Title
Identification of nonlinear oscillator models for speech analysis and synthesis

Journal
Nonlinear Speech Modeling and Applications, 3445

ISSN
0302-9743

Authors
Kubin, G
Lainscsek, Claudia
Rank, E

Publication Date
2005

Peer reviewed

Identification of Nonlinear Oscillator Models for Speech Analysis and Synthesis*

Gernot Kubin[1], Claudia Lainscsek[2], and Erhard Rank[1]

[1] Signal Processing and Speech Communication Laboratory
Graz University of Technology, Graz, Austria
g.kubin@ieee.org, erank@ieee.org
[2] Cognitive Science Department
University of California at San Diego, La Jolla (CA), USA
clainscsek@ucsd.edu

Abstract. More than ten years ago the first successful application of a nonlinear oscillator model to high-quality speech signal processing was reported (Kubin and Kleijn, 1994). Since then, numerous developments have been initiated to turn nonlinear oscillators into a standard tool for speech technology. The present contribution will review and compare several of these attempts with a special emphasis on adaptive model identification from data and the approaches to the associated machine learning problems. This includes Bayesian methods for the regularization of the parameter estimation problem (including the pruning of irrelevant parameters) and Ansatz library (Lainscsek et al., 2001) based methods (structure selection of the model). We conclude with the observation that these advanced identification methods need to be combined with a thorough background from speech science to succeed in practical modeling tasks.

1 Introduction

The introduction of nonlinear system modeling – including the system theoretic dimension up to chaos theory – has evolved as an interesting tool for speech analysis and synthesis. With speech synthesis systems based on concatenation of recorded speech segments currently yielding the highest quality at current, but being inflexible and having high storage demands, and the alternative of model based synthesis algorithms, which are more versatile, but often difficult to control, parametric nonlinear models for the speech production process are an auspicious option.

An introduction to nonlinear oscillator models and its application to speech signals can be found in [1, 2]. Applications include time-scale modification [3], adaptive-codebook pulse code modulation for speech coding [1], noise reduction [4, 5], fundamental frequency analysis (pitch extraction) [6], determination of

* This chapter corresponds to talks given at the Cost 277 summerschool at IIASS in Vietri sul Mare (IT), in Sept. 2004. We would sincerely like to thank Anna Esposito for organizing the summerschool, and for her patience editing this publication.

instants of equal phase inside the glottis cycle (epoch marking) [7], as well as new concepts for speech recognition [8].

The application to speech signal modeling and re-synthesis has been presented in a number of investigations [1, 2, 9–16], however, some common drawbacks still to be addressed are the occasional lack of *stability* of the model, the *high number of parameters*, and the missing *relation to the physical process* of speech production. Furthermore, the oscillator model is often applied for voiced speech signals only, or even only for vowels.

Here we present some attempts to improving model identification, generalization to mixed excitation speech sounds, to the reduction of model complexity, and to the application of differential equations as a model for the underlying the speech production process.

2 Bayesian Regularization for Nonlinear Oscillator Identification

In this section we will describe a learning algorithm for modeling the oscillatory part of stationary speech signals by an oscillator based on Takens' time delay embedding theorem. The Bayesian algorithm presented here was found to achieve the highest percentage of successfully re-synthesized stationary speech signals in our experiments, as compared to other methods. 'Successful re-synthesis' means the generation of an *output speech signal similar to the training signal* in the way that – for voiced or mixed-excitation speech signals used during training – a stable oscillatory signal is generated without severe deviations from the training signal in waveform shape, amplitude, and fundamental frequency. In the last subsection we briefly present a method to re-generate also the noise-like component of speech signals in addition to the oscillatory component.

2.1 Discrete-Time Oscillator Model

Nonlinear time-series prediction in discrete-time $\hat{x}(n+1) = f(x(n), x(n-1), x(n-2), \ldots)$ can be immediately used to build an oscillator by applying the prediction function $f(\cdot)$ to a vector $\boldsymbol{x}(n)$ composed of past predicted samples:

$$x(n+1) = f(\boldsymbol{x}(n)) \ . \tag{1}$$

According to Takens' embedding theorem [17] the vector $\boldsymbol{x}(n)$ is commonly built as a time delay embedding of dimension N,

$$\boldsymbol{x}(n) = [x(n), x(n-M), x(n-2M), \ldots, x(n-(N-1)M)]^T \ , \tag{2}$$

realized as a tapped delay line fed by the predicted sample and with taps every M samples. A schematic of the model is depicted in Fig. 1.

Issues for oscillator identification are the determination of the optimal embedding parameters – embedding dimension N, and embedding delay M – as well as the identification of the prediction function $f(\cdot)$ from the training signal.

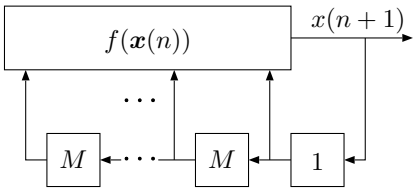


Fig. 1. Oscillator model.

2.2 Embedding Parameters

Based on Takens' theorem [17] the re-construction of dynamical properties of the system of dimension D, that gave rise to a time-series $x(n)$, is possible using a time delay embedding of dimension $N \geq 2D + 1$, regardless of the choice of the embedding delay (Takens derivation is for continuous time signals; of course the embedding delay must be $M > 0$). There exists a slightly refined theorem that allows for phase space re-construction in $N > 2D_f$, with D_f being the (possibly fractal) dimension of the attractor underlying the time-series [18]. However, both embedding theorems are sufficient, but not necessary conditions for re-construction of the dynamical properties [19], so we might well find an embedding of lower dimensionality than stated by the theorems with perfect (or sufficiently good) re-construction. The dimensionality of the embedding is most often related to the complexity of the prediction function $f(\cdot)$ in our oscillator.

The search for a reasonably low-dimensional embedding with good re-construction properties is facilitated by first looking for an optimal embedding delay M. To find a good predictor, i. e., to minimize the energy of the prediction error signal $e(n) = \hat{x}(n+1) - x(n+1)$, the single components of the embedding vector $x(n), x(n-M), \ldots$ shall each contribute as much information for the prediction as possible. Hence, ideally the components should be mutually independent. For a uniform embedding[3] as in (2) the embedding delay could be chosen at the first zero crossing of the auto-correlation function of the signal $s(n)$, making two neighboring components linearly independent (uncorrelated).

Minimizing linear dependence, however, might not be optimal for the embedding of signals coming from nonlinear systems (cf. [21, App. B]). Thus, the common approach is to minimize the statistical dependence between the components in the embedding by looking at the mutual information (MI) between delayed signal samples $x(n)$ and $x(n+L)$, and choosing the embedding delay as the delay at the first minimum of $\mathrm{MI}(L)$. Examples for the function $\mathrm{MI}(L)$ for some vowel signals are depicted in Fig. 2 (a). For vowel signals a first minimum of $\mathrm{MI}(L)$ is commonly found in the range of $5 \leq M \leq 15$ (for a signal sampling rate of $f_s = 16\,\mathrm{kHz}$). For mixed excitation signals, the first minimum of $\mathrm{MI}(L)$ – as depicted in Fig. 2 (b) – tends to be at smaller delays, or $\mathrm{MI}(L)$ monotonically decreases. A monotonically decreasing function $\mathrm{MI}(L)$ is also commonly

[3] Non-uniform embeddings, as suggested in [20], are considered in section 3 below.

observed for unvoiced speech signals. This raises the question of how to choose M for such signals.

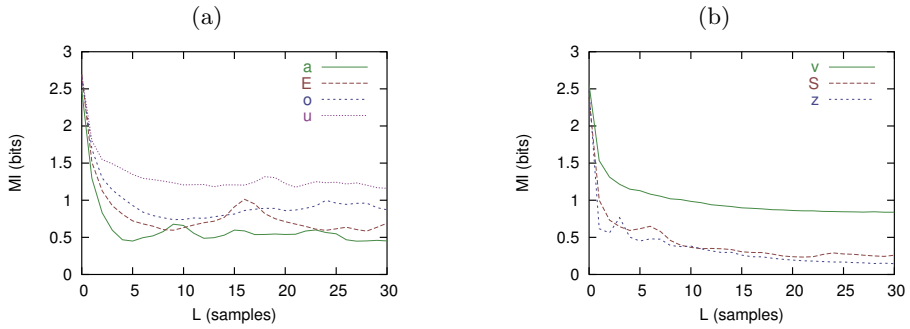


Fig. 2. Mutual information between signal samples as a function of delay L for (a) vowel signals, and (b) mixed excitation speech signals. Signals are refered to using labels from the machine-readable phonetic alphabet SAMPA. Signal sampling frequency is 16 kHz.

Still, there is another aspect of setting the embedding delay for the oscillator: Besides finding an embedding delay that minimizes the error of the *predictor*, another aim for the optimization of the embedding delay for the *oscillator* is to 'unfold' the signal trajectory in phase space, thus to prevent self-intersections (cf. [18]). As an example the time signal and the two-dimensional embeddings for two choices of the embedding delay for a mixed excitation speech signal are shown in Fig. 3. For the small embedding delay $M = 2$, chosen according to the first minimum in MI(L) (cf. Fig. 2 (b)), the trajectory of the signal in phase space evolves mainly on the diagonal and due to the noise-like signal component we can not identify the oscillatory component from the phase space plot in Fig. 3 (b). For a larger embedding delay $M = 13$, Fig. 3 (c), the oscillatory component becomes visible as a (still noisy) open loop of the trajectory. The identification of the signal structure by the nonlinear function in the oscillator can be considered equally difficult as the task of visually identifying the trajectory structure in the phase space plots, hence an oscillator based on an embedding with small embedding delay (and a reasonable low embedding dimension) will fail to re-produce the oscillatory signal component in this case.

Embedding delay M and embedding dimension N thus cannot always be chosen according to some fixed rule. Optimization of these parameters still may require (manual) inspection of the resulting oscillator output signal.

Our choice for the embedding parameters used throughout this section is motivated by the finding that for embeddings of *stationary* vowel signals a minimum of false neighbors [22] for $N \geq 4$ as well as a saturation of redundancy [23] for $N \geq 3$ is reached. Hence, an embedding dimension of $N = 4$ is used here. The embedding delay was chosen $M = 13$ for all signals, which represents a compromise choice based on MI(L) for vowel signals (cf. Fig. 2 (a)), but also yields

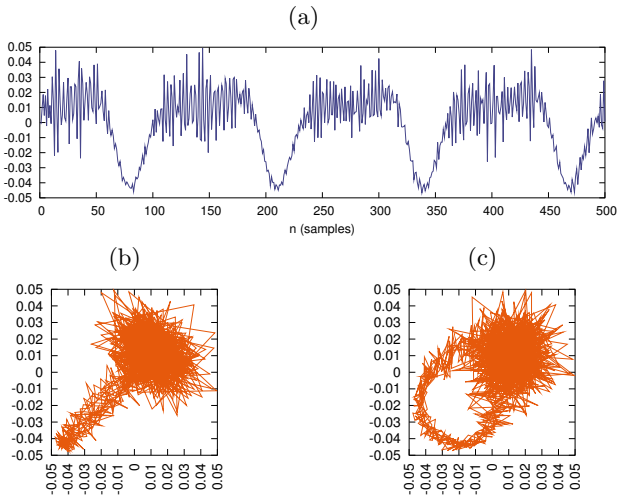


Fig. 3. Time-domain signal (a) of a mixed excitation speech sound /z/ and two-dimensional trajectories with (b) embedding delay $M = 2$ (at the first minimum of MI(L)), and (c) $M = 13$.

a sufficient unfolding of the trajectories of voiced speech signals (cf. Fig. 3). A fixed embedding delay is prefered over one optimized for each individual speech sound in our work to enable the synthesis of transitions (not described here).

2.3 Prediction Function

The prediction function $f(\cdot)$ in the oscillator is commonly realized as a trainable nonlinear mapping function, e. g., an artificial neural network (ANN) – such as a multilayer perceptron (MLP)[13], a radial basis function (RBF) network [12, 24, 15, 25, 16], or support vector machine (SVM) [26] – or by other function approximations such as multivariate adaptive regression splines (MARS) [27], classification and regression trees (CART), or lookup tables [28, 3].

All these function models have to be trained on input-output examples

$$(\boldsymbol{x}(n), x(n+1)) \quad , \qquad \mathbb{R}^N \times \mathbb{R} \quad , \tag{3}$$

derived from the original speech signal $x(n)$ according to a chosen embedding.

For many of the nonlinear function approximations listed above some kind of regularization has to be applied to avoid over-fitting – at least by controlling the complexity (number of parameters) of the function model – and to come up eventually with a stable oscillator for stationary speech signals. The exceptions are function realizations by lookup tables and CART, which produce a bounded output according to the range covered by the training signal. Oscillators based on lookup tables for stationary speech signals moreover entail an implicit modeling

of the noise-like signal part and have been found very appropriate, e. g., for time-scale modification [3][4].

MLPs with a nonlinear output layer may as well be confined to a bounded output. In our experiments, however, we found that even with a bounded output function MLP based oscillators often display no successful synthesis behavior that cannot be generally mitigated by reduction of model complexity or regularization methods such as early stopping or cross-validation. Nevertheless, more elaborate MLP structures and training algorithms (e. g., as described in [13]) may possibly yield better performance than our simple attempts.

Here, we shall focus on the realization of the nonlinear function by RBF based models, and present some methods for regularization that showed the highest number of 'successfully re-synthesized' stationary speech signals in our experiments. Particularly, we will limit the scope of our elaborations to RBF networks with a priori fixed Gaussian basis functions, i. e., RBF networks with center positions and widths of the Gaussian basis functions set to fixed values before network training. This restriction reduces the parameters of the RBF network that are optimized in the training process to the network output weights – and the training process for minimizing the squared prediction error to a problem which is linear in these parameters.

2.4 RBF Networks with Fixed Basis Functions

In an RBF network at each unit the distance of the input vector from a *center* (in input space) is computed and used as input for the according basis function. The network output is computed as the weighted sum of the basis function output values:

$$f_{\mathrm{RBF}}(\boldsymbol{x}) = \sum_{i=1}^{N_c} w_i \; \varphi_i(\|\boldsymbol{x} - \boldsymbol{c}_i\|) \;\; , \qquad \mathbb{R}^N \rightarrow \mathbb{R} \;\; . \tag{4}$$

In this equation $\boldsymbol{x}$ is the N-dimensional input vector, $\boldsymbol{c}_i$ are the center positions, $\varphi_i(\cdot)$ are the basis functions, and w_i the weighting coefficients. N_c is the number of basis functions and weights[5]. An RBF network has – as opposed to other kernel based function approximations – radially invariant basis functions. The basis functions used here are Gaussian functions,

$$\varphi_i(\|\boldsymbol{x} - \boldsymbol{c}_i\|) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_i\|^2}{2\sigma_i^2}\right), \tag{5}$$

centered at positions $\boldsymbol{c}_i$ in input space, and with variance σ_i^2. $\|\boldsymbol{x} - \boldsymbol{c}_i\|$ is the Euclidean vector distance between $\boldsymbol{x}$ and $\boldsymbol{c}_i$, making the basis function rotationally invariant.

[4] Interactive demo at http://www.nt.tuwien.ac.at/dspgroup/tsm.

[5] For other applications RBF networks often comprise an additional bias term, incorporated in the network function by using a constant 'basis function' $\varphi_0(\boldsymbol{x}) = 1$ and an according weight w_0. Since our speech signals are ensured to have zero mean, we do not make use of this bias term.

