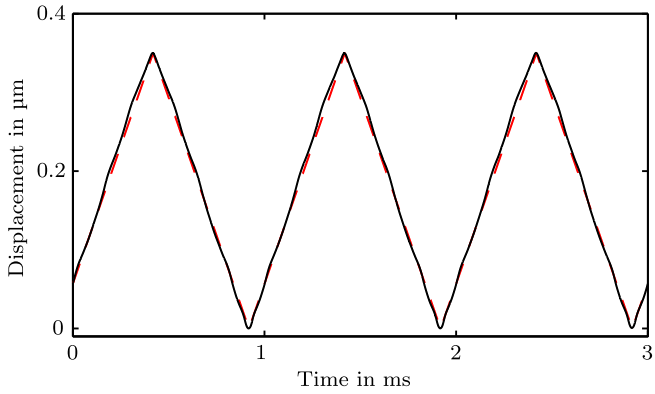

Fig. 11. Reference signal (dashed line) and resulting membrane displacement (solid line) using the pre-distorted input signal from Fig. 10.

Note that (68) is an implicit equation as $u(k)$ does not only appear explicitly but also in $\mathbf{\Phi}_{k+1}$.

The pre-distorted input signal found by the dynamic feedforward control law is shown in Fig. 10. It has been evaluated offline on a standard desktop computer using Matlab and has been applied to the MEMS speaker by means of a signal generator afterwards. This simple setup is possible as no feedback part is considered in this application. Finally, the resulting membrane displacement is given in Fig. 11. It becomes apparent that the membrane displacement dynamically follows the desired reference in an accurate way by using the proposed feedforward control already without feedback.

6. Conclusion and outlook

The application of feedback linearization to the generic model structure of local model networks yields a dynamic feedforward control law, which can be beneficially exploited in pre-distortion or two-degrees-of-freedom control schemes for example. For the automatic generation of the feedforward control law only input and output data of the underlying non-linear process are required. Altogether, directly from a greybox forward model with physical interpretability the inverse model is found. When the chosen model structure happens not to have full relative degree, the stability of the internal dynamics has to be checked. However, due to their complex structure, a global method to assess the stability is not yet available. By considering the eigenvalues of the internal dynamics, at least a momentary assertion can be made.

As the described state space representation and the choice of the state vector definition (3) is only one possible way to describe the input–output relation, other choices could be thought of as well. To find a state vector definition in an optimal way during system identification, there exist well-known subspace identification techniques (van Overschee and de Moor, 1994). In further research work, it will be investigated how these techniques can be applied to the local model network framework.

Acknowledgement

This work has been supported by the Christian Doppler Research Association and AVL List GmbH.

References

Boukezzoula, R., Galichet, S., Foulloy, L., 2003. Nonlinear internal model control: application of inverse model based fuzzy control. IEEE Trans. Fuzzy Syst. 11 (December (6)), 814–829.

Boukezzoula, R., Galichet, S., Foulloy, L., 2007. Fuzzy feedback linearizing controller and its equivalence with the fuzzy nonlinear internal model control structure. Int. J. Appl. Math. Comput. Sci. 17 (2), 233–248.

Byrnes, C.I., Isidori, A., 1984. A frequency domain philosophy for nonlinear systems, with applications to stabilization and to adaptive control. In: Proceedings of the 23rd Conference on Decision and Control, Las Vegas, NV, pp. 1569–1573, December.

Cerri, G., Borghetti, S., Salvini, C., 2006. Neural management for heat and power cogeneration plants. Eng. Appl. Artif. Intell. 19 (7), 721–730, Special issue on Novel Applications of Neural Networks in Engineering.

Chien, T.-L., Chen, C.-C., Huang, Y.-C., Lin, W.-J., 2008. Stability and almost disturbance decoupling analysis of nonlinear system subject to feedback linearization and feedforward neural network controller. IEEE Trans. Neural Netw. 19 (July (7)), 1220–1230.

Deng, H., Li, H.-X., Wu, Y., 2008. Feedback-linearization-based neural adaptive control for unknown nonaffine nonlinear discrete-time systems. IEEE Trans. Neural Netw. 19 (September (9)), 1615–1625.

