Oscillator-plus-Noise Modeling of Speech Signals

Erhard Rank

Dissertation

Submitted for consideration for the degree of
Doctor of Engineering Sciences (Dr.techn.)

Vienna University of Technology

Faculty of Electrical Engineering and
Information Technology

Institute of Communications and
Radio-Frequency Engineering

Nov. 2005

Examination Committee:

O. Univ.-Prof. Dr. Wolfgang Mecklenbräuker

Institute of Communications and Radio-Frequency Engineering
Vienna University of Technology
Gußhausstrasse 25
A–1040 Vienna
Austria

Univ.-Prof. Dr. Gernot Kubin

Signal Processing and Speech Communication Laboratory
Graz University of Technology
Inffeldgasse 12
A–8010 Graz
Austria

Thanks

During the work on this thesis I encountered the help of many people in many ways. I would like to thank all of them for their aid.

In particular, I want to thank everyone at the Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology. Beyond my appreciation of the general warm and fruitful ambiance in the signal processing group, I am indebted to Franz Hlawatsch for reviewing part of this thesis, and many thanks go to my roommate Bernhard Wistawel, and to Boris Dortschy for nearly infinite technical and human support over the years. I thank all students involved in the implementation of algorithms for my work. Not to forget, I like to thank Friederike Svejda, Manuela Heigl, Eva Schwab, and Michaela Frech, who keep everything running.

My gratitude for fruitful discussion and professional aid also extends to the people at my current affiliation, the Signal Processing and Speech Communications Laboratory at Graz University of Technology.

My fascination for speech signal processing has been greatly increased by participating in the European Cost action 258 "The Naturalness of Synthetic Speech," and besides all the people taking part in this action, I want to thank the organizers of Cost258, Eric Keller and Brigitte Zellner, as well as Gérard Bailly, who taught me whose side to be on.

Another driving force to my work was the Cost action 277 "Nonlinear Speech Processing," chaired by Marcos Faundez, and my thanks here extend to Steve McLaughlin and Iain Mann.

I further like to thank Renée Fürst, Hannes Pirker, Mike Strickland, the Bonn people (i.p., Gerit, Kalle, Thomas, and Petra), Friedrich Neubarth, Georg Niklfeld, and Michael Pucher, as well as everyone at the Austrian Research Institute for Artificial Intelligence (ÖFAI) and at the Telecommunications Research Center Vienna (ftw.) who supported me and my work.

Cordial thanks go to my parents, Wolfgang and Ilse Rank, and to my children Julian, Caroline, and Jonathan.

More than all others, I want to thank my advisers, Wolfgang Mecklenbräuker and Gernot Kubin, for trusting in me.

This document was typeset in Computer Modern using the LaTeX distribution from Thomas Esser. For phonetic symbols the TIPA font from Rei Fukui was used, and for the facsimilæ from [vK70] the fraktur font from Frank Mittelbach.

Figure plots were generated using *gnuplot* from the *octave* simulation environment, except for fig. 4.33, generated using *Matlab*. Figure drawings have been produced with *xfig*. Specific analyzes were performed using functions from the nonlinear time series analysis package *tisean*, and from the *Edinburgh speech tools*.

Facsimilæ from [vK70] are reproduced with kind permission of Verlag Frommann/Holzboog.

To *Caroline, Julian,* and *Suzie*
In memory of ***Jonathan***

Abstract

In this thesis we examine the autonomous oscillator model for synthesis of speech signals. The contributions comprise an analysis of realizations and training methods for the nonlinear function used in the oscillator model, the combination of the oscillator model with inverse filtering, both significantly increasing the number of 'successfully' re-synthesized speech signals, and the introduction of a new technique suitable for the re-generation of the noise-like signal component in speech signals.

Nonlinear function models are compared in a one-dimensional modeling task regarding their presupposition for adequate re-synthesis of speech signals, in particular considering stability. The considerations also comprise the structure of the nonlinear functions, with the aspect of the possible interpolation between models for different speech sounds. Both regarding stability of the oscillator and the premiss of a nonlinear function structure that may be pre-defined, RBF networks are found a preferable choice. In particular in combination with a Bayesian training algorithm, RBF networks with Gaussian basis functions outperform other nonlinear function models concerning the requirements for the application in the oscillator model.

The application of inverse filtering, in particular linear prediction as a model for speech production, in addition to nonlinear oscillator modeling, allows the oscillator to model an estimated speech source signal as evoked by the oscillatory motion of the vocal folds. The combination of linear prediction inverse filtering and the nonlinear oscillator model is shown to provide a significantly higher number of stably re-synthesized vowel signals, and better spectral reconstruction than the oscillator model applied to the full speech signal. However, for wide-band speech signals the reconstruction of the high-frequency band is still unsatisfactory. With a closer analysis it becomes clear that – while the oscillatory component can now be reproduced satisfactorily – a model for the noise-like component of speech signals is still missing.

Our remedy is to extend the oscillator model by a nonlinear predictor used to re-generate the amplitude modulated noise-like signal component of stationary mixed excitation speech signals (including vowels and voiced fricatives). The resulting 'oscillator-plus-noise' model is able to re-generate vowel signals, as well as voiced fricatives signals with high fidelity in terms of time-domain waveform, signal trajectory in phase space, and spectral characteristics. Moreover, due to the automatic determination of a zero oscillatory component, also unvoiced fricatives are reproduced adequately as the noise-like component only. With one instance of the proposed model all kinds of stationary speech sounds can be re-synthesized, by applying model parameters – i. e., the RBF network weights and linear prediction filter coefficients – learned from a natural speech signal for each sound.

In a first objective analysis of naturalness of the oscillator-plus-noise model generated signals measures for short-term variations in fundamental frequency and amplitude are found to better resemble the measures of the original signal than for the oscillator model only, suggesting an improvement in naturalness.