If the basis functions $\varphi_i(\boldsymbol{x})$ are fixed, i. e., if the center positions $\boldsymbol{c}_i$ and the variances σ_i^2 are set to a priori chosen values, the network parameters that have to be optimized in the training process are the weights w_i only. The training process becomes a linear problem: Applying the network equation (4) to all pairs of training input-output examples $(\boldsymbol{x}_k, t_k),\ k = 1, 2, \ldots, P$ results in the following vector-matrix equation:

$$\boldsymbol{t} = \boldsymbol{\Phi}\, \boldsymbol{w} \ , \tag{6}$$

with the training output examples ('targets') collected in vector $\boldsymbol{t} = [t_1, t_2, \ldots, t_P]^T$, the weights in vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_{N_c}]^T$, and the response of the basis functions to the training input examples in matrix

$$\boldsymbol{\Phi} = \begin{bmatrix} \varphi_1(\boldsymbol{x}_1) & \cdots & \varphi_{N_c}(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{x}_P) & \cdots & \varphi_{N_c}(\boldsymbol{x}_P) \end{bmatrix} \ .$$

Assuming that there are more training examples than basis functions $P > N_c$, training of the RBF network relates to solving (6) for the weight vector $\boldsymbol{w}$, which is done in a minimum mean squared error (MMSE) sense using the pseudo inverse $\boldsymbol{\Phi}^\dagger = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\,\boldsymbol{\Phi}^T$ of $\boldsymbol{\Phi}$,

$$\hat{\boldsymbol{w}}_{\mathrm{MMSE}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\,\boldsymbol{\Phi}^T\,\boldsymbol{t} \ . \tag{7}$$

Using the weights $\hat{\boldsymbol{w}}_{\mathrm{MMSE}}$ in the RBF network (4) yields the lowest possible squared prediction error for prediction of the training data. In the case when the number of network centers is equal to the number of training examples[6], $N_c = P$, the training output examples might even be perfectly predicted by the RBF network.

Notwithstanding the optimal prediction on the *training data*, an RBF network with weights $\hat{\boldsymbol{w}}_{\mathrm{MMSE}}$ often displays bad generalization, i. e., bad prediction on unseen *test data*. Due to the aim of optimal prediction of the training data the network output function may take arbitrary output values in regions of input space between (and outside) the training data samples. In the application of modeling speech signals with the oscillator model this commonly results in the occurrence of large amplitude intermittent spikes in the oscillator generated signal, as depicted in Fig. 4. Several investigations on using RBF networks in the oscillator model indicate the necessity of using regularization for RBF network training [12, 24, 15].

2.5 Regularized RBF Networks

Regularization refers to trading prediction accuracy on the training data (leading to over-fitting, and to large amplitude spikes in the oscillator output) for

[6] For example if the network centers are chosen equal to the training input vectors: $\boldsymbol{c}_i = \boldsymbol{x}_i,\ i = 1, 2, \ldots, P$.

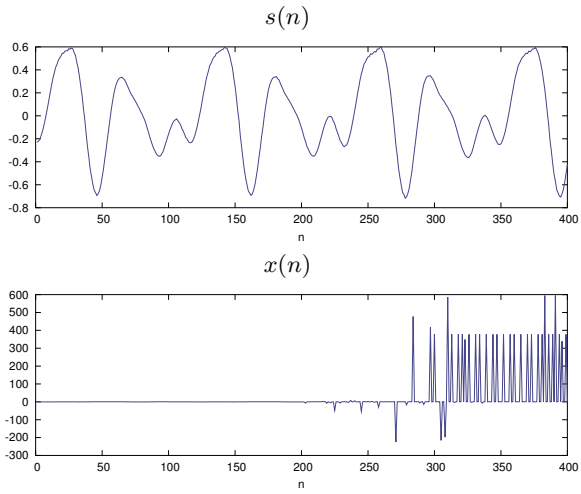


Fig. 4. Original speech signal $s(n)$, and oscillator output signal $x(n)$ using non-regularized RBF network training according to (7). Note the different scaling of the y-axes.

some other desired property of the prediction function. In our case this desired property is *smoothness* of the RBF network output function. Two ways of modifying the weight estimation (7) of an RBF network that impose smoothness on the network function are stated in the following.

The first is regularization of matrix inversion in (7), i. e., the regularization of the linear weight estimation task without taking the nonlinear nature of the basis functions into account. The modified training equation is [29]

$$\hat{\boldsymbol{w}}_{\mathrm{reg}} = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{t} \ , \tag{8}$$

with λ being the regularization parameter, and $\boldsymbol{I}$ the $N_c \times N_c$ identity matrix.

A method considering the nonlinear background, the Generalized Radial Basis Function (GRBF) expansion thoroughly derived in [30], arrives at a very similar equation for the RBF network weights

$$\hat{\boldsymbol{w}}_{\mathrm{GRBF}} = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \boldsymbol{\Phi}_0\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{t} \ , \tag{9}$$

again with λ being the regularization parameter, but the place of the identity matrix is taken by the $N_c \times N_c$ matrix $\boldsymbol{\Phi}_0$. This matrix is composed of the response of the basis functions to input vectors equal to the center positions:

$$\boldsymbol{\Phi}_0 = \begin{bmatrix} \varphi_1(\boldsymbol{c}_1) & \cdots & \varphi_{N_c}(\boldsymbol{c}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{c}_{N_c}) & \cdots & \varphi_{N_c}(\boldsymbol{c}_{N_c}) \end{bmatrix} \ .$$

For uni-modal basis functions (like Gaussians) $\boldsymbol{\Phi}_0$ has all ones on the diagonal, and smaller off-diagonal values.

Determining an optimal value for the regularization parameter λ is often based on cross-validation: The prediction error of a network is computed for an unseen validation data set for networks trained using a number of λ values of interest. The λ value yielding the minimum mean squared error on the validation set is chosen. k-fold cross-validation [31] makes best use of a small data set by partitioning the data and doing several cross-validation runs on the k partitions.

2.6 Bayesian Regularization and Pruning

The concept of Bayesian choice and training of function models has been presented in a series of papers [32–34] and refers to an iterative determination of function models (i. e., model structure and parameters) and regularization method and parameter(s) in the manner of the expectation-maximization (EM) algorithm [35]. We will here present the algorithm for Bayesian regularization for the RBF network with given fixed basis functions[7].

Bayesian Regularization of RBF Networks. In the Bayesian approach the RBF network weights $\boldsymbol{w}$ are considered random variables, with a prior probability density function (pdf) $p(\boldsymbol{w}|\alpha)$, parameterized by one parameter α. Incorporating regularization is done by stating a preference for smoother network functions by choosing the prior for the weights as a zero mean Gaussian distribution with variance α^{-1}:

$$p(\boldsymbol{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{-\frac{N_c}{2}} \exp\left(-\frac{\alpha}{2}\|\boldsymbol{w}\|^2\right) \ . \tag{10}$$

Furthermore, it is assumed that the training output samples t_k are produced by an additive noise model

$$t_k = f_{\mathrm{RBF}}(\boldsymbol{x}_k) + \epsilon_k \ , \qquad p(\epsilon) = \mathcal{N}(0, \sigma_n^2) \ , \tag{11}$$

with the function $f_{\mathrm{RBF}}(\cdot)$ from (4) and additive zero-mean Gaussian noise samples ϵ_k with variance σ_n^2. The noise variance σ_n^2 is the second additional parameter (besides α) introduced by this Bayesian formulation. As in the scope of the expectation-maximization algorithm these additional parameters are called *hidden* or *hyper-parameters*. Like the weights the hyper-parameters are unknown parameters, and have to be characterized by an a priori chosen pdf, too. Since both additional parameters are scaling parameters, proper prior distributions are, e. g., uniform distributions on a logarithmic scale.

[7] That means, we shall ignore the possibility of a Bayesian choice of hypothesis $\mathcal{H}$ (network complexity) and set of basis functions $\mathcal{A}$ considered in [32]. Furthermore we will restrict the derivation of the training algorithm to one specific choice of regularizer $\mathcal{R}$.

The aim in Bayesian network training is to find the most probable values for the weights and the hyper-parameters given the training data (let $\boldsymbol{X}$ represent the collected training input vectors), i. e., to maximize $p(\boldsymbol{w}, \alpha, \sigma_n^2 | \boldsymbol{X}, \boldsymbol{t})$:

$$(\boldsymbol{w}, \alpha, \sigma_n^2)_{\text{bay}} = \arg\max(p(\boldsymbol{w}, \alpha, \sigma_n^2 | \boldsymbol{X}, \boldsymbol{t})) \ . \tag{12}$$

Since this maximization cannot be accomplished analytically, the task is divided in two steps, maximizing the probability of the weights values for given training data and hyper-parameters, and updating the hyper-parameters, corresponding to a decomposition $p(\boldsymbol{w}, \alpha, \sigma_n^2 | \boldsymbol{X}, \boldsymbol{t}) = p(\boldsymbol{w} | \boldsymbol{X}, \boldsymbol{t}, \alpha, \sigma_n^2)\, p(\alpha, \sigma_n^2 | \boldsymbol{X}, \boldsymbol{t})$.

The first part of this decomposition can be evaluated analytically: For given hyper-parameters the resulting pdf for the weights $p(\boldsymbol{w} | \boldsymbol{X}, \boldsymbol{t}, \alpha, \sigma_n^2)$ is a product of Gaussian pdfs, and thus a multivariate Gaussian distribution itself,

$$p(\boldsymbol{w} | \boldsymbol{X}, \boldsymbol{t}, \alpha, \sigma_n^2) = (2\pi)^{-\frac{N_c}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{w} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{w} - \boldsymbol{\mu}) \right) \ , \tag{13}$$

with covariance and means, respectively:

$$\begin{aligned} \boldsymbol{\Sigma} &= (\tfrac{1}{\sigma_n^2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \alpha \boldsymbol{I})^{-1} \ , \\ \boldsymbol{\mu} &= \tfrac{1}{\sigma_n^2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{t} \ . \end{aligned} \tag{14}$$

Maximization of (13) consists of setting the weights $\boldsymbol{w}$ equal to the mean values $\boldsymbol{\mu}$:

$$\hat{\boldsymbol{w}}_{\text{bay}} = \boldsymbol{\mu} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \underbrace{\alpha\, \sigma_n^2}_{\lambda_{\text{bay}}} \boldsymbol{I})^{-1} \boldsymbol{\Phi}^T \boldsymbol{t} \ . \tag{15}$$

Note, that – relating to regularization by matrix inversion (8) – the product of the hyper-parameters takes the role of the regularization parameter, $\lambda_{\text{bay}} = \alpha\, \sigma_n^2$.

The second part in the above decomposition is again split by $p(\alpha, \sigma_n^2 | \boldsymbol{X}, \boldsymbol{t}) \propto p(\boldsymbol{t} | \boldsymbol{X}, \alpha, \sigma_n^2)\, p(\alpha)\, p(\sigma_n^2)$ (where we skip the normalization by $p(\boldsymbol{t}) = \text{const.}$ and assume no conditioning of the hyper-parameters on the input training data $p(\alpha | \boldsymbol{X}) = p(\alpha), p(\sigma_n^2 | \boldsymbol{X}) = p(\sigma_n^2)$). Maximization of the right hand side terms depends on assumptions for the prior distributions of the hyper-parameters $p(\alpha)$ and $p(\sigma_n^2)$, and is treated for uniform distributions on logarithmic scale and for Gamma distributions in [32, 36]. For uniform prior distributions on a logarithmic scale the resulting update equations[8] for the hyper-parameters to maximize

[8] Deviating from the exact Bayesian approach the following approximations are made: As already noted, the derivation of the weights from (13)-(15) assumes the hyper-parameters to be known, which corresponds to taking a delta-distribution for $p(\alpha, \sigma_n^2 | \boldsymbol{t})$. On the other hand in the update of the hyper-parameters (16) the parameters of the posterior pdf for the weights are assumed to be known. Furthermore, the update of the hyper-parameters (16) is not exactly a maximization step, since the calculation of γ involves the old value for α on the right-hand side of the equation. Hence, the Bayesian learning can only be accomplished in an iterative procedure.

$p(\alpha, \sigma_n^2 | \boldsymbol{X}, \boldsymbol{t})$ are

$$\begin{aligned} \alpha^{\text{new}} &= \frac{\gamma}{\|\boldsymbol{\mu}\|^2} \ , \\ \left(\frac{1}{\sigma_n^2}\right)^{\text{new}} &= \frac{\|\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2}{P - \gamma} \ , \\ \gamma &= N_c - \alpha \, \text{Trace}(\boldsymbol{\Sigma}) \ . \end{aligned} \tag{16}$$

Bayesian optimization of weights and hyper-parameters (and thus of regularization) comprises iterating (14) and (16), in the manner of the EM algorithm:

$$\begin{aligned} \boldsymbol{\Sigma}^{(i)} &= (\tfrac{1}{\sigma_n^{2\,(i)}} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \alpha^{(i)} \boldsymbol{I})^{-1} \ , \\ \boldsymbol{\mu}^{(i)} &= \tfrac{1}{\sigma_n^{2\,(i)}} \boldsymbol{\Sigma}^{(i)} \boldsymbol{\Phi}^T \boldsymbol{t} \ , \\ \gamma^{(i)} &= N_c - \alpha^{(i)} \, \text{Trace}(\boldsymbol{\Sigma}^{(i)}) \ , \\ \alpha^{(i+1)} &= \frac{\gamma^{(i)}}{\|\boldsymbol{\mu}^{(i)}\|^2} \ , \\ \frac{1}{\sigma_n^{2\,(i+1)}} &= \frac{\|\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{\mu}^{(i)}\|^2}{P - \gamma^{(i)}} \ . \end{aligned} \tag{17}$$

Here, $i = 1, 2, \ldots$ is the iteration index.

For the application in the oscillator model the iteration in (17) was found to converge in the generic case[9]. Starting with an initialization with $\sigma_n^{2\,(1)} = 10^{-4}$, $\alpha^{(1)} = 10^{-4}$, the mean number of iterations for less than 1 % variation in the hyper-parameters γ and σ_n^2 was 19.3 for signals from a database of sustained vowels and nasals, and even smaller for voiced and unvoiced fricatives. This means that the computational complexity of the Bayesian algorithm for determining the regularization factor is comparable to the computational complexity of cross-validation over a range of, e. g., $\lambda \in [10^{-12}, 10^2]$ with one λ value per decade and one validation set, and smaller by a factor of k as compared to k-fold cross-validation.

The value of the regularization parameter λ_{bay} from Bayesian training for vowel signals is, on average, four orders of magnitude higher than the value determined by cross-validation. A closer look at the error function on the validation set in the cross-validation procedure reveals that, for many vowel signals, this function displays a flat minimum with very small variation over several orders of magnitude, as in Fig. 5, due to the high signal to noise ratio (SNR) of the training signal. As a consequence, an arbitrary λ value within this large range can be chosen to yield approximately the same small prediction error on the validation set. The actual position of the minimum in the validation error highly depends on training parameters, like choice of validation set, training length, etc., as exemplified in Fig. 6. For the application in a *predictor* the actual choice of λ_{xv} does not make a big difference. However, for the *oscillator* a minimum λ value is

[9] A 'non-generic' case encountered was the oscillator training on a strictly periodic signal, where the Bayesian algorithm did not converge.

necessary to yield stable oscillation (as indicated in Figs. 5 and 6). The Bayesian algorithm, on the other hand, always chooses a λ value at the higher end of a flat validation error function (resulting in a possibly slightly larger squared error), as shown in Figs. 5 and 6, and thus results in a significantly larger number of stably re-synthesized vowel signals.