Devasia, S., Chen, D., Paden, B., 1996. Nonlinear inversion-based output tracking. IEEE Trans. Autom. Control 41 (July (7)), 930–942.

Gao, R., O'Dwyer, A., Coyle, E., 2002. A nonlinear PID controller for CSTR using local model networks. In: Proceedings of the Fourth World Congress on Intelligent Control and Automation, Shanghai, PR China, pp. 3278–3282.

Gregorčič, G., Lightbody, G., 2007. Local model network identification with Gaussian processes. IEEE Trans. Neural Netw. 18 (5), 1404–1423.

Gregorčič, G., Lightbody, G., 2008. Nonlinear system identification: from multiple-model networks to Gaussian processes. Eng. Appl. Artif. Intell. 21, 1035–1055.

Gregorčič, G., Lightbody, G., 2010. Nonlinear model-based control of highly nonlinear processes. Comput. Chem. Eng. 34, 1268–1281.

Grizzle, J., 1986. Feedback Linearization of Discrete-Time Systems. In: Analysis and Optimization of Systems. In: Proceedings of the Seventh International Conference on Analysis and Optimization of Systems. Lecture Notes in Control and Information Sciences. Springer-Verlag Berlin, Heidelberg, New York, Tokyo, Antibes, pp. 1–9, 25–27 June.

Hafner, M., Schueler, M., Nelles, O., Isermann, R., 2000. Fast neural networks for Diesel engine control design. Control Eng. Pract. 8 (November (11)), 1211–1221.

Hagan, M.T., Demuth, H.B., De Jesús, O., 2002. An introduction to the use of neural networks in control systems. Int. J. Robust Nonlinear Control 12, 959–985.