Contents

1 Introduction 1
1.1 Motivation . . . 1
1.2 Human speech production . . . 2
1.3 Nonlinear modeling of speech signals . . . 3
1.4 Thesis outline . . . 6

2 Current speech synthesis techniques 7
2.1 Articulatory synthesis . . . 7
2.2 Formant synthesis . . . 8
2.3 Glottal waveform generation . . . 8
2.4 Concatenative synthesis . . . 11
2.4.1 Pitch synchronous overlap add (PSOLA) . . . 12
2.4.2 Linear prediction (LP) based synthesis . . . 13
2.4.3 Sinusoidal/harmonic-plus-noise modeling . . . 15
2.4.4 Comparison of prosody manipulation algorithms . . . 15
2.5 Unit selection . . . 16
2.6 Challenges for speech synthesis . . . 17

3 Nonlinear function models 19
3.1 Radial basis function (RBF) networks . . . 20
3.1.1 Pruning centers distant from the training data . . . 25
3.1.2 Regularization . . . 27
3.1.3 Bayesian learning of regularization factor and noise variance . . . 30
3.1.4 The relevance vector machine (RVM) . . . 34
3.2 Other models . . . 36
3.2.1 Locally constant (nearest neighbor) approximation . . . 36
3.2.2 Multi-layer perceptrons . . . 38
3.2.3 Multivariate adaptive regression splines (MARS) . . . 40
3.3 Conclusion . . . 41

4 Speech synthesis with the oscillator model 45
4.1 Time series prediction . . . 45
4.1.1 Embedding dimension . . . 46
4.1.2 Embedding delay . . . 48
4.1.3 Non-uniform embedding . . . 49
4.2 The oscillator model . . . 50
4.3 Comparison of nonlinear function models . . . 51
4.3.1 Choice of embedding and nonlinear function model structure . . . 51
4.3.2 Need for regularization . . . 52
4.3.3 Determining the regularization factor . . . 55

4.3.4 Performance using other nonlinear function models . . . 58
4.3.5 Properties of re-synthesized signals . . . 62
4.4 Combined LP and oscillator model . . . 62
4.4.1 Inverse filtering and source-filter modeling . . . 64
4.4.2 Linear prediction of speech . . . 65
4.4.3 Estimation of the ‘glottal signal’ . . . 67
4.4.4 Synthesis of stationary vowels . . . 70
4.4.5 Cross-validation vs. Bayesian learning . . . 73
4.4.6 Noise variance estimation by Bayesian RBF learning . . . 77
4.5 Non-stationary modeling . . . 79
4.5.1 Interpolation of fundamental frequency . . . 81
4.5.2 Morphing of vowel sounds . . . 84
4.6 Conclusion . . . 86

5 Oscillator-plus-noise model **89**
5.1 Noise-like component in speech signals . . . 90
5.1.1 Objective measures . . . 90
5.1.2 Error signal in models for voiced speech . . . 92
5.1.3 Requirements for natural re-synthesis . . . 93
5.2 Derivation of the oscillator-plus-noise model . . . 94
5.2.1 Properties of the prediction error . . . 94
5.2.2 Synthesis of vowels with noise . . . 96
5.2.3 Analysis of mixed excitation speech signals . . . 101
5.2.4 Oscillator-plus-noise model with a second LP path . . . 103
5.3 Assessment of naturalness . . . 109
5.3.1 Jitter and shimmer . . . 109
5.3.2 Trajectory divergence . . . 113
5.4 Conclusion . . . 116

6 Summary and conclusions **119**
6.1 Contributions . . . 119
6.2 Discussion and potentials . . . 120

A Abbreviations, acronyms, and mathematical symbols **123**
A.1 Abbreviations and acronyms . . . 123
A.2 Mathematical symbols . . . 124

B Bayesian training of an RBF network **125**
B.1 Derivation of the iterative algorithm . . . 125
B.2 Scale-invariance for uniform prior pdfs on logarithmic scale . . . 128

C Example signals **129**

Bibliography **141**

Chapter 1

Introduction

Speech synthesis is a key function in many systems for human-machine communication, particularly in applications like information retrieval, dialog systems, but also in inter-human communication applications, for example in automatic translation. Current speech synthesis systems – though often rated well in terms of intelligibility – still have to be improved concerning naturalness of the synthetic speech signal. This thesis is concerned with a technique to generate synthetic speech signals based on concepts from nonlinear dynamical systems theory and focusses on improving stability of the speech signal synthesis system and naturalness of the synthetic speech signals generated.

1.1 Motivation

Speech production is a nonlinear process whose physics are governed by the equations describing the oscillatory motion of the vocal folds and flow of air through the glottis for voiced speech signals, as well as by the effect of turbulent air flow for noise-like excitation. Consequently, speech analysis and, in particular, speech synthesis should be performed in the framework of the theory of nonlinear dynamical systems to capture the important nonlinear phenomena and to produce naturally sounding synthetic speech signals.

In most state-of-the-art signal generation algorithms for speech synthesis the nonlinear aspect of speech production is greatly neglected and properties inherent to nonlinear systems have to be artificially introduced to the synthesis system. For example, qualities like continuous phase of the generated signal – which is an implicit property of nonlinear dynamical systems – or short-term variations in amplitude and fundamental frequency – which may naturally be generated by a nonlinear oscillating system – have to be introduced artificially to many speech synthesizers, e. g., by pitch-synchronous processing and by means of fundamental frequency control, respectively.

The appropriate reproduction of such features of natural speech signals is, however, a key issue for naturalness of synthetic speech. Naturalness – as opposed to intelligibility, which is attained well – is an attribute that still has to be improved for most state-of-the-art speech synthesis systems. Beyond that, the synthesis of emotional speech, variations and modifications in speaking styles or speaker identity also require high-quality versatile speech generation models.

In previous investigations on nonlinear signal generation algorithms for speech synthesis, the oscillator model based on the prediction of the signal trajectory in phase space [Kub95, PWK98] is laid out as a prospective tool for re-generation of natural speech signals. In particular an appropriate modeling of the above mentioned short-term variations in the oscillator generated signal is reported in several studies (e. g., [MM99, NPC99, MM01]), leading to adequate reproduction of, e. g., spectral properties of speech signals [Man99]. Also, phase coherence is a natural feature of the oscillator model.

The successful application of the oscillator model is, however, limited to the reproduction of a small number of stationary speech sounds only, generally to some specific vowel signals. When attempting to reproduce the results from previous studies one soon finds that for new speech signals, like arbitrary stationary vowel signals, or even the same vowel signal from a different speaker, a specific implementation of the oscillator model reported to yield successful re-synthesis in previous studies fails to adequately reproduce the signal, and often displays "unstable" behavior, comprising, for example, large amplitude peaks in the generated signal, an output signal tending to infinity, or high-frequency oscillations. Thus the robust identification of a stable oscillator model is the first task to be tackled here.

Another shortcoming of the oscillator model as applied until now is the lack of high-frequency components in the synthetic speech signal, identified in several investigations [Bir95, Bir96, MM99, Man99]. To some extent this can be explained by the necessity of regularization applied for oscillator identification, and in [Man99], for example, the control of the amount of high-frequency components by varying the regularization factor is demonstrated for one vowel signal. In general, however, this method cannot be applied successfully. From simple perception experiments, it becomes clear that particularly noise-like high-frequency components of the natural speech signal are missing in the oscillator generated speech signals, even for voiced speech like vowels.

For both of these main challenges a thorough incorporation of knowledge from other speech synthesis methods and from phonetics and speech science seems desirable, for example the consideration of speech production models like the source-filter model, and a basic understanding of the human speech production process is requisite.