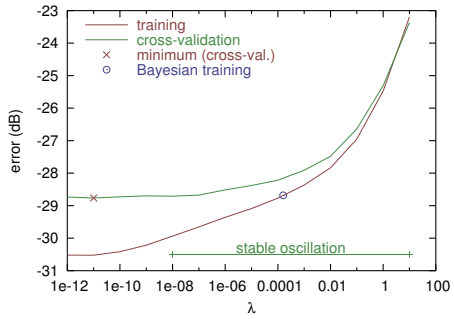


Fig. 5. Training and validation error as a function of λ for male vowel /o/, along with the choice of λ_{xv} (cross) according to cross-validation, and the value of λ_{bay} (circle) found by the Bayesian algorithm.

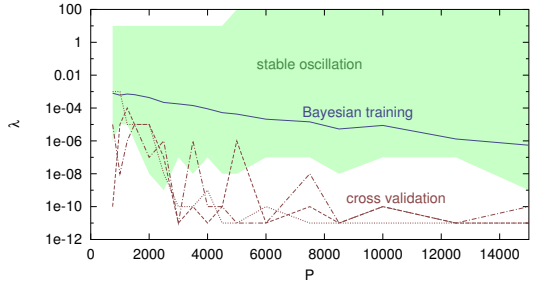


Fig. 6. Value λ for the regularization parameter as a function of training examples P found by the Bayesian algorithm (solid line), and according to cross-validation with a validation set of 10, 20, and 30 % of the training data (dashed/dotted lines). The shaded region indicates λ values, where stable re-synthesis of this vowel is possible.

The difference between the value of the regularization parameter found by cross-validation and by the Bayesian algorithm is less stringent for voiced and unvoiced fricatives, which do not display a flat minimum of the validation error function.

Testing the Bayesian algorithm on artificial speech source signals according to the Liljencrants-Fant model [37] with additive noise, and artificially introduced

variations in fundamental period length (jitter) and amplitude of individual fundamental cycles (shimmer), a very robust behavior is found. In Fig. 7 it can be seen that regularization is increased with increasing noise level, and, regardless of the amount of jitter and shimmer, the Bayesian algorithm yields a robust estimate of the actual noise level. A similar robust behavior has been found for the modeling of chaotic signals from the Lorenz system with the oscillator model and Bayesian regularization [16].

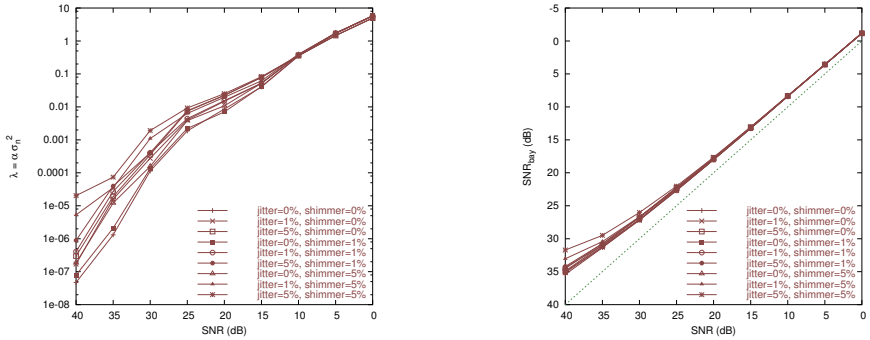


Fig. 7. Regularization factor λ_{bay} and noise variance σ_n^2 (displayed as SNR_{bay}) found by the Bayesian training as a function of training signal SNR for artificial speech source signals.

Pruning of Basis Functions. An extension of the Bayesian learning algorithm for kernel based function approximation, like RBF networks, is the relevance vector machine (RVM) as described in [36]. As opposed to the above approach where regularization is introduced by the choice of a Gaussian prior pdf for the norm of the weights vector (10), in the RVM the prior pdf for the network weights is a product of individual Gaussians for each weight:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N_c} p(w_i|\alpha_i) \ ,$$
$$p(w_i|\alpha_i) = \left(\tfrac{\alpha_i}{2\pi}\right)^{-\frac{1}{2}} \exp\left(-\tfrac{\alpha_i}{2} w_i^2\right) \ . \tag{18}$$

Instead of the hyper-parameter α, a number of N_c hyper-parameters $\alpha_i, \quad i = 1 \ldots N_c$ are introduced by this model.

Again a decomposition of the pdf for the unknown parameters similar as for (12) and an iterative training algorithm can be applied (for details see [36]). During RVM learning, however, some of the hyper-parameters α_i attain large values. This means that the pdf for the according weight w_i is concentrated around zero, and that the weight value almost certainly is close to zero. Hence, the basis function φ_i does not contribute a *relevant* part to the output, and can be pruned.

When pruning basis functions for $\alpha_i > 10^6$, and stopping iterations when the number of network centers is not reduced within the last ten iterations, the mean number of iterations is 60.2 for our vowel database. The number of initial basis functions used ($N_c = 625$) is, however, significantly reduced during the RVM training iterations[10], as depicted in Fig. 8.

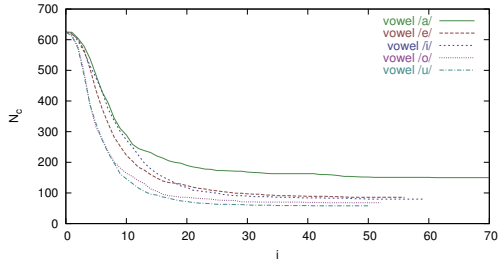


Fig. 8. Reduction of the number of basis functions N_c during iterations of RVM training for some example vowel signals.

Using the RVM for prediction yields a prediction error not more than 1 dB higher than for Bayesian trained RBF networks without pruning. Also the oscillator model with an RVM yields stable re-synthesis of almost the same number of vowel signals than without pruning. One possible reason that some signals could be re-synthesized without pruning, and cannot with the RVM, seems to be the fact that, since less basis functions are used in the RVM, the weights for these basis functions may again take larger values than without pruning.

Concerning the almost equal prediction gain and number of stably re-synthesized vowel signals the RVM can be considered a valuable tool for reducing the complexity of kernel based function models like RBF networks in applications like speech signal prediction and the oscillator model. Other applications of the RVM algorithm comprise, for example, the determination of relevant weights in adaptive filters, where the complexity of the adaptive filters is also considerably reduced without impairing performance [38].

2.7 Inverse Filtering and Oscillator Model

The application of regularization for nonlinear function learning increases the number of signals that can be stably re-synthesized with the oscillator model. However, for modeling the *full speech signal* this number is still fairly low (only 18 % for our vowel database and a fixed embedding with $N = 4$ and $M = 13$).

[10] Note, that – in spite of the higher number of iterations – the total computational complexity may be less than in the Bayesian training without pruning: The main effort is the inversion of an $N_c \times N_c$ matrix, which is of order $\mathcal{O}({N_c}^3)$. Since N_c is reduced during the iterations, computational effort is also shrinking considerably.

Inspection of the structure of the signal trajectories in phase space reveals, that vowels with a simple trajectory structure, like /o/ or /u/ can be stably re-synthesized more likely than vowels with a more complicated structure, like /a/, /e/, and /i/. To gain a generally simple structure of the signals that are to be modeled by the oscillator, inverse filtering in combination with the oscillator model can be applied [13, 16]. Inverse filtering refers to the identification and compensation of the influence of the vocal tract on a glottis source signal. Here we utilize a simple inverse filtering process, since we do not aim at the identification of the source signal, but want to arrive at a simple trajectory structure only.

The inverse filtering process used for the examples here consists of identification and application of a signal dependent linear prediction (LP) inverse filter with transfer function $A(z) = 1 + a_1 z^{-1} + \ldots + a_{N_{\mathrm{LP}}} z^{-N_{\mathrm{LP}}}$, and subsequent low-pass filtering. First-order pre-emphasis filtering with a zero at $z_n = 0.75$ is used for LP analysis (but not in the signal path). For the low-pass filter $H(z)$ a first order recursive filter with a pole at $z_p = 0.95$ is applied. This low-pass filter can be stably inverted for synthesis.

Oscillator model identification is now done for the output signal $x_g(n)$ of the low-pass filter. For re-synthesis, the oscillator generated signal $y_g(n)$ is filtered by $1/H(z)$ and the LP synthesis filter $1/A(z)$ to gain the synthetic full speech signal $y(n)$. To exemplify the benefit of inverse filtering, time signals and phase space trajectories for the vowel /a/ from a female speaker are depicted in Fig. 9. This signal could not be re-synthesized in the full speech signal domain. By inverse filtering, however, the highly intermingled trajectory of the full speech signal $x(n)$ is converted to an open loop trajectory of $x_g(n)$. This simpler structure can be identified and stably re-synthesized by the oscillator model with Bayesian regularization (without pruning, in this example).

By means of inverse filtering the percentage of stably re-synthesized vowel signals from our database for an embedding with $N = 4$ and $M = 13$ is increased from 18 % to 56 %. Also the spectral reproduction of speech signals is improved as compared to full speech signal modeling. However, perceptually many re-generated vowel signals still are not satisfactory, and in particular for mixed excitation speech signals, like voiced fricatives, the oscillator fails to adequately reproduce the signal quality. We attribute this to the missing modeling of *high-dimensional* or *stochastic* speech signal components by the low-dimensional and deterministic oscillator model, and present a model that accounts for a stochastic noise-like signal component in speech signals in the following.

2.8 Oscillator-plus-Noise Model

For purely unvoiced speech signals a satisfactory synthesis system is the auto-recursive stochastic model (i. e., noise excited LP) [39]. Concerning the generation of mixed excitation speech signals it is known that a modulation of the noise-like signal component synchronized in phase with the oscillatory signal component is requisite [40–46]. To achieve this synchronized modulation for oscillator model generated speech signals we propose to use an amplitude prediction for the noise-like signal component by a second nonlinear function $f_n(\cdot)$

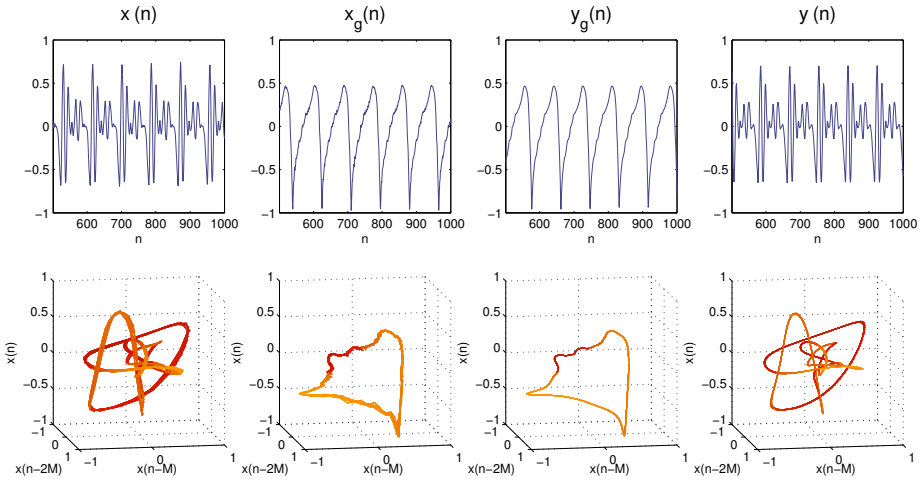


Fig. 9. Time signals (top row), phase space trajectories (bottom row) of the original full speech signal $x(n)$, the signal after inverse filtering $x_g(n)$, as well as the oscillator generated signal $y_g(n)$ and the resulting synthetic full speech signal $y(n)$ for vowel /a/ from a female speaker.

from the state of the oscillator model:

$$\tilde{a}(n+1) = f_n(\boldsymbol{x}(n)) \ , \tag{19}$$

and to add a modulated noise signal to the oscillator model output signal.

Based on the assumption, that the predictor used in the oscillator model is able to capture the low-dimensional oscillatory component of speech signals, but not the high-dimensional noise-like component, the prediction function for the noise-like signal component's amplitude $f_n(\cdot)$ in (19) is trained to predict the amplitude of the prediction error signal $e(n) = \hat{x}(n) - x(n)$ of the nonlinear signal predictor (1).

For modeling general mixed excitation speech sounds we found [47] that it is also necessary to use a second LP analysis path for the noise-like signal part to achieve the spectral shaping of this signal component independent from the spectral properties of the oscillatory signal component.

A schematic of the analysis and synthesis process of the resulting *oscillator-plus-noise model* is given in Fig. 10. The upper signal path – from input signal $x(n)$ to oscillatory output component $y_{\mathrm{osc}}(n)$ represents the model comprising inverse filtering as described in the last subsection. All additional processing is based on the above assumption that the *prediction error* of the nonlinear function $f()$ (in the oscillator) is *related to the noise-like signal component*. Hence, the predicted signal $\hat{x}_g(n)$ is considered to be the oscillatory part of the training signal. To arrive at the noise-like signal component, $\hat{x}_g(n)$ is filtered by the synthesis filters $1/H(z)$ and $1/A(z)$ and subtracted from the training signal to yield an estimate for the noise-like signal component $\hat{x}_{\mathrm{noi}}(n)$. Synthesis filtering

on $\hat{x}_g(n)$ is done to provide the means for an *individual spectral shaping* of the noise-like component independent from the oscillatory component by the LP filter $A_{\mathrm{noi}}(z)$. From the according residual signal $x_{r\_\mathrm{noi}}(n)$ the amplitude trajectory is extracted by rectification and moving average filtering. The function $f_n(\cdot)$ in (19) is trained to predict this amplitude trajectory $\hat{a}_{\mathrm{noi}}(n)$ based on the trajectory of the state of the predictor $\boldsymbol{x}_g(n-1)$.

At the bond between analysis and synthesis the speech signal is represented by parameters only, namely the coefficients for the nonlinear functions in the oscillator and the noise amplitude predictor, and the coefficients for the LP filters.

In the synthesis stage the oscillatory speech signal component is generated by autonomous synthesis as in the previous models. In addition the amplitude trajectory $\tilde{a}_{\mathrm{noi}}(n)$ of the synthetic signal's noise-like component is predicted from the oscillator state $\boldsymbol{y}_g(n-1)$ by the function $f_n(\cdot)$. A white Gaussian noise source is modulated in amplitude by $\tilde{a}_{\mathrm{noi}}(n)$ and fed to the LP synthesis filter $1/A_{\mathrm{noi}}(z)$ to yield the noise-like signal component in the full speech signal domain $y_{\mathrm{noi}}(n)$, which is added to the oscillatory component $y_{\mathrm{osc}}(n)$, yielding the output signal $y(n)$.

As depicted for an example signal in Fig. 11, the oscillator-plus-noise generated signals display a spreading of the signal trajectory in phase space and a spectral noise floor, resembling the behavior of the natural signals' trajectory and spectrum, in contrary to the signals generated by the oscillator alone – which have a concentrated trajectory and a line spectrum.

Examples for the noise amplitude modulation achieved by the oscillator-plus-noise model are given in Fig. 12. For both the vowel /o/ and the voiced fricative /v/ the model achieves a *pitch-synchronous modulation* of the amplitude $\tilde{a}(n)$ for the noise-like signal component similar to the amplitude modulation $\hat{a}(n)$ of the prediction error signal. However, the two example signals display a very distinct form of the noise amplitude trajectory over the pitch cycle: Whereas the vowel signal has triangular maxima of the noise amplitude at the minima of the signal $y_g(n)$ – which is a typical behavior for vowels and often modeled using a parametric envelope [45, 48] – the noise amplitude for the voiced fricative has *minima* synchronized with the minima of $y_g(n)$.

In the oscillator-plus-noise model the envelope for the noise modulation is, for any individual training signal, identified by the prediction function $f_n(\cdot)$, in a similar manner as the oscillatory waveform is identified by the nonlinear function $f(\cdot)$ in the oscillator model.