Hametner, C., Jakubek, S., 2011. Nonlinear identification with local model networks using GTLS techniques and equality constraints. IEEE Trans. Neural Netw. 22 (September (9)), 1406–1418.
Hametner, C., Jakubek, S., 2013. Local model network identification for online engine modelling. Inf. Sci. 220, 210–225.
Hametner, C., Mayr, C., Kozek, M., Jakubek, S., 2014. Stability analysis of data-driven local model networks. Math. Comput. Model. Dyn. Syst. 20 (3), 224–247.
Hametner, C., Mayr, C.H., Kozek, M., Jakubek, S., 2013. PID controller design for nonlinear systems represented by discrete-time local model networks. Int. J. Control 86 (9), 1453–1466.
He, S., Reif, K., Unbehauen, R., 1998. A neural approach for control of nonlinear systems with feedback linearization. IEEE Trans. Neural Netw. 9 (November (6)), 1409–1421.
Henson, M.A., Seborg, D.E. (Eds.), 1997. Nonlinear Process Control, 1st edition Prentice Hall, January.
Hirschorn, R.M., 1979. Invertibility of multivariable nonlinear control systems. IEEE Trans. Autom. Control 24 (December (6)), 855–865.
Hunt, K., Sbarbaro, D., Zbikowski, R., Gawthrop, P., 1992. Neural networks for control systems—a survey. Automatica 28 (6), 1083–1112.
Isidori, A., 1995. Nonlinear Control Systems. Communications and Control Engineering Series, 3rd edition. Springer-Verlag, Berlin, Heidelberg, New York.
Isidori, A., 2013. The zero dynamics of a nonlinear system: from the origin to the latest progresses of a long successful story. Eur. J. Control 19, 369–378.
Isidori, A., Byrnes, C.I., 1990. Output regulation of nonlinear systems. IEEE Trans. Autom. Control 35 (February (2)), 131–141.
Jakubek, S., Hametner, C., 2009. Identification of neurofuzzy models using GTLS parameter estimation. IEEE Trans. Syst. Man Cybern. Part B Cybern. 39 (October (5)), 1121–1133.
Johansen, T.A., Shorten, R., Murray-Smith, R., 2000. On the interpretation and identification of dynamic Takagi–Sugeno fuzzy models. IEEE Trans. Fuzzy Syst. 8 (3), 297–313.
Karer, G., Mušič, G., Škrjanc, I., Zupančičs, B., 2011. Feedforward Control of a class of hybrid systems using an inverse model. Math. Comput. Simul. 82, 414–427.
Kotman, P., Bitzer, M., Kugi, A., 2010. Flatness-based feedforward control of a two-stage turbocharged diesel air system with EGR. In: 2010 IEEE International Conference on Control Applications (CCA), Yokohama, Japan, pp. 979–984, 8–10 September.
Lee, H.-G., Arapostathis, A., Marcus, S.I., 1987. Linearization of discrete-time systems. Int. J. Control 45 (5), 1803–1822.
Ljung, L., 2010. Perspectives on system identification. Annu. Rev. Control 34 (April (1)), 1–12.
Maass, B., Stobart, R., Deng, J., 2009. Diesel engine emissions prediction using parallel neural networks. In: 2009 American Control Conference, St. Louis, MO, USA, pp. 1122–1127, 10–12 June.
McKenney, D., White, T., 2013. Distributed and adaptive traffic signal control within a realistic traffic simulation. Eng. Appl. Artif. Intell. 26 (1), 574–583.
Monaco, S., Normand-Cyrot, D., 1987. Minimum-Phase Nonlinear Discrete-Time Systems and Feedback Stabilization. In: Proceedings of the 28th conference on Decision and Control, Los Angeles, CA, pp. 979–986, December.
Moulin, P., Chauvin, J., 2011. Modeling and control of the air system of a turbocharged gasoline engine. Control Eng. Pract. 19 (3), 287–297.
Murray-Smith, R., Johansen, T.A. (Eds.), 1997. Multiple Model Approaches to Modelling and Control. Taylor & Francis, London, UK.
Nelles, O., 2001. Nonlinear System Identification. Springer, Berlin, Heidelberg.
Nentwig, M., Mercorelli, P., 2008. Throttle valve control using an inverse local linear model tree based on a fuzzy neural network. In: The Seventh IEEE International Conference on Cybernetic Intelligent Systems. London, pp. 1–6, 9–10 September.
Nielsen, C., Fulford, C., Maggiore, M., 2010. Path following using transverse feedback linearization: application to a Maglev positioning system. Automatica 46 (3), 585–590.
Norgaard, M., Ravn, O.E., Poulsen, N.K., Hansen, L.K., 2000. Neural Networks for Modelling and Control of Dynamic Systems. Advanced Textbooks in Control and Signal Processing. Springer-Verlag London Limited.
Novak, J., Bobal, V., 2009. Predictive control of the heat exchanger using local model network. In: The 17th Mediterranean Conference on Control and Automation, Thessaloniki, Greece, pp. 657–662, 24–26 June.
Silverman, L.M., 1969. Inversion of multivariable linear systems. IEEE Trans. Autom. Control 14 (June (3)), 270–276.
Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Deylon, B., Glorennec, P., Hjalmarsson, H., Juditsky, A., 1995. Nonlinear Black-Box modeling in system identification: a unified overview. Automatica 31, 1691–1724.
Slotine, J.-J.E., Li, W., 1991. Applied Nonlinear Control. Prentice Hall, Englewood Cliffs, NJ.
Tanaka, K., Sugeno, M., 1992. Stability analysis and design of fuzzy control systems. Fuzzy Sets Syst. 45, 135–156.
Townsend, S., Irwin, G.W., 2001. Nonlinear model based predictive control using multiple local models. In: Non-Linear Predictive Control: Theory and Practice. IEE Control Engineering Series, vol. 61. Institution of Electrical Engineers, London, pp. 223–243.
Tuan, L., Lee, S.-G., Dang, V.-H., Moon, S., Kim, B., 2013. Partial feedback linearization control of a three-dimensional overhead crane. Int. J. Control Autom. Syst. 11 (4), 718–727.
Tumpold, D., Kaltenbacher, M., Glacer, C., Nawaz, M., Dehé, A., 2014. Multi field modeling of a microelectromechanical speaker system with electrostatic driving principle. Microsyst. Technol. 20 (4–5), 995–1006.
Tumpold, D., Stark, M., Euler-Rolle, N., Kaltenbacher, M., Jakubek, S., 2015. Linearizing an electrostatically driven MEMS speaker by applying pre-distortion. Sens. Actuators A 236, 289–298.
van Overschee, P., de Moor, B., 1994. N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica 30 (1), 75–93.