1.2 Human speech production

Human speech production is a complex process. Even neglecting the complicated higher-lever linguistic background, or the motor control of muscles and tissue, the mere physical process that leads to the generation of acoustic speech waves is impressive: Air pressure is generated by the lungs – the source of energy – evoking air flow through the larynx, where, for voiced sounds, the flow is modulated according to the nonlinear relation between air pressure and velocity and the dynamical system of the larynx. The larynx is made up of vocal folds which are capable of closing completely together or, as they move apart, creating an opening called the glottis. During normal respiration or for the production of unvoiced sounds, air passes almost freely through the glottis.

For the production of voiced sounds the vocal folds are set under tension and, with air passing through, an *almost periodic*[1] vibration is evoked being the source of an according acoustic wave propagating through the vocal tract. Since the geometry of the vocal tract is not uniform the acoustic wave is partially reflected (and absorbed to a little extent) along the vocal tract before it is emitted from mouth and/or nose as acoustic speech signal. The partial reflections result in a filtering of the glottal source signal depending on the configuration of the vocal tract that allows humans to form specific phonemes. Vowels, for example, can be distinguished by the first two resonance frequencies (formants) of the vocal tract filter.

For unvoiced sounds the glottis is open and no almost periodic source signal is present, but

[1] We shall use the term *almost periodic* to describe a motion or signal with possibly slight variations in frequency, amplitude or exact waveform shape of individual 'fundamental' cycles, i. e., corrupted by some additive signal or noise – but with a strong and clearly identifiable underlying periodic waveform. The term *almost periodic* should be differentiated from *quasi-periodic*: A quasi-periodic signal is a signal composed of a sum of sinusoids, which may be periodic or non-periodic.

the speech sound is evoked by turbulent air flow at constrictions along the vocal tract, e.g., between upper front teeth and lower lip for the fricative /f/, or between the tongue and the palate (roof of the mouth) for /ʃ/[2].

Both the motion of the vocal folds for voiced sounds and the turbulent air flow for unvoiced sounds are *nonlinear processes* converting the direct current (DC) signal of air flow due to lung pressure into audible acoustic waves. While turbulence is a high-dimensional phenomenon, which is probably better modeled by a statistical approach [KAK93], the equations that govern the almost periodic signal evoked by vocal fold oscillations can be modeled by a low-dimensional nonlinear system.

While the presence of an almost periodic source signal for *voiced phonemes* is evident, and can be easily verified from the speech signal, the *presence of noise-like signal components in voiced speech signals* is somewhat more difficult to observe. From the mere signal the noise-like components may be best visible for *mixed excitation* phonemes, like voiced fricatives (/v/, /z/, /ʒ/). However, also for purely voiced phonemes like vowels a certain small noise-like component is present in the speech signal. In general the noise-like component in voiced speech signals is modulated by the oscillatory component.

To add meaning to a speech signal, different speech sounds are connected to form words and sentences, making speech production a highly *non-stationary process*. In normally spoken fluent speech the portion of the speech signal that belongs to *transitions* between phonemes is quite large, meaning that for many phonemes actually *no stationary signal* portion may be observed. This fact also leads to the bad quality of speech synthesis systems based on concatenation of single phoneme elements (in comparison to systems based on elements containing transitions between phonemes).

In the scope of this thesis the non-stationary aspect of speech signals is covered only in a limited way: Since the training of the nonlinear models used here requires a large number of training examples, the models are derived from recordings of *artificially sustained* speech signals for each speech sound. However, in sect. 4.5 some attempts to generate non-stationary synthetic speech signals using the models derived from sustained signals are depicted.

For speech signal processing a main advance is based on the introduction of speech production models [Fan70] into speech analysis, coding, and synthesis algorithms. In particular, algorithms considering the source-filter model of speech production are advantageously applied in speech coding and synthesis.

1.3 Nonlinear modeling of speech signals

Our work aims at modeling the speech signal by a dynamic system, not necessarily in a *physical modeling* sense, like, for example, by mimicking the physical speech production process, but using a dynamic system model that correlates with the natural signal generation system of human speech production, e. g., a low-dimensional oscillatory system for the voiced speech source. There are only specific cases of linear systems that display stable oscillatory output without a driving input signal, the simplest being a purely recursive second order filter with the filter poles at the unit circle. In the 'real world' – i. e., taking energy dissipation into account – autonomous oscillatory systems have to contain *nonlinear elements*.

The oscillator-plus-noise model for speech production developed in this thesis is built upon the *autonomous oscillator model* [Kub95, HP98b, PWK98], based on the *phase-space reconstruction* of scalar time signals by a *time-delay embedding* [Tak81, SYC91] and a *nonlinear*

[2] Symbols within slashes denote *phonemes*, e. g., /f/ and /ʃ/ stand for the initial sound in the words "phoneme" and "she", respectively. We make no special distinction between *different acoustic realizations (allophones)* of one phoneme here, which may, e. g., occur for speech sounds uttered by different speakers. Phonemes are stated in the form of the international phonetic alphabet (IPA).

predictor. We ground on the basis of several recent investigations of this model, particularly investigations regarding the modeling of speech signals.

Basic findings from these investigations include the fact that voiced speech signals are low-dimensional. In [Tow91] a correlation dimension of 2.9 for speech signals (in general) is stated. In particular for sustained vowel signals even lower dimensionality is found, e. g., a saturation in prediction gain for an embedding dimension $N \geq 3$ and a correlation dimension between 1.2 and 1.7 in [BK91]. This means that such signals can be modeled in a reasonably low-dimensional phase space.

A second important finding is that for a given signal the embedding delay of the time-delay embedding can be optimized using mutual information between delayed signal samples [Fra89, BK91], for which computationally fast algorithms are available [BK94, HKS99, BD99]. Hence, the best structure (embedding delay) of the oscillator model can be chosen for each individual signal.

Another finding is that, using the oscillator model the reproduction of important properties of nonlinear systems, like signal dimension or Lyapunov exponents, is possible, as shown, e. g., for modeling the Lorenz system in [HP98b, Ran03].

For the application to speech synthesis a number of investigations [Kub95, Bir95, Kub96b, Kub96a, Bir96, Ber97, Kub98, Ber98, HK98, Man99, MM99, NPC99, LMM00, RK01, Ran01, MM01, Ran03, RK03, LZL03] show encouraging results. In particular, the oscillator generated speech signals often display characteristic features of natural speech signals better than other signal generation algorithms, for example cycle-to-cycle variations [Man99, NPC99, MM01]. Signal generation for speech synthesis based on the oscillator model thus seems to be suitable for achieving natural synthesis results.

Besides synthesis, the applications of phase-space modeling and of the oscillator model for speech signal processing comprise time scale modification [KK94], adaptive-codebook pulse code modulation for speech coding [Kub95], noise reduction [Sau92, HKM01], fundamental frequency analysis (pitch extraction) [Ter02a], determination of instants of equal phase inside the glottal cycle (epoch marking) [MM98, HK05], as well as new concepts for speech recognition [PM02, LJP03, LJP04].