3 ODE and DDE Models from Delay- and Differential-Embeddings

Recovery of nonlinear dynamical processes from single scalar time series observations is extremely difficult when no information on the functional form of the underlying process is available. In this section we present the reconstruction of a dynamical process from a differential embedding using ordinary differential

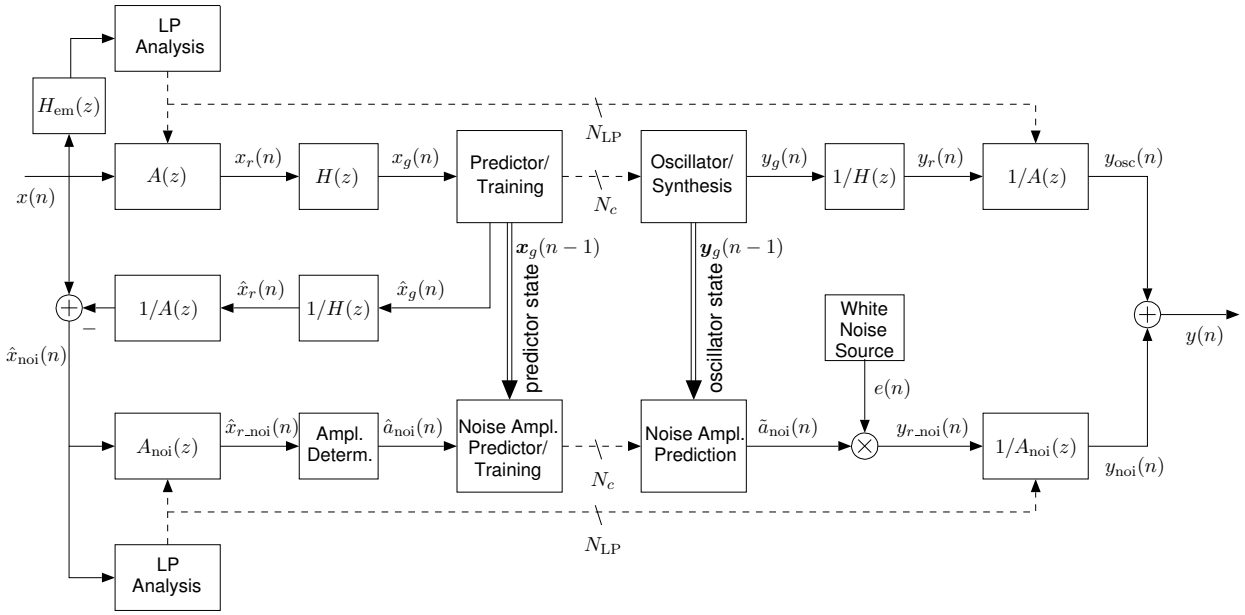


Fig. 10. Oscillator-plus-noise model.

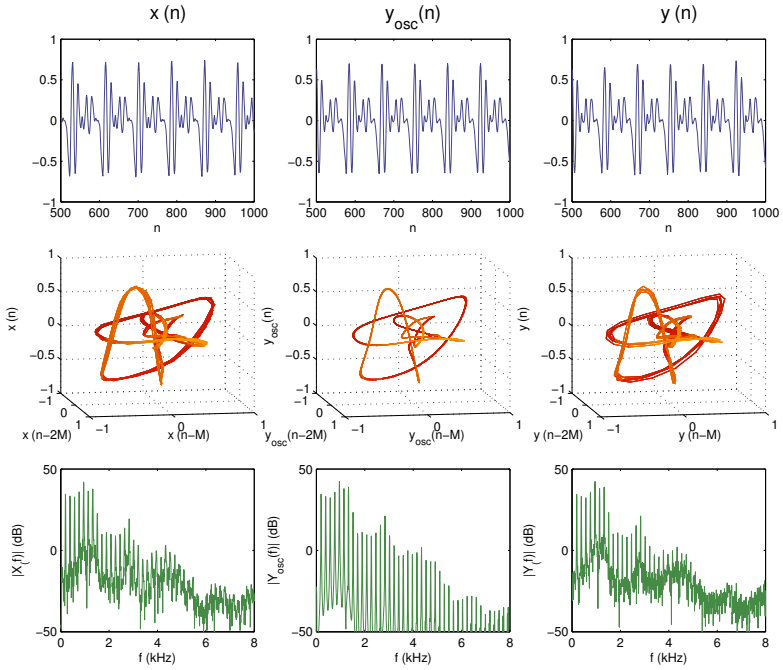


Fig. 11. Time signals (top row), phase space trajectories (middle row), and DFT spectra (bottom row) of the original full speech signal $x(n)$, the oscillator generated synthetic full speech signal $y_{\mathrm{osc}}(n)$, and the oscillator-plus-noise generated synthetic full speech signal $y(n)$ for female /a/.

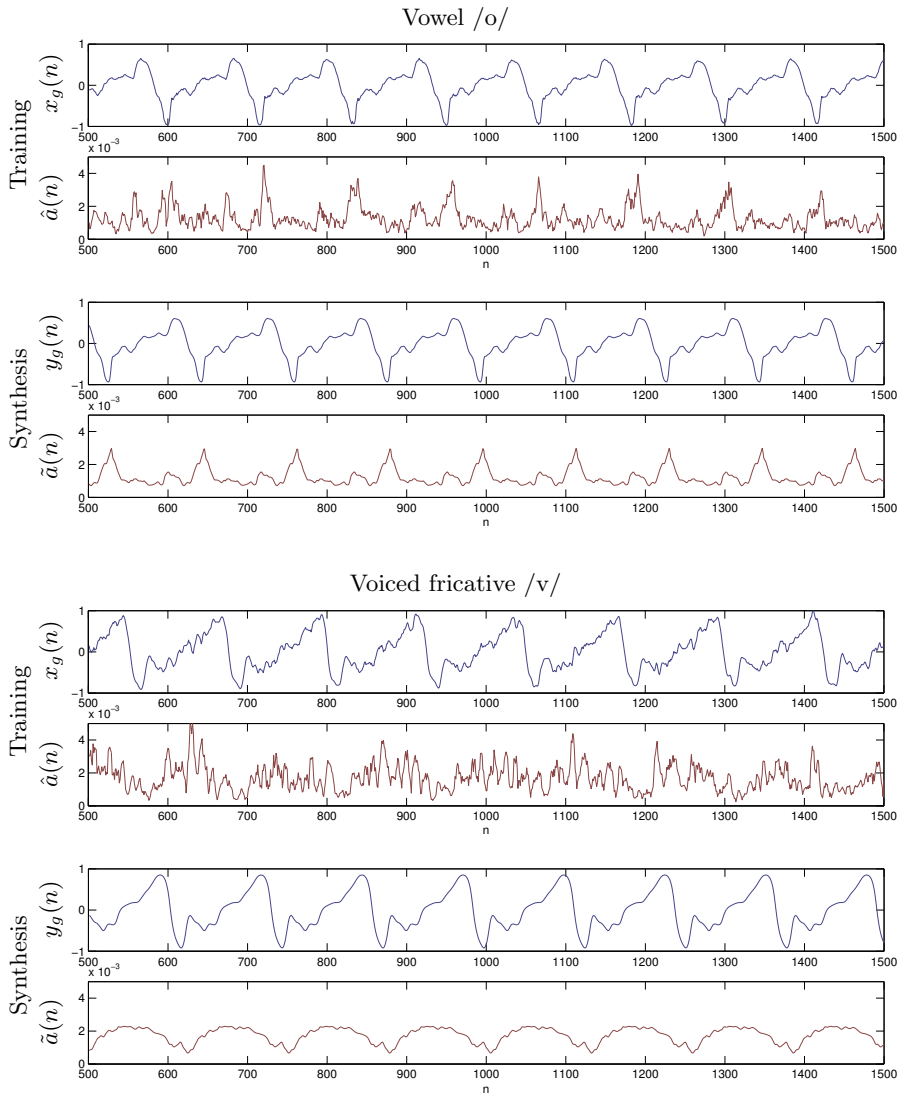


Fig. 12. Two examples for the modeling of the noise amplitude trajectory.

equations (ODEs) from an "Ansatz library", the optimization of delay differential equations (DDEs) using a genetic algorithm (GA), and some analysis results for speech characterization that can be read from the parameter values of GA optimized DDEs.

Many time series analysis methods for modeling, prediction, and classification of experimental observations rest upon nonlinear dynamical systems theory. An important analysis tool in nonlinear system theory is reconstructing phase space topological properties of a dynamical system from a single scalar time series. The embedding theorems of Takens [17], and Sauer [18] assure us that an embedding constructed from a single variable time series preserves the topological properties of the underlying dynamical system and, therefore, lays out the foundations for such reconstruction methods.

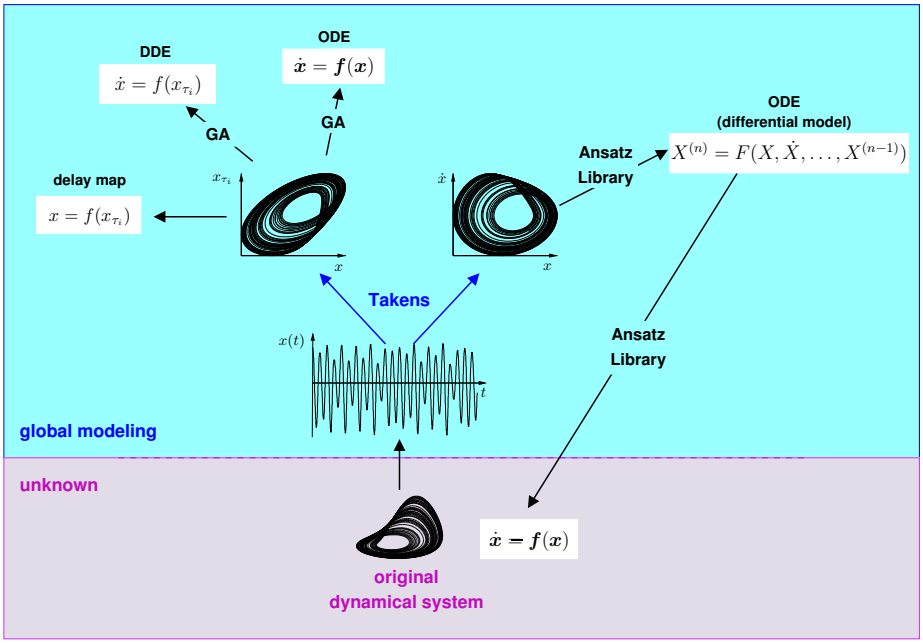


Fig. 13. Framework for global modeling from a single time series via delay- or differential- embedding: typically only a restricted set of measurements, here a single time series, is available. The underlying dynamical system is unknown. To recover information about the underlying system, following the theorems of Takens [17], either a delay- or a differential-embedding can be constructed in order to reconstruct a global model. The model form reconstructed from a delay embedding can be a delay map, a set of ODEs, or a DDE and, from a differential embedding, a set of ODEs.

Thus, it is possible to model a multi-dimensional nonlinear process directly from the scalar data without any prior knowledge of the physical processes, such

that these models reproduce the data with the original dynamical properties. Two types of models have been used. The first encompasses local models of the dynamics which allow to predict, step by step, the evolution of the system. The most common model of this type is the auto-regressive moving average (ARMA) model. The second type consists of global models that attempt to characterize the global underlying dynamics of a given process. Ideally, global models can generate long time series with the same dynamical properties as the original ones.

In the simple case of noise-free observations of all original dynamical variables of a physical system, the global model can be exactly estimated if we know the precise model order, that is if we know the exact underlying structure of the original system for which we estimate the parameters. In reality, one typically does not know the proper dimensionality or functional model form of the underlying dynamical process, especially when only a scalar observation of the process is available. If this is the case, estimation of a global model can be very difficult. Any error in the assumptions made about the model dimension or its functional form may critically effect the model quality.

Here we address the problem of finding a global dynamical model in the form of a set of coupled ordinary differential equations (ODEs) from a single time series. We introduce an Ansatz library [49, 50] based method (see Fig. 13) in Sect. 3.1 and a Genetic Algorithm (GA) based method in Sect. 3.2 to find such a system of ODEs which models the entire underlying nonlinear dynamical process. In these two methods we construct from the time series either a differential embedding (Ansatz library based method) or a delay embedding (GA based method) using a non-uniform embedding [51, 20] which assumes that the given time series can have different time scales. In Sect. 3.3 we estimate DDE (delay differential equation) models from the time series. This method is here not used to find a model for synthesis of speech, but can be used to characterize certain features of the time series, e. g., for segmentation of a speech signal.

3.1 Differential Embedding: Ansatz Library Based Method

Ansatz Library. The framework of reconstructing an ODE model via an Ansatz library [49, 50] based method is illustrated in Fig. 14.

To obtain the general form of an ODE model in the differential embedding space, $\mathbb{R}^3(X, Y, Z)$, where $X = s(t)$ is the measured time series, $Y = \dot{X}$, and $Z = \dot{Y}$ are the successive derivatives, first consider a continuous-time system in $\mathbb{R}^3(x_1, x_2, x_3)$

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \ , \qquad \text{that is} \qquad \begin{aligned} \dot{x}_1 &= f_1(\boldsymbol{x}) \\ \dot{x}_2 &= f_2(\boldsymbol{x}) \\ \dot{x}_3 &= f_3(\boldsymbol{x}) \end{aligned} \ , \tag{20}$$

with $\boldsymbol{x} = [x_1, x_2, x_3]$, and let $s = h(\boldsymbol{x})$ be an observed scalar signal, where $h : \mathbb{R}^3 \to \mathbb{R}$ is a smooth function. The Lie derivative $L_f h(\boldsymbol{x})$ of the function

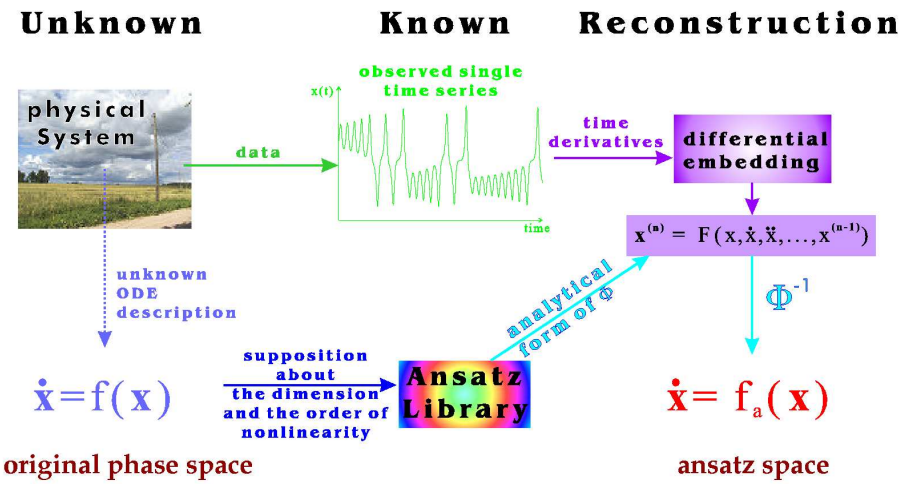


Fig. 14. Framework for reconstructing an ODE model via an Ansatz library:
The original physical system is unknown and only a scalar time series (here a vowel speech signal) can be measured. From this time series a multidimensional object, an embedding, is constructed. The embedding theorems of Takens [17] and Sauer [18] assure us that such an embedding constructed from one single time series preserves the topological properties of the underlying dynamical system. The embedding used here is a differential embedding, where the time series itself and successive derivatives are used.
To find the right model form of the differential model $X^{(n)} = F(X, \dot{X}, \ddot{X}, \ldots, X^{(n-1)})$ we make assumptions about the model form of the original dynamical system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ in the original phase space and then choose the model form of the differential model accordingly. This is done as follows: We make assumptions about the model form, the dimension of the original dynamical system, and the order of nonlinearities. All models $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ with such a model form that can be transformed to a differential model $X^{(n)} = F(X, \dot{X}, \ddot{X}, \ldots, X^{(n-1)})$ are collected in the so-called Ansatz library [49, 50]. To use the inverse Φ^{-1} (see Fig. 13) of the transformation between the original dynamical model and the differential model, we further require that the map Φ is one-to-one. The concatenation of all possible differential models is then used as Ansatz for estimating the model from the time series. If the model falls into the class of models that were considered in the original phase space, the model can be re-transformed to an Ansatz model $\dot{\boldsymbol{x}} = \boldsymbol{f}_a(\boldsymbol{x})$ that is topologically equivalent to the original dynamical system and typically of 'simpler' model form than the differential model (see [50]).

$h(\boldsymbol{x})$ with respect to $f(\boldsymbol{x})$ is defined as

$$L_f h(\boldsymbol{x}) = \sum_{k=1}^{3} f_k(\boldsymbol{x}) \frac{\partial h(\boldsymbol{x})}{\partial x_k} \tag{21}$$

and recursively for the higher-order derivatives $L_f^j h(\boldsymbol{x}) = L_f \left(L_f^{j-1} h(\boldsymbol{x}) \right)$. Using successive Lie derivatives we can build a model from the scalar signal s as follows

$$\begin{aligned} X &= s = h(\boldsymbol{x}) \,, \\ Y &= L_f h(\boldsymbol{x}) \,, \\ Z &= L_f^2 h(\boldsymbol{x}) \,. \end{aligned} \tag{22}$$

The phase portrait can thus be reconstructed in the differential space $\mathbb{R}^3(X, Y, Z)$. With these coordinates, a model can be obtained from the recorded scalar signal via a global modeling procedure. A *general form for a differential model* $\mathcal{D}$ is given by

$$\begin{aligned} \dot{X} &= Y \ , \\ \dot{Y} &= Z \ , \\ \dot{Z} &= F(X, Y, Z, \alpha_n) = \sum_{n=1}^{N_\alpha} \alpha_n P_n \ , \end{aligned} \tag{23}$$

where α_n are the coefficients of the model function F to be estimated and P_n are the monomials $X^i Y^j Z^k$ [52]. The indices (i, j, k) for monomials may also be negative, yielding a model with rational monomials. System (23) is called the *differential model* [52], and its parameters can be obtained using a least square procedure, such as singular value decomposition (SVD) [53].

Here we do not choose the set of monomials P_n in (23) as truncated Taylor series expansion. Instead we use the Ansatz library approach introduced in [49, 50] where only monomials are considered that correspond to an underlying dynamical system in a three dimensional phase space $\mathbb{R}^3(x_1, x_2, x_3)$, where $x_1 = s(t)$ is the observable and $x_2 = x_2(t)$ and $x_3 = x_3(t)$ are unobserved state space variables.

The first library made of six Ansatz systems $\mathcal{A}_l$ (with indices $l = 1, 2, 3, 18, 19, 21$, cf. Table 1) for defining the structure of 3D differential models was presented in [49]. An extended Ansatz Library of systems of ODEs in a three dimensional phase space was derived in [54] for the case when the right hand sides can be written as polynomials containing up to second order non-linearities. We briefly detail how this library was built.

A three dimensional system of ODEs with the right hand sides containing polynomials with up to second order non-linearities can be written in a general form as

$$\begin{aligned} \dot{x}_1 = \sum_{i=0}^{9} a_i \, Q_i \ , \quad \dot{x}_2 = \sum_{i=0}^{9} b_i \, Q_i \ , \quad \dot{x}_3 = \sum_{i=0}^{9} c_i \, Q_i \ , \quad \text{with} \\ Q = \{1, x_1, x_2, x_3, x_1^2, x_1 x_2, x_1 \, x_3, x_2^2, x_2 \, x_3, x_3^2\} \,. \end{aligned} \tag{24}$$

To derive the Ansatz library in [54] we restrict the terms of the differential model to the set of all monomials of the form $X^i Y^j Z^k$, where i, j and k are integers, positive or negative. Given that the order of the differential equations is interchangeable, we fix the x_1-variable as the observable in all cases, i.e. $x_1 = s$, to obtain a set of non-redundant libraries. We then find which model structures allow us to invert the maps Φ_l to express the coefficients α_n of the differential models $\mathcal{D}_l$ in (23), in terms of the coefficients a_i, b_j, c_k of the second order system equation (24). Through this process, we find which coefficients a_i, b_j, c_k in (24) must be zero for the individual differential model structures. In the end we obtain a set of ODEs containing a limited number of terms for which the coefficients a_i, b_j, c_k are non-zero. Note that we do not use data to build the library.

The library for the case of second order non-linearities consists of 26 such model structures listed in Table 1. In Table 2, the monomials involved in the differential models D_l corresponding to the 26 Ansatz A_l reported in Table 1 are listed. The models in Tables 1 and 2 are general forms. Systems with some of the coefficients a_i, b_j, c_k equal to zero can yield the same differential model, where some of the coefficients α_n can also be zero. To be more explicit, let us explain an example (see also [50]). The systems

$$\begin{aligned} \dot{x}_1 &= a_0 + a_1 x_1 + a_5 x_1 x_2 \ , \\ \dot{x}_2 &= b_0 + b_1 x_1 + b_2 x_2 + b_6 x_1 x_3 \ , \\ \dot{x}_3 &= c_0 + c_2 x_2 + c_3 x_3 \ , \end{aligned} \tag{25}$$

and 13 sub-systems with some of the coefficients a_i, b_j, c_k of (25) equal to zero, as well as the following system

$$\begin{aligned} \dot{x}_1 &= a_0 + a_1 x_1 + a_5 x_1 x_2 \ , \\ \dot{x}_2 &= b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ , \\ \dot{x}_3 &= c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3 + c_5 x_1 x_2 \ , \end{aligned} \tag{26}$$

and 155 sub-systems with some of the coefficients a_i, b_j, c_k of (26) equal to zero yield exactly the same differential model

$$\begin{aligned} \dot{X} &= Y \ , \\ \dot{Y} &= Z \ , \\ \dot{Z} &= \alpha_1 + \alpha_6 X + \alpha_7 X^2 + \alpha_{21} Y + \alpha_{25} \frac{Y}{X} + \alpha_{26} XY + \alpha_{35} \frac{Y^2}{X^2} \\ &\quad + \alpha_{36} \frac{Y^2}{X} + \alpha_{44} \frac{Y^3}{X^2} + \alpha_{52} Z + \alpha_{55} \frac{Z}{X} + \alpha_{67} \frac{YZ}{X} \ . \end{aligned} \tag{27}$$

This means, that 170 possible systems with up to second order nonlinearities correspond to one and the same form of differential model (27) and therefore could have the same embedding. Only those systems of the 170 possible systems that correspond to a differential model (27) with exactly the same set and values of the coefficients α_i have the same embedding. All such systems generate the same time series and are therefore topologically equivalent. All non-equivalent systems have at least one different coefficient α_i.

Table 1. Ansatz library for systems of ODEs with up to quadratic non-linearities. Each line represents one general system of ODEs which can be represented as a differential model in the form of (23). An '$\star$' in the table indicates that the corresponding coefficient (a_i, b_j, c_k) from the general system (24) is present in the Ansatz-model. The coefficients with blank entries are zero.

Ansatz	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
$\mathcal{A}_1$	$\star$	$\star$	$\star$		$\star$						$\star$	$\star$	$\star$		$\star$	$\star$	$\star$	$\star$			$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
$\mathcal{A}_2$	$\star$	$\star$	$\star$		$\star$						$\star$	$\star$	$\star$	$\star$	$\star$	$\star$		$\star$			$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
$\mathcal{A}_3$			$\star$								$\star$	$\star$	$\star$		$\star$	$\star$		$\star$	$\star$		$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
$\mathcal{A}_4$	$\star$	$\star$	$\star$	$\star$	$\star$						$\star$	$\star$	$\star$	$\star$	$\star$						$\star$	$\star$	$\star$	$\star$	$\star$					
$\mathcal{A}_5$	$\star$	$\star$	$\star$	$\star$	$\star$						$\star$	$\star$			$\star$						$\star$	$\star$			$\star$		$\star$			
$\mathcal{A}_6$	$\star$	$\star$	$\star$	$\star$	$\star$						$\star$	$\star$			$\star$						$\star$	$\star$			$\star$	$\star$				
$\mathcal{A}_7$	$\star$	$\star$	$\star$	$\star$	$\star$		$\star$				$\star$	$\star$			$\star$						$\star$	$\star$			$\star$					
$\mathcal{A}_8$	$\star$	$\star$	$\star$	$\star$	$\star$					$\star$	$\star$	$\star$			$\star$										$\star$					
$\mathcal{A}_9$	$\star$	$\star$	$\star$	$\star$	$\star$					$\star$	$\star$	$\star$			$\star$							$\star$								
$\mathcal{A}_{10}$	$\star$	$\star$			$\star$				$\star$				$\star$		$\star$	$\star$								$\star$			$\star$			
$\mathcal{A}_{11}$	$\star$	$\star$			$\star$				$\star$			$\star$	$\star$			$\star$								$\star$			$\star$			
$\mathcal{A}_{12}$	$\star$	$\star$			$\star$				$\star$		$\star$		$\star$			$\star$								$\star$			$\star$			
$\mathcal{A}_{13}$	$\star$	$\star$		$\star$	$\star$				$\star$		$\star$		$\star$											$\star$			$\star$			
$\mathcal{A}_{14}$	$\star$	$\star$		$\star$	$\star$				$\star$							$\star$								$\star$			$\star$			
$\mathcal{A}_{15}$	$\star$	$\star$		$\star$	$\star$				$\star$						$\star$									$\star$			$\star$			
$\mathcal{A}_{16}$	$\star$	$\star$		$\star$	$\star$				$\star$			$\star$												$\star$			$\star$			
$\mathcal{A}_{17}$									$\star$				$\star$			$\star$								$\star$			$\star$			$\star$
$\mathcal{A}_{18}$									$\star$				$\star$			$\star$			$\star$					$\star$			$\star$			
$\mathcal{A}_{19}$	$\star$	$\star$			$\star$	$\star$					$\star$	$\star$	$\star$		$\star$	$\star$	$\star$	$\star$			$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
$\mathcal{A}_{20}$	$\star$	$\star$			$\star$	$\star$					$\star$	$\star$	$\star$	$\star$	$\star$	$\star$		$\star$			$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
$\mathcal{A}_{21}$						$\star$					$\star$	$\star$	$\star$		$\star$	$\star$		$\star$	$\star$		$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
$\mathcal{A}_{22}$	$\star$	$\star$			$\star$	$\star$	$\star$				$\star$	$\star$			$\star$						$\star$	$\star$			$\star$		$\star$			
$\mathcal{A}_{23}$	$\star$	$\star$			$\star$	$\star$	$\star$				$\star$	$\star$			$\star$						$\star$	$\star$			$\star$	$\star$				
$\mathcal{A}_{24}$	$\star$	$\star$			$\star$	$\star$	$\star$				$\star$	$\star$			$\star$						$\star$	$\star$		$\star$	$\star$					
$\mathcal{A}_{25}$	$\star$	$\star$			$\star$	$\star$	$\star$				$\star$	$\star$			$\star$						$\star$	$\star$	$\star$		$\star$					
$\mathcal{A}_{26}$	$\star$	$\star$		$\star$	$\star$	$\star$	$\star$				$\star$	$\star$			$\star$						$\star$	$\star$			$\star$					

Table 2. Monomials of the differential models corresponding to the 26 Ansatz reported in Table 1. A '⋆' indicates that the monomial is present in the differential model.

monomial \ Ansatz		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	1	⋆	⋆	⋆	⋆						⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆	⋆						
2	$\frac{1}{X^4}$																			⋆							
3	$\frac{1}{X^3}$																			⋆	⋆						
4	$\frac{1}{X^2}$																			⋆	⋆						
5	$\frac{1}{X}$	⋆																		⋆	⋆						
6	X	⋆	⋆	⋆	⋆	⋆	⋆				⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆	⋆	⋆			⋆	⋆	
7	X^2	⋆	⋆	⋆	⋆	⋆	⋆			⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆	⋆	⋆	⋆	⋆	⋆	⋆	
8	X^3	⋆	⋆	⋆		⋆	⋆				⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆	⋆	⋆	⋆	⋆	⋆	⋆	
9	X^4	⋆	⋆						⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆	⋆	⋆	⋆	⋆			
10	X^5	⋆	⋆																		⋆						
11	X^6	⋆	⋆																								
12	X^7	⋆	⋆																								
13	X^8		⋆																								
14	$\frac{1}{Y}$			⋆																							
15	$\frac{X}{Y}$			⋆																							
16	$\frac{X^2}{Y}$			⋆																		⋆					
17	$\frac{X^3}{Y}$			⋆																		⋆					
18	$\frac{X^4}{Y}$			⋆																		⋆					
19	$\frac{X^5}{Y}$																					⋆					
20	$\frac{X^6}{Y}$																					⋆					
21	Y	⋆	⋆	⋆	⋆			⋆	⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆
22	$\frac{Y}{X^4}$																			⋆							
23	$\frac{Y}{X^3}$																			⋆	⋆						
24	$\frac{Y}{X^2}$																			⋆	⋆						
25	$\frac{Y}{X}$	⋆				⋆	⋆		⋆	⋆	⋆	⋆			⋆	⋆	⋆			⋆	⋆				⋆		
26	XY	⋆	⋆	⋆	⋆	⋆	⋆	⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆	⋆	⋆
27	X^2Y	⋆	⋆	⋆		⋆		⋆			⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆
28	X^3Y	⋆	⋆																		⋆						
29	X^4Y	⋆	⋆																								
30	X^5Y	⋆	⋆																								
31	X^6Y		⋆																								
32	Y^2	⋆	⋆	⋆	⋆			⋆	⋆		⋆		⋆	⋆		⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆
33	$\frac{Y^2}{X^4}$																			⋆							
34	$\frac{Y^2}{X^3}$																			⋆	⋆						
35	$\frac{Y^2}{X^2}$																			⋆	⋆		⋆	⋆	⋆	⋆	
36	$\frac{Y^2}{X}$	⋆				⋆	⋆		⋆	⋆	⋆	⋆			⋆	⋆	⋆			⋆	⋆	⋆			⋆		
37	XY^2	⋆	⋆	⋆																	⋆						
38	X^2Y^2	⋆	⋆																								
39	X^3Y^2	⋆	⋆																								
40	X^4Y^2		⋆																								

Table 2. continued

	monomial \ Ansatz	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
41	Y^3	⋆	⋆	⋆																							
42	$\frac{Y^3}{X^4}$																			⋆							
43	$\frac{Y^3}{X^3}$																			⋆	⋆						
44	$\frac{Y^3}{X^2}$																			⋆	⋆	⋆	⋆	⋆	⋆	⋆	
45	$\frac{Y^3}{X}$	⋆																			⋆						
46	$X\,Y^3$	⋆	⋆																								
47	$X^2\,Y^3$		⋆																								
48	Y^4		⋆																								
49	$\frac{Y^4}{X^4}$																			⋆							
50	$\frac{Y^4}{X^3}$																				⋆						
51	$\frac{Y^4}{X}$	⋆																									
52	Z	⋆	⋆	⋆	⋆	⋆	⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆			⋆		
53	$\frac{Z}{X^3}$																			⋆							
54	$\frac{Z}{X^2}$																			⋆	⋆						
55	$\frac{Z}{X}$	⋆																		⋆	⋆		⋆	⋆	⋆	⋆	
56	$X\,Z$	⋆	⋆	⋆	⋆	⋆	⋆		⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	
57	$X^2\,Z$	⋆	⋆																		⋆						
58	$X^3\,Z$	⋆	⋆																								
59	$X^4\,Z$		⋆																								
60	$\frac{Z}{Y}$			⋆				⋆																			⋆
61	$\frac{X\,Z}{Y}$			⋆				⋆														⋆					⋆
62	$\frac{X^2\,Z}{Y}$			⋆				⋆														⋆					⋆
63	$\frac{X^3\,Z}{Y}$							⋆														⋆					⋆
64	$Y\,Z$	⋆	⋆	⋆																	⋆						
65	$\frac{Y\,Z}{X^3}$																			⋆							
66	$\frac{Y\,Z}{X^2}$																			⋆	⋆						
67	$\frac{Y\,Z}{X}$	⋆				⋆	⋆		⋆	⋆	⋆	⋆			⋆	⋆	⋆			⋆	⋆	⋆	⋆	⋆	⋆	⋆	
68	$X\,Y\,Z$	⋆	⋆																								
69	$X^2\,Y\,Z$		⋆																								
70	$Y^2\,Z$		⋆																								
71	$\frac{Y^2\,Z}{X^3}$																			⋆							
72	$\frac{Y^2\,Z}{X^2}$																				⋆						
73	$\frac{Y^2\,Z}{X}$	⋆																									
74	Z^2		⋆																								
75	$\frac{Z^2}{X^2}$																			⋆							
76	$\frac{Z^2}{X}$	⋆																			⋆						
77	$\frac{Z^2}{Y}$			⋆				⋆										⋆	⋆			⋆					⋆

Note that the equations in this library are also referred to as "jerky dynamics" in the literature. Attempts to build a complete jerky dynamics library were presented in [55], but that library was able to capture only a part of our list.