For speech synthesis with the oscillator model a *nonlinear function model* for signal prediction is applied to capture the signal dynamics in embedding phase space. The parameters for the nonlinear function model are, in general, learned from recorded speech signals. A number of possible realizations of the nonlinear function have been investigated in the literature, e. g., lookup tables [Tow91, KK94], artificial neural networks – such as the multi-layer perceptron [PWK98, HP98b, NPC99], radial basis function networks [Bir95, HP98b, MM99, Man99, MM01], or, recently, the support vector machine [LZL03] – or multivariate adaptive regression splines [HK98]. All these nonlinear function models are reported to provide the means for natural speech synthesis when applied in the oscillator model for the re-generation of some example signals. However, positive synthesis results are often achieved for only *a small number* of *stationary vowel signals*. In general, a main challenge for speech synthesis using the oscillator model still lies in the task to obtain a *stable oscillator*. Considering this, a comparison of possible realizations of the nonlinear function is pursued here, first in an easy to visualize one-dimensional regression task in Chapter 3, and specifically for the application in the oscillator model in Chapter 4.

Regarding signal prediction, it was found that nonlinear prediction with predictors of reasonably low complexity actually outperforms linear prediction for speech signals in terms of prediction gain [Tow91, TNH94, FMV97], in particular in long-term prediction tasks with equal predictor complexity [BBK97]. Hence, in most investigations the nonlinear oscillator modeling of the *full speech signal* has been pursued without a possible pre-processing by linear prediction inverse filtering, since the nonlinear predictor is assumed to optimally incorporate the linear prediction part.

For the modeling of voiced speech signals, however, the *pre-processing by linear prediction* (or by other inverse filtering algorithms) should be considered, since it shifts the dynamic modeling task to the regime of the oscillatory system of human speech production, the oscillatory movement of the vocal folds. For this purpose a decomposition of the speech signal into a linear prediction filter or a similar system (e. g., an acoustic tube model) and a residual signal, which corresponds to the glottal source signal (the output of the actual nonlinear system in speech production), will be investigated in combination with the oscillator model in Chapter 4. To our knowledge, a combination of inverse filtering and a nonlinear oscillator has only been investigated in [NPC99] and our own work [RK01, Ran01].

Another important aspect of nonlinear dynamic modeling of speech signals is that only the *deterministic component* can be captured by the model. However, speech generally contains a *noise-like component*, which is generated by turbulent air flow, i. e., by a high-dimensional system. As noted above, this process – though nonlinear and deterministic – should be considered a *stochastic process* [KAK93] and can commonly not be mimicked by a low-dimensional dynamic system. Carefully considering the properties of the noise-like component of speech signals we will develop the means to re-generate a proper noise-like component that can be added to the oscillatory component and allows for faithful re-production of all kinds of stationary speech signals in Chapter 5: The *oscillator-plus-noise model.*

Nonlinear speech modeling – or rather: Modeling of the nonlinear system in speech production – can be traced back to the year 1791 when Wolfgang van Kempelen published a book comprising an astonishingly concise analysis of human speech production and a description of a mechanical *speaking machine* [vK70]. Our work runs a lot in parallel with the evolution of van Kempelen's machine: We start on the foundation of ≫*modeling stationary vowel signals*≪[3] laid in previous investigations on the oscillator model, and aim at the re-generation of general mixed excitation speech sounds as well as unvoiced sounds (≫*consonants*≪) with one and the same model, including the aim to create transitions between different sounds and thus ≫*combine them to syllables and words*≪. Hence, we shall include some citations from van Kempelen's work here, starting with the encouraging notion that, following the path indicated above, ≫*it is possible to make an all-speaking machine*≪:

> Eine ſprechende Maſchine erfinden, und ſie nach einem überdachten Plan ausführen wollen, wäre wohl einer der verwegneſten Entwürfe geweſen, die je in eines Menſchen Seele entſtanden ſind. Eh ich zur Beſchreibung meiner Sprachmaſchine ſchreite, muß ich dem Leſer das aufrichtige Geſtändniß machen, daß mir anfangs gar nicht in den Sinn gekommen iſt, an einer ſolchen Maſchine zu arbeiten. Als ich anfieng Verſuche zu machen, war höchſtens meine Abſicht einige Selbſtlauter, einige Töne der menſchlichen Stimme durch irgend ein Inſtrument nachzuahmen; an die Mitlauter, die mir gar ſchwer ſchienen, getraute ich mich gar nicht zu gedenken, und ſie vollends mit den Selbſtlautern zu verbinden, hielt ich ganz für unmöglich, ja ich war ſogar mit den wichtigſten Lauten oder Buchſtaben im Einzelnen ſchon Jahre lang fertig, eh' ich die Möglichkeit nur von weiten einſah ſie je aneinander zu hängen, und dadurch Sylben und Wörter hervorbringen zu können. Man wird aus dem Folgenden ſehen, wie ich nur nach und nach, und zwar ſehr ſpät auf den Gedanken gekommen bin: Es iſt möglich eine alles ſprechende Maſchine zu machen.

Wolfgang van Kempelen [vK70, § 210, pp. 388f]. Von der Sprachmaschine.

[3] Texts enclosed by ≫ ≪ quotes are partial translations of the corresponding reproductions from [vK70] by the author.

1.4 Thesis outline

The structure of the remainder of this thesis is the following: In Chapter 2 we provide a brief overview of current speech synthesis techniques. We focus particularly on signal generation – as opposed to text processing, prosody generation, or parameter control. The motivation to include this description of 'linear' synthesis techniques is that we will refer to several of them in the development of the oscillator-plus-noise model.

Chapter 3 deals with a number of realizations of the nonlinear function to be used in the oscillator model, mainly with models based on radial basis function (RBF) networks. Since in the oscillator model the nonlinear function is applied in a higher than two- or three-dimensional space, thus being difficult to visualize, the nonlinear function models will be compared on a one-dimensional regression task in this chapter. Characteristics of the different nonlinear function realizations important for speech modeling are deduced.

In Chapter 4, first, the fundamentals for nonlinear prediction of scalar signals are presented, including signal embedding in phase space, and leading to the *autonomous oscillator model*. Application of the oscillator model to the re-generation of stationary vowel signals is exemplified for different nonlinear function realizations and RBF network training methods. Second, a combination of vocal tract modeling by linear prediction filtering and the oscillator model is developed, resulting in improved stability and spectral reconstruction of the re-generated speech signals. Furthermore, the possibility of varying oscillator parameters for non-stationary modeling by interpolation of RBF network weights is considered.