Our objective is to select the differential model $\mathcal{D}_l$ that best captures the dynamics under investigation. In order to do this, we start with the structure resulting from the concatenation of the 26 differential models $\{\mathcal{D}_l\}_{l=1}^{26}$ which reads as follows:

$$
\begin{aligned}
&\dot{X} = Y \ , \\
&\dot{Y} = Z \ , \\
&\dot{Z} = \alpha_1 + \alpha_2 \tfrac{1}{X^4} + \alpha_3 \tfrac{1}{X^3} + \alpha_4 \tfrac{1}{X^2} + \alpha_5 \tfrac{1}{X} + \alpha_6 X + \alpha_7 X^2 + \alpha_8 X^3 + \\
&\quad \alpha_9 X^4 + \alpha_{10} X^5 + \alpha_{11} X^6 + \alpha_{12} X^7 + \alpha_{13} X^8 + \alpha_{14} \tfrac{1}{Y} + \alpha_{15} \tfrac{X}{Y} + \\
&\quad \alpha_{16} \tfrac{X^2}{Y} + \alpha_{17} \tfrac{X^3}{Y} + \alpha_{18} \tfrac{X^4}{Y} + \alpha_{19} \tfrac{X^5}{Y} + \alpha_{20} \tfrac{X^6}{Y} + \alpha_{21} Y + \alpha_{22} \tfrac{Y}{X^4} + \\
&\quad \alpha_{23} \tfrac{Y}{X^3} + \alpha_{24} \tfrac{Y}{X^2} + \alpha_{25} \tfrac{Y}{X} + \alpha_{26} X Y + \alpha_{27} X^2 Y + \alpha_{28} X^3 Y + \\
&\quad \alpha_{29} X^4 Y + \alpha_{30} X^5 Y + \alpha_{31} X^6 Y + \alpha_{32} Y^2 + \alpha_{33} \tfrac{Y^2}{X^4} + \alpha_{34} \tfrac{Y^2}{X^3} + \\
&\quad \alpha_{35} \tfrac{Y^2}{X^2} + \alpha_{36} \tfrac{Y^2}{X} + \alpha_{37} X Y^2 + \alpha_{38} X^2 Y^2 + \alpha_{39} X^3 Y^2 + \\
&\quad \alpha_{40} X^4 Y^2 + \alpha_{41} Y^3 + \alpha_{42} \tfrac{Y^3}{X^4} + \alpha_{43} \tfrac{Y^3}{X^3} + \alpha_{44} \tfrac{Y^3}{X^2} + \alpha_{45} \tfrac{Y^3}{X} + \\
&\quad \alpha_{46} X Y^3 + \alpha_{47} X^2 Y^3 + \alpha_{48} Y^4 + \alpha_{49} \tfrac{Y^4}{X^4} + \alpha_{50} \tfrac{Y^4}{X^3} + \alpha_{51} \tfrac{Y^4}{X} + \\
&\quad \alpha_{52} Z + \alpha_{53} \tfrac{Z}{X^3} + \alpha_{54} \tfrac{Z}{X^2} + \alpha_{55} \tfrac{Z}{X} + \alpha_{56} X Z + \alpha_{57} X^2 Z + \\
&\quad \alpha_{58} X^3 Z + \alpha_{59} X^4 Z + \alpha_{60} \tfrac{Z}{Y} + \alpha_{61} \tfrac{X Z}{Y} + \alpha_{62} \tfrac{X^2 Z}{Y} + \alpha_{63} \tfrac{X^3 Z}{Y} + \\
&\quad \alpha_{64} Y Z + \alpha_{65} \tfrac{Y Z}{X^3} + \alpha_{66} \tfrac{Y Z}{X^2} + \alpha_{67} \tfrac{Y Z}{X} + \alpha_{68} X Y Z + \\
&\quad \alpha_{69} X^2 Y Z + \alpha_{70} Y^2 Z + \alpha_{71} \tfrac{Y^2 Z}{X^3} + \alpha_{72} \tfrac{Y^2 Z}{X^2} + \alpha_{73} \tfrac{Y^2 Z}{X} + \\
&\quad \alpha_{74} Z^2 + \alpha_{75} \tfrac{Z^2}{X^2} + \alpha_{76} \tfrac{Z^2}{X} + \alpha_{77} \tfrac{Z^2}{Y} \ .
\end{aligned}
\tag{28}
$$

Because this differential model only contains terms leading to a 3D Ansatz with up to quadratic nonlinearities, the presence of spurious terms in this model structure is already greatly reduced. Note, that the Ansatz library approach does not require the knowledge of the order of the non-linearity in the system investigated. Given that our procedure can eliminate spurious model terms that we might obtain, if the order is not known, one can use a library built with an order of non-linearities higher than the one expected for the system. A library for the case of polynomials containing up to third order non-linearities was derived in [54], and libraries for 4th and higher order non-linearities can be derived analogously. Here we use the simplest case of the library obtained for the second order non-linearities. The extension to libraries that capture higher order non-linearities is postponed to future work.

Note, that involving a higher order of nonlinearities would increase the number of candidate terms considerably when the concatenation of all possible models is used as detailed here. For higher dimensions, the numerical estimation of the derivatives may also become problematic. A more favorable approach is to use candidate models $\mathcal{D}_l$ separately to reduce the number of involved terms as done in [49]. Nevertheless, since the terms used here are fractional, the number of situations which can be captured for a given order of nonlinearity is significantly increased as compared to the polynomial expansion used in [52].

Modeling of Vowels. To estimate a global model, the differential embedding from the time series $s(n)$ of the recording of the vowel /o/ (see Fig. 15a) is constructed as shown in Fig. 15b. The derivative $\dot{s}(n)$ of the time series $s(n)$ is estimated according to

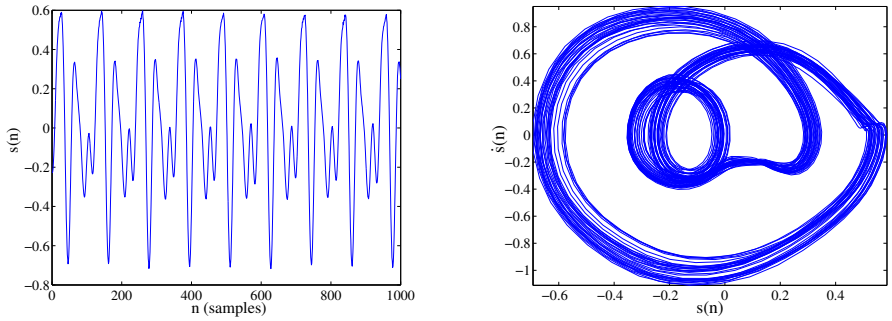
$$\dot{s}(n) = \frac{1}{12\,\delta t}\left(8\left(s(n+1) - s(n-1)\right) - \left(s(n+2) - s(n-2)\right)\right) \quad . \tag{29}$$

The second and third derivatives are then computed from the first and second derivatives, respectively. Then the model of the form (28) is estimated using a least square algorithm, which here is SVD. For our present task, we set the coefficients α_i, that correspond to monomials $\frac{\star}{Y}$ to zero, because such terms make numerical integration very unstable and represent only a highly restricted class of original dynamical systems. For the modeling of the vowel /o/ this yields the following differential model

$$\begin{aligned}
\dot{X} &= Y \ , \\
\dot{Y} &= Z \ , \\
\dot{Z} &= -127.7 + \tfrac{0.0034}{X^4} - \tfrac{1.35585}{X^2} - \tfrac{19.4471}{X} - 477.915\,X - 1100.65\,X^2 - \\
&\quad 1597.75\,X^3 - 1436.2\,X^4 - 734.386\,X^5 - 154.263\,X^6 + 20.1431\,X^7 + \\
&\quad 10.8221\,X^8 - 2304.92\,Y - \tfrac{0.26133\,Y}{X^4} - \tfrac{7.45016\,Y}{X^3} - \tfrac{89.0811\,Y}{X^2} - \\
&\quad \tfrac{583.536\,Y}{X} - 5723.88\,X\,Y - 9077.48\,X^2\,Y - 9134.39\,X^3\,Y - \\
&\quad 5627.95\,X^4\,Y - 1935.58\,X^5\,Y - 285.133\,X^6\,Y + 2044.41\,Y^2 + \\
&\quad \tfrac{3.06178\,Y^2}{X^4} + \tfrac{52.5805\,Y^2}{X^3} + \tfrac{339.683\,Y^2}{X^2} + \tfrac{1100.29\,Y^2}{X} + 2386.52\,X\,Y^2 + \\
&\quad 1852.23\,X^2\,Y^2 + 893.248\,X^3\,Y^2 + 191.93\,X^4\,Y^2 + 6084.05\,Y^3 + \\
&\quad \tfrac{28.7058\,Y^3}{X^4} + \tfrac{390.125\,Y^3}{X^3} + \tfrac{1951.32\,Y^3}{X^2} + \tfrac{4750.06\,Y^3}{X} + 3962.2\,X\,Y^3 + \\
&\quad 1042.75\,X^2\,Y^3 + 434.01\,Y^4 - \tfrac{24.9278\,Y^4}{X^4} - \tfrac{113.943\,Y^4}{X^3} + \tfrac{541.889\,Y^4}{X} + \\
&\quad 613.845\,Z + \tfrac{1.24638\,Z}{X^3} + \tfrac{23.2783\,Z}{X^2} + \tfrac{168.273\,Z}{X} + 1252.69\,X\,Z + \\
&\quad 1452.6\,X^2\,Z + 887.493\,X^3\,Z + 218.578\,X^4\,Z + 1148.8\,Y\,Z + \\
&\quad \tfrac{3.82964\,Y\,Z}{X^3} + \tfrac{70.656\,Y\,Z}{X^2} + \tfrac{445.685\,Y\,Z}{X} + 1234.68\,X\,Y\,Z + \\
&\quad 455.848\,X^2\,Y\,Z + 1434.85\,Y^2\,Z + \tfrac{135.137\,Y^2\,Z}{X^3} + \tfrac{1035.42\,Y^2\,Z}{X^2} + \\
&\quad \tfrac{2253.09\,Y^2\,Z}{X} + 55.0741\,Z^2 + \tfrac{3.1996\,Z^2}{X^2} + \tfrac{26.5625\,Z^2}{X} \quad .
\end{aligned} \tag{30}$$

The original time signal and the signal generated by integrating this model are shown in Fig. 16 (a), the according magnitude spectra are depicted in Fig. 16 (b), and the embedding of the generated signal is shown in Fig. 16 (c).

When comparing the time series of the original recording of the vowel /o/ and the integrated model in Fig. 16(a) we see a similar structure. The generated signal of the vowel /o/ sounds very unnatural, though. When looking at the power spectra in Fig. 16(b) we see, that the low frequency parts are similar, but the high frequency parts are completely missing. Also the embedding in Fig. 16(c) has only a similar structure when compared to the embedding of the original recording in Fig. 15. The embedding of the generated sound is periodic, while the original sound is not strictly periodic. A non-periodic, or even chaotic behavior of the model could be achieved by changing some of the coefficients slightly. Such a change would have to be done very carefully since this model is



(a) (b)

Fig. 15. (a) Time series and (b) differential embedding of the recorded signal for the vowel /o/.

very sensitive to tiny changes and gets easily numerically unstable and, therefore, such investigations are postponed for future work. One way to reduce the model complexity and to enhance the quality would be to use model (30) as basis for some pruning procedure. We currently develop a Genetic Algorithm that fulfills such a task.

3.2 Delay Embedding: Genetic Algorithm based Method

Genetic Algorithm. A genetic algorithm (GA) [56, 57] is a search algorithm that is based on natural genetics. A given problem is encoded as an array (population) of artificial strings (chromosomes). In the cases considered here, where an optimization problem has to be solved, the guesses for possible solutions are encoded. The GA is split into two parts: the first one is devoted to estimation of the time delay set and the second part is used for estimating the ODE model. In the model-selection part, different guesses for models are encoded, while in the delay-selection part, possible delay-combinations are encoded into binary strings. These chromosomes can be strings of 1's and 0's. The GA will then manipulate this representation of the solution, but not the solution itself. A GA also must have a criterion for discriminating good from bad solutions according to the fitness measure of these solutions. This criterion is used to guide the evolution towards future generations. In the case considered here, we use a complex criterion composed of different objectives that include stability of the model, topology, and, of course, similarity of the original and the generated time series.

After encoding the problem in a chromosomal manner and finding a discrimination strategy for good solutions, an initial population of encoded solutions is created. This is done by using a random number generator without any prior knowledge of possibly good solutions. For the model-selection part, it is a set

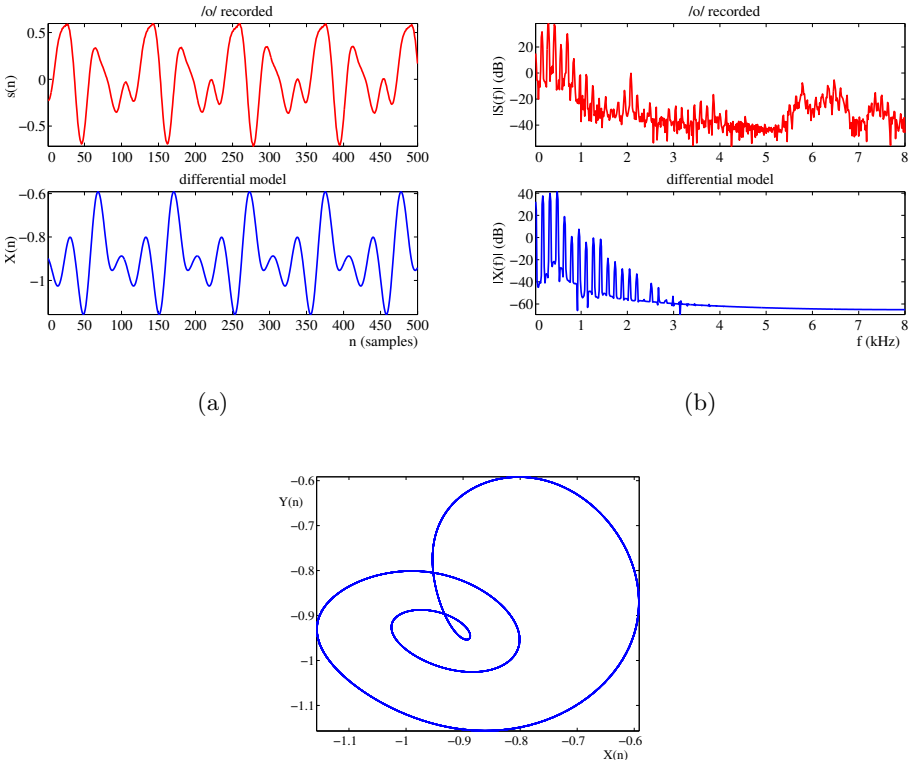


Fig. 16. Time series (a) and DFT spectra (b) of the original vowel signal (top bar) and the signal generated by the estimated differential model in (30) (bottom bar), and (c) embedding of the signal generated by the estimated differential model.

of different ODE models and for the delay-selection part, it is a set of possible delay-combinations.

The evolution of this initial population towards later generations is done by applying genetic operators in an iterative process. The most common genetic operators are (a) selection, (b) recombination, and (c) mutation [56, 57]. Selection allocates greater survival to better individuals. Better solutions are preferred to worse ones. Additional new, possibly better, individuals not present in the original population have to be created. This is done via recombination and mutation. Recombination combines bits of parental solutions to form a better offspring. It combines parental traits in a novel manner. Mutation, on the other hand, modifies a single individual. It is a random walk in the neighborhood of a particular solution.

The GA proposed here is implemented in two parts to solve the given optimization problem. The algorithm is initialized by selecting a first set of delays. If no *a priori* information on the delays is available, a first set of delays can be obtained by visually inspecting the embedded attractors. The algorithm then uses the model-selection-GA to optimize a system of ODEs while the delays are kept fixed. Once the modeling error is minimized for the given delay set, the found model is fixed and the second, delay-selection-GA, is used to optimize over the delays. The process is repeated until the selected model and the delays do not change over a given number of iterations.

The flexibility of GAs allows us to design a strict and, at the same time, complex fitness criterion composed of four different objectives. The modeling error in our algorithm is defined as the least squares error weighed differentially over time to penalize later observations. The penalty for later observations is included because nonlinear systems can only be predicted within the Lyapunov time limit. Since we are working with a single time series, which is also noisy, we typically can only make predictions within the time range which are considerably less then the Lyapunov limit. We further take into account that the dynamics can be different for selected data segments. Our algorithm computes the modeling error from randomly selected data segments and the corresponding segments are integrated. A good model should also be stable when numerically integrated over long time intervals. The algorithm, therefore, automatically discards all models that do not fulfil a long-term stability criterion. Yet, the fourth optimization constraint used is the topological equivalence of the model to the original embedded data. This is implemented by comparing the topology, which we define as the density of the embedded input data with the corresponding integrated data in a two dimensional projection. The nonlinear series generated by the resulting global models not only produce the smallest point-to-point error to the original process, but also recover the topological properties of the embedded data. The GA approach allows us to implement this complex optimization criteria in a straightforward fashion.

GA for Modeling Vowel Signals. The aim here is to find the optimal ODE model for a given time series $s(n)$ and a time delay embedding, with simultaneous optimization of the embedding lags, using a GA for model selection.

We allow models with quadratic order of nonlinearity, leading to a system of three equations with a maximum of 10 coefficients in each equation:

$$\begin{aligned} \dot{x}_1 = \sum_{i=0}^{9} a_i\, Q_i \ , \quad \dot{x}_2 = \sum_{i=0}^{9} b_i\, Q_i \ , \quad \dot{x}_3 = \sum_{i=0}^{9} c_i\, Q_i \ , \quad \text{with} \\ Q = \{1, x_1, x_2, x_3, x_1^2, x_1 x_2, x_1\, x_3, x_2^2, x_2\, x_3, x_3^2\}\,. \end{aligned} \tag{31}$$

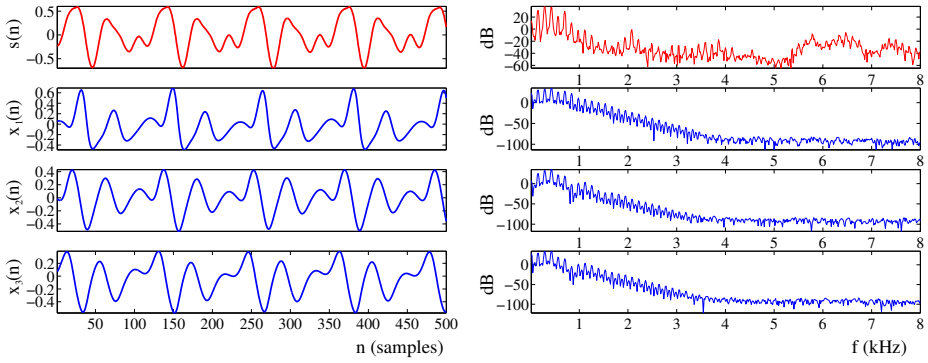
where $x_1(n) = s(n - M_1), x_2(n) = s(n - M_2), x_3(n) = s(n - M_3)$ are delayed versions of the original input-data, and a_i, b_j, c_k are the coefficients, estimated with SVD [53]. Note, that the GA prefers models with as many coefficients a_i, b_j, c_k as possible to be equal to zero. The minimal delay is set to 0 and the maximal one is 120.

To initialize the GA an initial set of delays $M_1 = 0, M_2 = 12, M_3 = 20$ is chosen. The initial population size was set to 100. Our GA increases the population size, if for 2 generations no better individual was found and decreases the population size, if evolution was successful, but never below 100 individuals. Since the number of possible models, together with the number of possible delay combinations is huge, no absolute convergence in reasonable time can be expected. To find a reasonable model, we start the GA a couple of times and then compare the resulting models. All these models are then fed into a new GA. The best model found by the GA for the vowel signal /o/ is

$$\begin{aligned} \dot{x}_1 &= 0.006 - 0.1x_2 + 0.06x_1^2 + 0.05x_1x_2 - 0.15x_1x_3 - 0.22x_2x_3 - 0.16x_3^2 \ , \\ \dot{x}_2 &= -0.011 + 0.08x_1 - 0.06x_3 + 0.05x_1^2 + 0.12x_1x_3 + 0.2x_3^2 \ , \\ \dot{x}_3 &= 0.01 + 0.15x_2 - 0.06x_3 - 0.05x_1^2 - 0.05x_2^2 \ . \end{aligned} \tag{32}$$

This model was numerically integrated and the resulting signals for x_1, x_2 and x_3 as well as the magnitude spectra are depicted in Fig. 17, the differential embeddings are given in Fig. 18. During numerical integration the number of used digits was fixed to 6. A nonlinear dynamical system is very sensitive to small changes in the initial conditions. Trajectories with slightly different initial conditions can exponentially diverge after a few cycles. Fixing the number of used digits during numerical integration adds a random component to the system.

The reconstructed time-series as shown in Fig. 17 (a) on the lower three plots of $x_1(n)$, $x_2(n)$, and $x_3(n)$ look somehow similar to the time-series of the original recording in the upper plot. When looking at the magnitude spectra in Fig. 17 (b) missing parts in the higher frequency ranges in the lower three plots of the reconstructed signal-components can immediately be seen. The embeddings in Fig. 18 look very similar to the embedding of the original signal. To improve this model, a more general Ansatz in (31) could be used or the GA could be restricted to only models that are also part of the Ansatz library.



(a) (b)

Fig. 17. Time series (a) and DFT spectra (b) of the original vowel signal $s(n)$, and the state variables $x_1(n), x_2(n)$, and $x_3(n)$ of the ODE model found by the GA.

3.3 DDE Models

We also can use a GA to estimate a model of a given signal in form of a DDE,

$$\dot{x} = F(x_1, x_2, \ldots) \ , \tag{33}$$

where $x_i = x(n - M_i)$ and $M_i \in \mathbb{N}_0$. The discrimination strategy for better models are obtained by minimizing the error of the model, i.e.,

$$f \equiv \frac{\left\langle [\dot{x} - F(x_1, x_2, \ldots)]^2 \right\rangle}{\sigma_{\dot{x}}^2} \ , \tag{34}$$

where $\sigma_{\dot{x}}^2$ is the variance of the time series $\dot{x}$. The coefficients of the models are numerically estimated by a least square algorithm, which is in our case a singular value decomposition (SVD) [53]. The principal idea of minimizing a function using a GA can be found in [56].

The GA works in two steps, the delay-selection and the model-selection part, which can be described as follows: The GA depends on four modeling parameters, (i) the number of delays, n_τ, (ii) the maximal number of coefficients in the models, N_c, (iii) the order of nonlinearity, m, and (iv) the initial population size, N_p. Then a first set of delay(s) and the initial population of models are generated with a random number generator. The model-selection GA is applied and is stopped when the modeling error does not change for 5 iteration steps. The best model is selected and, starting from the initial population of delays, the delay-selection GA is applied and is stopped when the modeling error does not change for 5 iteration steps. Then the model-selection GA is applied again starting from the best models of former runs. When the modeling error does not change for 5 iteration steps again, the delay-selection GA is applied once, and

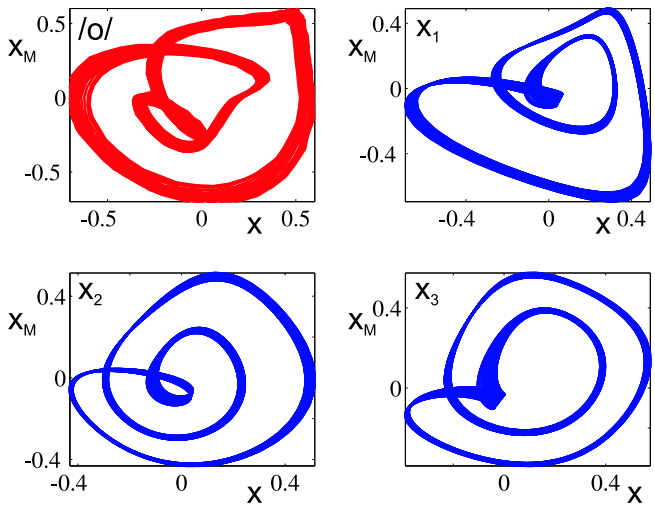


Fig. 18. Differential embeddings of the signals in Fig. 17.

so on. This alternative run of the two codes is stopped, when the modeling error remains constant for both parts of the GAs.

The choice of the population-size is a critical point for a fast convergence to the global minimum of the solution space and should be related to the number of possible combinations of solutions. After some runs of our code, we found empirically that 0.1% of all possible combinations of solutions is a good choice for the population size. Furthermore, we do not keep the population size constant, but change it dynamically during a run. For instance, when the new generation has a better winner which is the same as in the former generation, the population size can be reduced. This could mean that the solution is possibly trapped in a local minimum. With a larger population size the escape from local minima towards the global one is accelerated.

To find a good DDE model to characterize speech signals we first run this GA on a set of 1000 randomly chosen speech signal segments of 1200 data points each. Here we do not aim to find a model that can be used for synthesis of speech, but for characterizing different features of the data. We therefore restrict our search to models with up to five terms and up to three delays where smaller models are preferred in the algorithm.

Our finding is that three-delay models have on average about the same modeling error as two-delay models and therefore we choose to use only two delays for our analysis. Furthermore a three-term model seems to characterize as many features as more term models. Therefore we use the three-term model that is the statistical winner of this run. Note that this model was good for all different kinds of sounds and sound combinations since the signals for this run were randomly chosen from a set of speech signals from different speakers and sentences.

For our further analysis we use the DDE model with two delays,

$$\dot{x} = a_1 x_1 + a_2 x_2 + a_3 x_1 x_2 \ . \tag{35}$$

The delays are adjusted by a global search procedure, a GA (genetic algorithm) for windows of L_W points in the signal such that the least square error of (35) is minimal. The choice of L_W tunes if we want to look at more or less global effects. For example emotional expressions in speech can be better seen if L_W is larger and on the other hand the segmentations into phonemes requires a smaller L_W. For the optimal delays of each window the coefficients $a_{1,2,3}$ are computed directly using SVD (singular value decomposition). Our set of features for classification will then be the delays $M_{1,2}$, the coefficients $a_{1,2,3}$, and the error ρ.

In Fig. 19 the sentences "My dog and my neighbor's cat are hiding under the chair. They are extremely good friends." were analyzed by such a procedure. The window length L_W was 1200 points which corresponds to about 6 characteristic cycles.

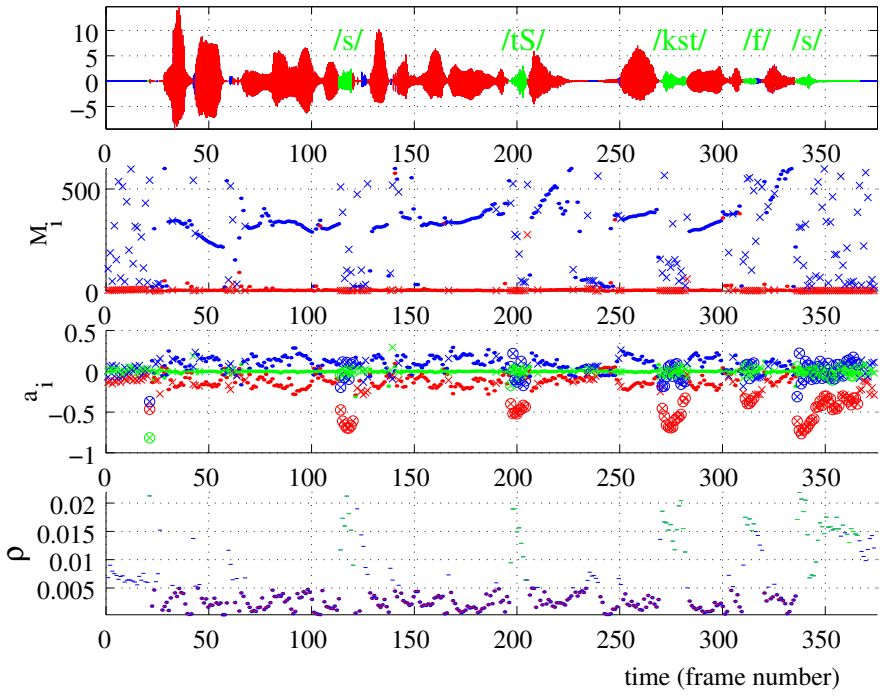


Fig. 19. Time series of the sentences "My dog and my neighbor's cat are hiding under the chair. They are extremely good friends", optimized delays $M_{1,2}$, coefficients $a_{1,2,3}$, and the least square error ρ of (35).

Several things can be seen immediately:

- delays $M_{1,2}$:
  The bigger one of the two delays, M_2 has regions where it is somehow constant for some time. The mean value and its deviation of these regions is characteristic for the speaker. Female speakers have a lower mean value than male speakers. The variance around the mean is characteristic for the speaker. It expresses the melody of speech. There is a direct connection to the fundamental frequency F_0.
  The smaller one of the two delays, M_1 can sometimes jump up to the second delay. This is a characteristics for emotions.
- coefficients $a_{1,2,3}$:
  For harmonic parts of the signal the two linear coefficients have symmetric values, $a_1 \approx -a_2$ and the nonlinear coefficient is $a_3 \approx 0$. This is the case for vowels, nasals and approximants.
  In some regions of the signal one of the coefficients, a_1 has significantly smaller values, the second linear coefficient, a_2 is not correlated to a_2, and the nonlinear coefficient a_3 has nonzero values. This is the characteristics for fricatives and affricates.
- error ρ:
  The error is small for voiced sounds with a harmonic structure – such as vowels, nasals, and approximants – and large for unvoiced sounds, like fricatives and affricates.