The still unsatisfactory spectral re-construction in the high-frequency range of wide-band vowel signals motivates the introduction of the *oscillator-plus-noise* model in Chapter 5. Speech signals generally comprise a noise-like component due to the high-dimensional fluid dynamics of turbulent air flow. Since this high-dimensional component cannot be modeled by a low-dimensional oscillator, we propose to re-generate the noise-like signal component using a random noise signal that is pitch-synchronously modulated in amplitude, with an individual modulation envelope learned for each speech sound. Thus, vowel signals are re-synthesized satisfactorily, and with the extension of a second linear prediction path for individual spectral shaping of the noise-like component, besides vowels, also stationary mixed excitation and unvoiced speech signals can be re-synthesized. As performance measure, synthesized signals are compared to natural speech signals regarding objective measures related to naturalness.

The scientific contributions and the main conclusions and potentials of this thesis are summarized in Chapter 6.

Chapter 2

Current speech synthesis techniques

In this section we will give a summary of a number of *signal generation* techniques of current speech synthesis systems, traditionally classified into the three categories *articulatory synthesis* (sect. 2.1), *formant synthesis* (sect. 2.2), and *concatenative synthesis* (sect. 2.4). A special case of concatenative synthesis is the so called *unit selection* method (sect. 2.5).

Several aspects of all these synthesis algorithms are important for this work. Since we claim to capture the dynamics of the actual nonlinear oscillator in speech generation – although not necessarily in a one-to-one modeling of vocal fold movements – we also refer to some models for *glottal waveform generation* (sect. 2.3) utilized in articulatory and formant synthesis. It shall be advantageous to relate the oscillator model to signal generation models that have been used in the 'traditional' synthesis systems.

The last section (sect. 2.6) is dedicated to the challenges arising from some *commonly encountered problems* using current speech synthesis techniques.

As mentioned, we will primarily focus on the *signal generation* stage of the different synthesis algorithms, greatly neglecting the (no less important and challenging) task of *control parameter generation*, which is not covered in this thesis.

2.1 Articulatory synthesis

Articulatory synthesis is based on modeling the physics of the human articulators ('physical modeling'), like vocal tract geometry and vocal fold movements. The speech production process is imitated by inducing 'movements' of the model articulators akin to the way a human person would do. The concept of mimicking the human speech production system has been first exploited more than 200 years ago by van Kempelen's mechanical 'speaking machine' that was able to reproduce 'all sounds of the German language' [vK70]. Nowadays, however, articulatory synthesis relies on mathematical models rather than mechanical models for the physical structure involved in the human speech production process [Scu90].

Control of articulatory movement is accomplished by *speech gestures* (i. e., stylized movements of tongue, lips, etc.) [SA83, Scu86, Bai97] or even by *neuromotor command*: To speak, a person thinks of a message and sends commands to her/his lungs and vocal tract muscles, which cause air flow from the lungs, vocal fold oscillations (for voiced speech), and articulator movement, changing the shape of the vocal tract and resulting in the production of the speech signal [O'S87]. The increase in computing power over the last years allows for a fine control of articulatory movement in current articulatory synthesizers (see, e. g., [IGW$^+$03]), and the combination with a virtual 'talking head' [BBEO03]. The physical modeling of evolution and of propagation of the acoustic signal naturally incorporates feedback from the vocal tract filter on the glottal dynamics.

Albeit the convincing concept, and despite the power of current computing systems, the implementation of an articulatory synthesizer is quite involved and not crowned with the success of resulting in high quality synthetic speech. The reasons are twofold: First, building an articulatory synthesizer relies on a number of measurements (e. g., X-ray data for vocal tract shapes [BBRS98]), assumptions and simplifications that may not be as comprehensive and accurate as necessary. Second, in the actual synthesis procedure a fine-grained control of the synthesizer parameters (articulator movements, etc.) is necessary, that accounts for variations in the natural speech production process, like different time constants for the various articulators, co-articulation, etc. Generating appropriate trajectories for the control parameters as a function of time is a task that still requires manual optimization and cannot be accomplished satisfactorily by a fully automatic system.

Some of the models used for articulatory synthesis are, however, also relevant for other synthesis techniques, namely the approximation of the vocal tract shape by area functions (the cross section areas of a series of, usually, equal-length uniform tubes), which is closely related to linear prediction [MG76]. Linear prediction can be utilized to estimate area functions for articulatory synthesis or the synthesis filter transfer function for formant synthesis (see sect. 2.2 below), and is also used in a number of concatenative synthesis techniques (sect. 2.4).

2.2 Formant synthesis

Formant synthesizers are based on the source-filter model for speech production [Fan70], which distinguishes between a system that gives rise to acoustic waves (the source) and a system that influences the properties of the acoustic waves on its way from the source to the free-field (the filter). For synthesis the voice source is either modeled by a periodic signal related to the glottal pressure signal for a voiced source or by a noise signal for an unvoiced source. The source signal is fed through a linear, slowly time-varying filter that resembles the vocal tract characteristics and, e. g., establishes the distinctive formant resonances for different vowels. In formant synthesis there is commonly no possibility to include feedback from the filter to the source.

The simplest formant synthesizers use a periodic train of impulses with a fundamental period T_0, corresponding to the fundamental frequency $F_0 = 1/T_0$, for voiced sounds or white noise for unvoiced sounds as source signal, as well as a time-varying all-pole filter $H(f,t)$ (or a set of parallel or cascaded resonator filters) to establish the relevant formants (fig. 2.1). More elaborate models use a variable mixing of both periodic and noise source signals, a glottal pulse shaping filter for the voiced source, additional anti-resonance filters (for nasals), a filter for lip radiation, and amplitude modulation of the noise source signal related to the fundamental period for mixed excitation. A prominent example for an elaborate formant synthesis system is the Klatt-synthesizer (e. g., [Kla80]).

Formant synthesizers can generate high-quality synthetic speech if appropriate control parameter sequences are supplied. However, for this purpose the control parameter sequences commonly have to be optimized manually and cannot be generated entirely automatically. Formant transitions between phonemes, for example, depend on the phoneme context (co-articulation) and cannot be re-generated satisfactorily by simple rules. Thus, as for articulatory synthesis, the difficulty of automatically generating adequate control parameter trajectories hinders the use of formant synthesizers for general purpose text-to-speech synthesis.