Figure 19 shows the speech signal, the delays, the coefficients $a_{1,2,3}$, and the least square error. In the signal plot (top bar) of Fig. 19 the segments highlighted in darker grayshading denote plosives or unvoiced fricatives. They are characterized by a coefficient $a_1 < -0.3$ and an error $\rho > 0.005$. Harmonic sounds are characterized by a low error ($\rho < 0.005$), symmetric linear coefficients ($a_1 \approx -a_2$), and delays that are around the characteristic value for the speaker.

Figure 20 shows the part of Fig. 19 where the word "extremely" is spoken. The previously discussed characteristics are very clear in these plots.

This technique can also be used to segment speech signals into phonemes. For our modeling techniques different samples of certain vowels could be selected by such a DDE model and then fed into our modeling algorithms of Secs. 3.1 and 3.2. This could yield more realistic models than starting from recordings of sustained speech sounds.

4 Summary and Conclusion

The identification of a nonlinear oscillator based on a Takens embedding for the re-generation of stationary speech signals requires – besides an adequate choice of embedding parameters – a careful modeling of the nonlinear function characterizing the dynamics of the signal trajectories in embedding phase space.

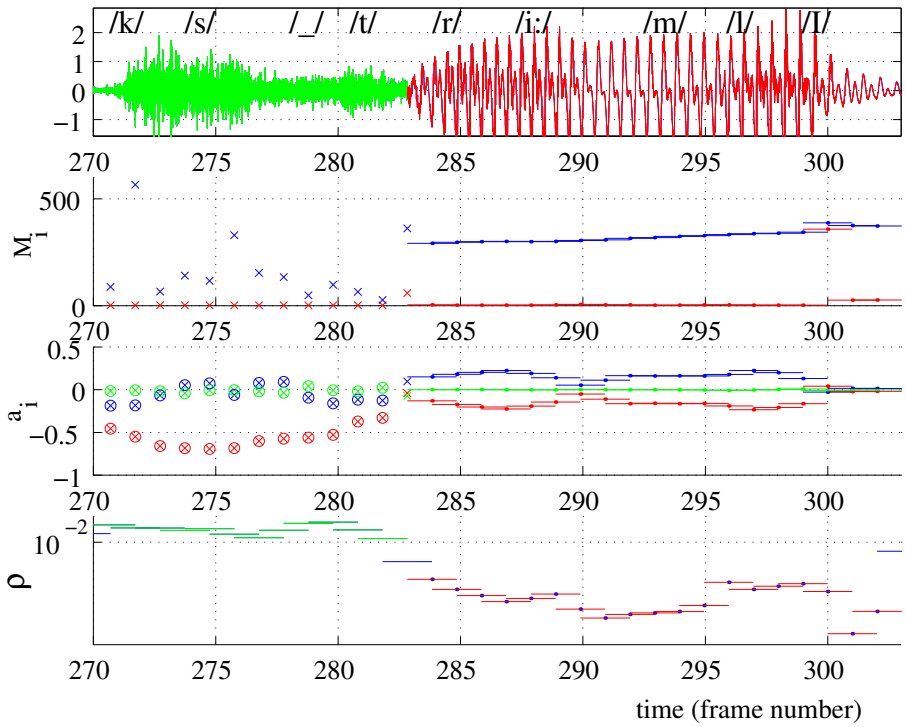


Fig. 20. Part of Fig. 19 where the word “extremely” is spoken.

For the oscillator model based on a time delay embedding some form of regularization of the nonlinear function model used in the oscillator has to be applied. We found that the Bayesian approach to determining weights and regularization parameter of an RBF network is a computationally equivalent and more robust alternative to cross-validation. Besides the automatic determination of adequate regularization the Bayesian approach also gives an accurate estimate for the power of an additive noise-like signal component. An extension of the Bayesian algorithm, the relevance vector machine, additionally allows for pruning of RBF basis functions, thus reducing the complexity of the nonlinear function model at only minimal impairment of prediction accuracy or oscillator stability.

The number of vowel signals that can be stably re-synthesized with the oscillator model is substantially increased when the model is complemented by inverse filtering. Up to now, vowels with a complicated trajectory structure of the full speech signal often could only be stably re-synthesized using a high embedding dimension, a specific set of embedding parameters, or a specific structure of the nonlinear function model. With a simple inverse filtering approach using linear prediction and low-pass filtering and a Bayesian trained RBF network more than half of the vowel signals in our database can be stably re-synthesized using one and the same low-dimensional embedding, since an open loop trajectory is attained for all vowels.

For the additional regeneration of the noise-like component in speech signals we propose the oscillator-plus-noise model, which is able to generate a pitch-synchronous modulation and individual spectral shaping of the noise-like signal component, with the modulation envelope and spectral characteristics automatically learned from the training speech signal.

The parameterization by an RBF model (and all other nonlinear function models referenced in Sect. 2) is, however, not directly related to physical parameters of the speech production process, nor to higher level speech parameters such as fundamental frequency. This fact hinders the use of the oscillator model in a speech synthesizer, since the robust control of fundamental frequency, for example, is still an unsolved problem. Models more closely related to the physical process of speech production are lumped mass-spring-damper models like the two-mass model by Ishizaka and Flanagan [58] and its descendants, which are, however, more difficult to control in terms of higher level speech parameters.

Another step towards a parameterization of the nonlinear oscillator in a physically sensible way is the system modeling by differential equations based on a differential embedding. Here the selection of polynomial terms from an Ansatz library provides the means for a computationally traceable and robust modeling. A three-dimensional differential model for a stationary vowel signal, as given by (30), for example, captures the signal dynamics with a number of 64 parameters.

A further reduction of system complexity is achieved by the application of a genetic algorithm for model structure selection. For modeling a vowel signal the GA with fitness criteria including long-term stability and topological equivalence may choose a model with only 18 parameters, as given by (32), from all possible models based on a certain three dimensional embedding. The GA can

also be used for optimizing low complexity nonlinear models for speech analysis, where, in the example given, the distinction of voiced and unvoiced speech, and the identification of an optimal embedding delay for prediction related to the fundamental frequency is demonstrated.

The benefit due to including inverse filtering, the proper re-generation of the noise-like signal part with an additional non-deterministic system, as well as the identification of different optimal model structures for voiced and unvoiced phonemes and of an optimal embedding delay related to the fundamental period point at the fact that for the application of oscillator models to general purpose speech synthesis – besides robust and elaborate nonlinear function identification methods – a broad spectrum of knowledge from phonetics and speech science is necessary. Based on such knowledge the further development of nonlinear oscillator models for speech analysis and synthesis is on a promising path to being employed as a standard tool for speech technology.

References

1. Kubin, G.: Nonlinear processing of speech. In Kleijn, W.B., Paliwal, K.K., eds.: Speech Coding and Synthesis. Elsevier, Amsterdam etc. (1995) 557–610
2. Kubin, G.: Synthesis and coding of continuous speech with the nonlinear oscillator model. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA (1996) 267–270
3. Kubin, G., Kleijn, W.B.: Time-scale modification of speech based on a nonlinear oscillator model. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Volume 1., Adelaide, South Australia (1994) 453–456
4. Sauer, T.: A noise reduction method for signals from nonlinear systems. Physica D **52** (1992) 193–201
5. Hegger, R., Kantz, H., Matassini, L.: Noise reduction for human speech signals by local projection in embedding spaces. IEEE Transactions on Circuits and Systems **48** (2001) 1454–1461
6. Terez, D.E.: Robust pitch determination using nonlinear state-space embedding. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Volume 1., Orlando (FL), USA (2002) 345–348
7. Mann, I., McLaughlin, S.: A nonlinear algorithm for epoch marking in speech signals using Poincaré maps. In: Proceedings of the European Signal Processing Conference. Volume 2. (1998) 701–704
8. Lindgren, A.C., Johnson, M.T., Povinelli, R.J.: Joint frequency domain and reconstructed phase space features for speech recognition. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Volume 1., Montreal, Quebec, Canada (2004) 533–536
9. Birgmeier, M.: A fully Kalman-trained radial basis function network for nonlinear speech modeling. In: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia (1995) 259–264
10. Kubin, G.: Synthesis and coding of continuous speech with the nonlinear oscillator model. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Volume 1., Atlanta (GA) (1996) 267–270

11. Haas, H., Kubin, G.: A multi-band nonlinear oscillator model for speech. In: Proceedings of the 32nd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA (1998)
12. Mann, I., McLaughlin, S.: Stable speech synthesis using recurrent radial basis functions. In: Proceedings of the European Conference on Speech Communication and Technology. Volume 5., Budapest, Hungary (1999) 2315–2318
13. Narasimhan, K., Príncipe, J.C., Childers, D.G.: Nonlinear dynamic modeling of the voiced excitation for improved speech synthesis. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Phoenix, Arizona (1999) 389–392
14. Rank, E., Kubin, G.: Nonlinear synthesis of vowels in the LP residual domain with a regularized RBF network. In Mira, J., Prieto, A., eds.: Lecture Notes in Computer Science. Volume 2085., Springer (2001) 746–753, part II
15. Mann, I., McLaughlin, S.: Synthesising natural-sounding vowels using a nonlinear dynamical model. Signal Processing **81** (2001) 1743–1756
16. Rank, E.: Application of Bayesian trained RBF networks to nonlinear time-series modeling. Signal Processing **83** (2003) 1393–1410
17. Takens, F.: Detecting strange attractors in turbulence. Lecture Notes in Mathematics **898** (1981) 366
18. Sauer, T., Yorke, J.A., Casdagli, M.: Embedology. Journal of Statistical Physics **65** (1991) 579–616
19. Haykin, S., Príncipe, J.: Making sense of a complex world. IEEE Signal Processing Magazine **15** (1998) 66–81
20. Judd, K., Mees, A.: Embedding as a modeling problem. Physica D **120** (1998) 273–286
21. Bernhard, H.P.: The Mutual Information Function and its Application to Signal Processing. PhD thesis, Vienna University of Technology (1997)
22. Hegger, R., Kantz, H., Schreiber, T.: Practical implementation of nonlinear time series methods: The TISEAN package. CHAOS **9** (1999) 413–435
23. Bernhard, H.P., Kubin, G.: Detection of chaotic behaviour in speech signals using Fraser's mutual information algorithm. In: Proc. 13th GRETSI Symp. Signal and Image Process., Juan-les-Pins, France (1991) 1301–1311
24. Mann, I.: An Investigation of Nonlinear Speech Synthesis and Pitch Modification Techniques. PhD thesis, University of Edinburgh (1999)
25. Rank, E., Kubin, G.: Nonlinear synthesis of vowels in the LP residual domain with a regularized RBF network. In Mira, J., Prieto, A., eds.: Lecture Notes in Computer Science. Volume 2085. Springer (2001) 746–753, part II
26. Li, J., Zhang, B., Lin, F.: Nonlinear speech model based on support vector machine and wavelet transform. In: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03), Sacramento, CA (2003) 259–264
27. Haas, H., Kubin, G.: A multi-band nonlinear oscillator model for speech. In: Proc. 32nd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA (1998)
28. Townshend, B.: Nonlinear prediction of speech. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. (1991) 425–428
29. Tikhonov, A.N., Arsenin, V.Y.: Solutions of Ill-posed Problems. W.H. Winston (1977)
30. Poggio, T., Girosi, F.: A theory of networks for approximation and learning. A.I. Memo 1140, Massachusetts Institute of Technology (1989)
31. Stone, M.: Cross-validation choice and assessment of statistical predictions. Journal of the Royal Statistical Society B **36** (1974) 111–147

32. MacKay, D.J.: Bayesian interpolation. Neural Computation **4** (1992) 415–447
33. MacKay, D.J.: A practical Bayesian framework for backprop networks. Neural Computation **4** (1992) 448–472
34. MacKay, D.J.: The evidence framework applied to classification networks. Neural Computation **4** (1992) 698–714
35. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelyhood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society B **39** (1977) 1–38
36. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research **1** (2001) 211–244
37. Fant, G., Liljencrants, J., Lin, Q.G.: A four parameter model of glottal flow. Quarterly Progress Status Report 4, Speech Transmission Laboratory/Royal Institute of Technology, Stockholm, Sweden (1985)
38. Köppl, H., Kubin, G., Paoli, G.: Bayesian methods for sparse RLS adaptive filters. In: Thirty-Seventh IEEE Asilomar Conference on Signals, Systems and Computers. Volume 2. (2003) 1273–1277
39. Kubin, G., Atal, B.S., Kleijn, W.B.: Performance of noise excitation for unvoiced speech. In: Proc. IEEE Workshop on Speech Coding for Telecommunication, St.Jovite, Québec, Canada (1993) 1–2
40. Holm, S.: Automatic generation of mixed excitation in a linear predictive speech synthesizer. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Volume 6., Atlanta (GA) (1981) 118–120
41. Hermes, D.J.: Synthesis of breathy vowels: Some research methods. Speech Communication **10** (1991) 497–502
42. Skoglund, J., Kleijn, W.B.: On the significance of temporal masking in speech coding. In: Proceedings of the International Conference on Spoken Language Processing. Volume 5., Sydney (1998) 1791–1794
43. Jackson, P.J., Shadle, C.H.: Aero-acoustic modelling of voiced and unvoiced fricatives based on MRI data. In: Proceedings of 5th Speech Production Seminar, Kloster Seeon, Germany (2000) 185–188
44. Jackson, P.J., Shadle, C.H.: Frication noise modulated by voicing, as revealed by pitch-scaled decomposition. Journal of the Acoustic Society of America **108** (2000) 1421–1434
45. Stylianou, Y., Laroche, J., Moulines, E.: High-quality speech modification based on a harmonic + noise model. In: Proceedings of the European Conference on Speech Communication and Technology, Madrid, Spain (1995) 451–454
46. Bailly, G.: A parametric harmonic+noise model. In Keller, E., Bailly, G., Monaghan, A., Terken, J., Huckvale, M., eds.: Improvements in Speech Synthesis. Wiley (2002) 22–38
47. Rank, E., Kubin, G.: An oscillator-plus-noise model for speech synthesis. Speech Communication (2005) Accepted for publication.
48. Lu, H.L., Smith, III, J.O.: Glottal source modeling for singing voice. In: Proc. International Computer Music Conference, Berlin, Germany (2000) 90–97
49. Lainscsek, C., Letellier, C., Schürrer, F.: Ansatz library for global modeling with a structure selection. Physical Review E **64** (2001) 016206:1–15
50. Lainscsek, C., Letellier, C., Gorodnitsky, I.: Global modeling of the Rössler system from the z-variable. Physics Letters A **314(5-6)** (2003) 409–127
51. Judd, K., Mees, A.: On selecting models for nonlinear time series. Physica D **82** (1995) 426–444
52. Gouesbet, G., Letellier, C.: Global vector-field reconstruction by using a multivariate polynomial l_2 approximation on nets. Phys. Rev. E **49** (1994) 4955

53. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical Recipes in C. Cambridge University Press (1990)
54. Lainscsek, C., Gorodnitsky, I.: Ansatz libraries for systems with quadratic and cubic non-linearities. `http://cloe.ucsd.edu/claudia/poster_DD_2002.pdf` (2002)
55. Eichhorn, R., Linz, S., Hänggi, P.: Transformations of nonlinear dynamical systems to jerky motion and its application to minimal chaotic flows. Physical Review E **58 (6)** (1998) 7151–7164
56. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1998)
57. Holland, J.H.: Adaptation in natural and artificial systems. MIT Press (1992)
58. Ishizaka, K., Flanagan, J.L.: Synthesis of voiced sounds from a two-mass model of the vocal cords. Bell Systems Technical Journal **51** (1972) 1233–1267