2.3 Glottal waveform generation

In the framework of formant and articulatory synthesis several models for the generation of the glottal source signal have evolved. Since here we are also concerned with finding an appropriate

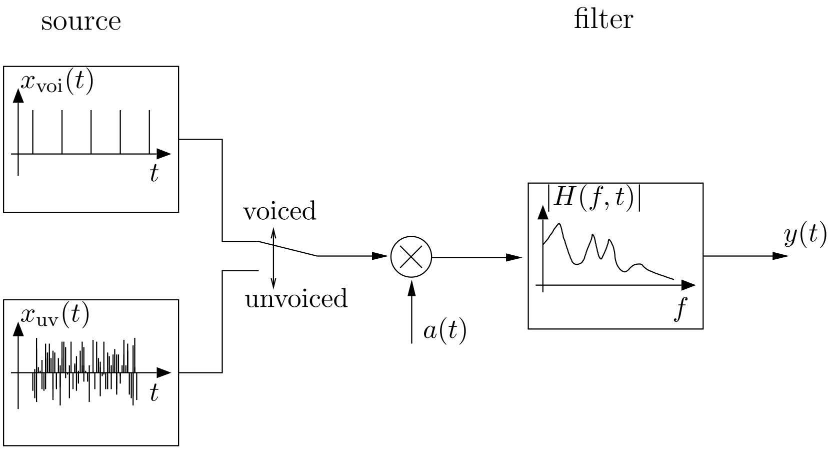


Figure 2.1: Simple speech synthesis system based on the source-filter model for human speech production. As signal source either a series of impulses $x_{\mathrm{voi}}(t)$ – corresponding to the glottal derivative waveform – for voiced speech, or a white noise signal $x_{\mathrm{uv}}(t)$ for unvoiced speech is used. The source signal is filtered by a (slowly time-varying) linear filter $H(f,t)$ corresponding to the vocal tract filtering in speech production to yield the synthetic speech signal $y(t)$. The coefficients for the filter $H(f,t)$ can be determined by rule, e. g., by specification of formant frequencies and bandwidths (formant synthesis), or may be deduced from recorded speech signals by linear prediction analysis (as used for pulse-excited LP synthesis, see sect. 2.4.2). The time varying amplitude is controlled by the parameter $a(t)$.

representation for the source signal we will shortly introduce some of them in the following.

The Liljencrants-Fant (LF) model is a parametric signal model for the glottal waveform [FLL85]. The glottal waveform shape is characterized by four parameters $t_{\mathrm{p}}, t_{\mathrm{e}}, t_{\mathrm{a}}$ and E_{e}, as well as the length of the fundamental period T_0. For one fundamental period the derivative of the glottal waveform – which corresponds to the acoustic pressure signal that is used as input to the vocal tract filter – is composed of two smooth functions joined at the instant of glottal closure t_{e}. An example of the glottal waveform and its derivative represented by the Liljencrants-Fant model is given in fig. 2.2 labeled with the model parameters. The main excitation is due to the large (negative) pulse in $g(t)$ at the instant of glottal closure t_{e} corresponding to the delta impulses in the simplified model in fig. 2.1. Similar parameterization of the glottal waveform is used in the Rosenberg model and other related models [Ros71, Vel98].

The parameters of the above models are not all directly related to physical properties of the glottis or acoustical properties of the source signal. Thus, appropriate parameter values often have to be determined from recorded speech signals by inverse filtering (cf. sect. 4.4.1) or in an analysis-by-synthesis procedure. Although the LF model only comprises four parameters for the fundamental waveform shape it is possible to achieve a wide variety of different speech characteristics, even when the shape parameters are derived from only one control parameter [Fan95]. However, for the adequate reproduction of specific speech qualities, like source-filter interaction, or for the identification of speaker dependent variations of the fundamental waveform shape, the parameterization by the LF model does not suffice, and additional modeling effort is necessary [PCL89, PQR99].

Other parametric models represent the glottal waveform by a weighted sum of *basis functions*. The most common example of such a signal decomposition into basis functions is the Fourier series representation, which is also the basis for sinusoidal modeling discussed in sect. 2.4.3.

The model for glottal signal generation proposed in [Sch90] and refined in [Sch92] deals

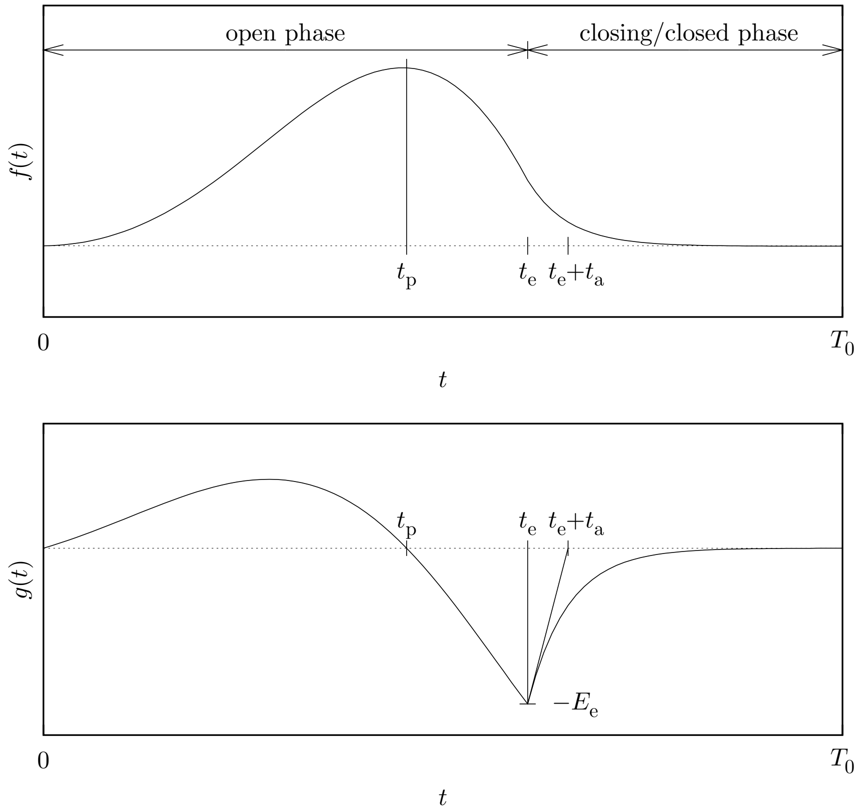


Figure 2.2: Example glottal flow waveform $f(t)$ and acoustic pressure waveform $g(t)$ (time derivative of $f(t)$) generated by the Liljencrants-Fant model for one glottis cycle. Model parameters are: t_p instant of maximum glottal flow, t_e nominal instant of glottal closure, t_a time constant of exponential recovery, and E_e absolute value of glottal flow derivative at t_e. The periodic repetition of the signal $g(t)$ can be used as the source signal of voiced speech instead of a series of pulses in formant synthesis (cf. fig. 2.1).

with *polynomial series* (Volterra shaping functions) of a sinusoidal basis function. The polynomial coefficients can be deduced from the Fourier coefficients of an inverse filtered recorded speech signal. Synthesis is performed by feeding a sinusoidal signal to a static nonlinear function. Fundamental frequency, duration, and amplitude are easily controlled by changing the respective parameters for the driving sinusoidal signal. This synthesis model allows for the generation of different spectral characteristics in the output signal by changing the amplitude and phase of the sinusoidal driving signal. For a low amplitude input signal the output signal will be almost sinusoidal. With increasing amplitude the amount of higher harmonics will increase, too. This is a feature also encountered in natural speech signals. For a fixed nonlinear function, however, the spectral characteristic of the output signal is solely determined by the amplitude and phase of the sinusoidal input signal and cannot be controlled independently of these parameters.

A glottal signal generation model based on a second order resonance filter is proposed in [DA00]. Here, the nonlinear function acts on the (two-dimensional) state vector of the resonator, and its output is fed back as filter input signal. Thus, the fundamental frequency can be controlled by the resonance filter parameters whereas the waveform shape is determined by the nonlinear function. Using a quite low number of parameters the system is able to regenerate glottal flow waveforms from inverse filtered speech signals, and allows for easy control of fundamental frequency due to the use of *physically informed* parameters.

The approach towards the generation of the glottal signal from the field of articulatory synthesis is *physical modeling* of the vocal fold oscillations. Here, a simplified description of the mechanics of the vocal folds is set up by means of masses, springs, and damping elements together with equations for the fluid dynamics of the air passing through the vocal folds. The resulting differential equations are solved to yield the acoustic signal.

The most prominent among physical models for the vocal folds is the *two-mass model* introduced in [IF72] and widely studied [Tit88, Per88, Per89, Luc93, TS97, LHVH98, JZ02], and extended, e. g., to a three-mass model in [ST95]. The main control parameter for the two-mass model is the sub-glottal pressure. To achieve a desired glottal waveform, however, a complex fine-tuning of the system parameters is necessary. However, it has been shown that, for example, by varying sub-glottal pressure and one other parameter (amount of coupling between the two model masses) a variety of possible system behaviors from nonlinear system theory is possible, like bifurcations resulting in sub-harmonics of the fundamental frequency [Luc93], and that introducing turbulence noise or random stiffness in the two-mass model induces chaotic behavior [JZ02]. If the two-mass model is directly coupled to a vocal tract model [IF72, TS97], the resulting articulatory synthesis system naturally includes feedback from the vocal tract filter to the source, which is not the case for formant or concatenative synthesis systems, in general. There is, however, no direct relation between the parameters of the two-mass model and prosodic parameters, or spectral content. Thus, as previously mentioned for articulatory synthesis in general, the control of the two-mass model in terms of F_0, amplitude, or spectral characteristics of the output signal becomes a difficult task.

2.4 Concatenative synthesis

The currently prevailing speech synthesis approach is *concatenative synthesis*, i. e., synthesis based on the concatenation of prerecorded speech segments. Compared to model-based speech synthesis (i. e., articulatory and formant synthesis), the use of recorded speech segments in concatenative synthesis yields a perceptually more natural synthetic speech output, as of today. Due to the use of recorded speech signals, the natural segmental quality of the signal is maintained during synthesis by concatenation to a great degree. Also, by an adequate choice for the length of the recorded signal segments the natural trajectories of signal parameters – like energy, formant frequencies and bandwidths, etc. – over time are reproduced in the synthetic speech signal. To that end, the speech signals chosen as inventory segments usually do not relate to single phonemes, but rather to *transitions* between phonemes, called *di-phones*, or even include more than two phonemes with a strong mutual influence in articulation and segmental duration (co-articulation). Strong co-articulation is, for example, encountered in the German language with its consonant clusters of various size, and here the recorded segments often comprise *demi-syllables* [Det83, Por94, RP98].

For concatenative synthesis with a fixed signal inventory, a medium size corpus of segments (between 1000 and 2000 segments [Por94, RP98]) has to be recorded in a controlled environment – that is, each segment embedded in a sentence at a neutral position regarding stress, lengthening, etc. The inventory segments have to be extracted from the corpus of recorded speech, which is a time-consuming task and requires considerable workload. Simple concatenative synthesis systems rely on a corpus with one instance of each segment and during the synthesis procedure the series of segments to be concatenated is chosen by rule.

More elaborate concatenative synthesis systems make use of more than one instance for each inventory segment, for example stressed/unstressed instances of one segment[1]. In this case, the instance of one segment is chosen during synthesis according to the information

[1] The different instances of the inventory segments may as well represent different speaking styles or different

about stress from text preprocessing. This is a first step towards unit selection synthesis (see sect. 2.5) and results in a concatenation of speech segments that reflect part of the natural prosody and in less artifacts due to signal manipulation and concatenation [KV98].

For natural speech synthesis we want the synthetic speech signal to resemble the trajectories of the prosodic parameters (F_0 contour, segmental duration, amplitude contour) of a human speaker. Since natural prosody greatly varies with the content of the spoken message, the synthetic speech signal concatenated from the inventory segments has to be modified to achieve a desired prosody[2]. This is accomplished by a suitable method to alter the prosodic parameters in the prerecorded speech segments, a *prosody manipulation algorithm*.

Many researchers in speech synthesis consider an adequate prosody the main key for "naturalness" – in contrast to intelligibility – of synthetic speech [TL88, BHY98, KBM$^+$02]. The advantage of yielding an adequate prosody, however, is of course coupled to a degradation in segmental quality when the speech signal is processed to embody the desired prosodic parameters. Or, to quote from [Bai02b]:

> (. . .) the modifications are often accompanied by distortions in other spatio-temporal dimensions that do not necessarily reflect covariations observed in natural speech.

In the following we present an overview of a number of prosody manipulation algorithms commonly used in current speech synthesis systems.

2.4.1 Pitch synchronous overlap add (PSOLA)

Pitch synchronous overlap add is possibly the most widely applied algorithm for prosody manipulation today. Is is based on a signal segmentation related to the individual fundamental cycles, based on estimated *glottal closure instants (GCIs)*, and achieves fundamental frequency modification by shifting windowed signal segments from each GCI to a position in time such that the lags are the inverse of the desired fundamental frequency. The series of overlapping windowed signal segments are then added to give the modified speech signal [MC90], as depicted in fig. 2.3. Duration is altered by repeating or leaving out signal segments corresponding to individual pitch cycles in the overlap and add procedure.

PSOLA is easy to implement, relies on a good estimation of glottal closure instants[3], and can be applied to the full speech signal without any additional signal processing. However, for large scale modifications of fundamental frequency, and for duration modification of unvoiced phonemes processing artifacts are clearly audible, like artificial periodicity in unvoiced segments. In spite of the simplicity of the algorithm, the processing artifacts introduced using PSOLA for fundamental frequency modifications are not easy to describe analytically. In the original publication of the algorithm [MC90] the effect of a comb-filter (periodic spectral zeros) has been identified if the window function length is chosen four times the fundamental period, i. e., considerably longer than the usual window length of twice the fundamental period (as in fig. 2.3). This is due to the destructive interference of specific frequency components in the overlay-add process. Analysis of processed speech signals for the 'classical' window of length of twice the fundamental period revealed a smoothing of the formant peaks in the spectrum of the output signal, but other effects – e. g., on the phase spectrum – were described as 'difficult to analyze'. An important prerequisite for PSOLA to avoid additional processing artifacts is

emotional states of the speaker and can be chosen, e. g., according to background information about the current speech act in a dialog system [ICI$^+$98].

[2] In text-to-speech systems the 'desired prosody' is commonly derived from the text input, either by rule or by machine learning techniques.

[3] Robust estimation of the GCIs is a complex task, see [Hes83].

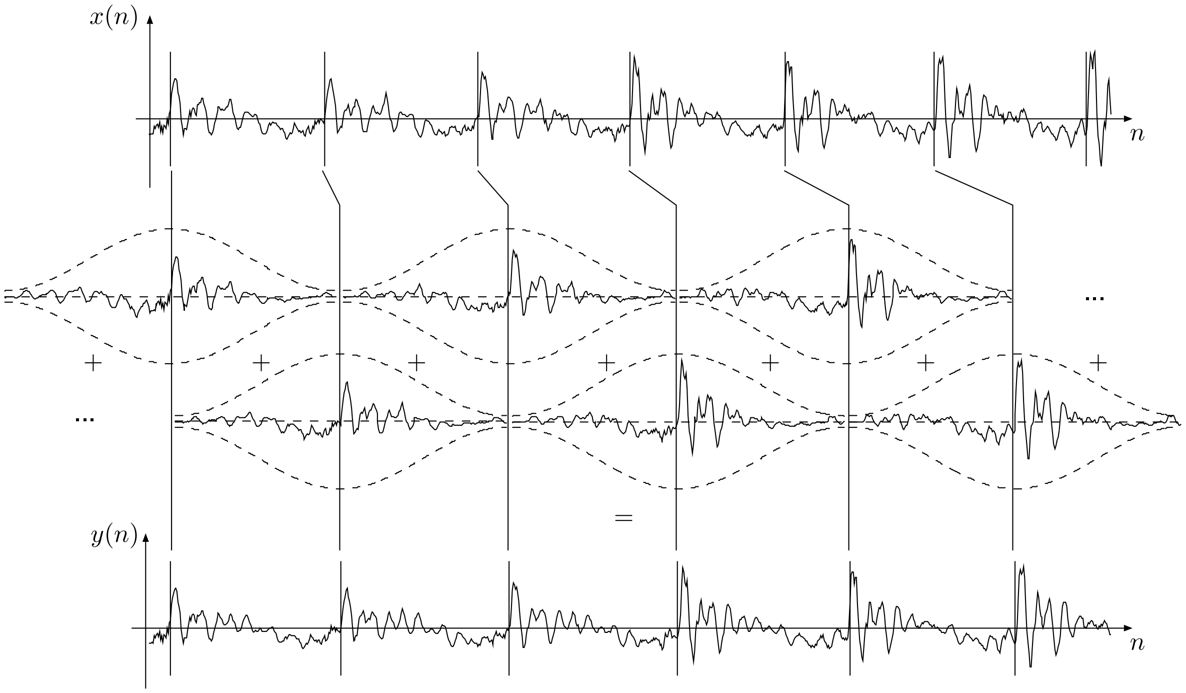


Figure 2.3: Fundamental frequency modification using PSOLA. The original speech signal $x(n)$ (taken from the vowel /a/, at top) is partitioned according to GCI markers (vertical lines). Segments are centered at GCIs and comprise two fundamental periods in this example. To alter fundamental frequency the windowed segments are shifted in time to positions representing GCIs according to the desired fundamental frequency contour. The shifted segments are then added to form the output signal $y(n)$ (at bottom).

the precise and consistent determination of the GCIs, in particular the consistency of GCIs for signal parts that shall be concatenated, to avoid phase mismatches at the concatenation.

Multi-band re-synthesis overlap add (MBROLA) [DL93] has been developed to overcome some of the processing artifacts of PSOLA. It applies a normalization of the fundamental period and of the glottal closure instants by harmonic re-synthesis [MQ86] of voiced segments in an off-line preprocessing stage. This normalization results in a reduction of phase mismatches in the PSOLA synthesis procedure and provides the means for spectral interpolation of voiced parts [DL93].

The MBROLA algorithm is widely used in research and experimental synthesis systems since a text-to-speech system is freely available for non-commercial use with pre-processed corpora for a variety of languages[4].

2.4.2 Linear prediction (LP) based synthesis

Linear prediction or inverse filtering (see also Chapter 4.4.1) is widely used in speech signal processing, e. g., in speech coding to reduce the bit rate, in speech analysis for formant estimation, in speech recognition for feature extraction, and also in speech synthesis. Linear prediction is related to the source-filter model of speech production in the way that the LP synthesis filter resembles the acoustic tube model of the vocal tract [MG76, sect. 4.3]. LP based concatenative synthesis uses recorded speech signals to estimate the coefficients of the LP filter for a sequence of speech frames. Frames are either of fixed length (typically about

[4] http://tcts.fpms.ac.be/synthesis/mbrola.html

20 ms) or related to the pitch periods (pitch synchronous analysis). In the synthesis procedure the LP synthesis filter is excited either by impulses, by the residual from LP analysis, or by an artificial glottal pressure signal. Prosodic features are solely related to the excitation signal.

Pulse-excited LP synthesis

In pulse-excited LP synthesis, the LP synthesis filter is excited by a series of impulses. The temporal distance between two impulses determines the fundamental period T_0 and thus the fundamental frequency F_0. This resembles the simple formant synthesis model of fig. 2.1 with the filter transfer function $H(f, t)$ determined by LP analysis.

In an elaborate model also used for speech coding, the excitation consists of a series of codewords with several impulses per pitch period [CA83, CA87]. Codewords are chosen according to the LP analysis residual signal by minimizing a distortion criterion.

Residual excited linear prediction (RELP) synthesis

The residual from LP analysis of a speech signal is used to excite the LP synthesis filter. Without prosodic manipulations, perfect reconstruction of the speech signal is possible, only limited by a possible quantization of the residual signal for storage. For prosodic manipulations, the residual has to be processed. Commonly utilized algorithms for the prosody manipulation are LP-PSOLA and SRELP.

LP-PSOLA

For LP-PSOLA the PSOLA algorithm is applied to the residual signal of LP analyzed speech. Like PSOLA on the full speech signal, LP-PSOLA is performed pitch synchronously. To that means a glottis closure instant analysis (pitch marking) is necessary and also the LP analysis can be performed pitch synchronously.

Fundamental frequency manipulations of voiced phonemes using PSOLA on the residual signal often result in less artifacts than for using PSOLA on the full speech signal, since for voiced speech the energy in the residual signal is commonly concentrated at the glottis closure instant. For unvoiced speech similar problems are encountered as for PSOLA applied to the full speech signal.

SRELP

A prosody manipulation algorithm closely related to LP-PSOLA is *simple residual excited linear predictive* synthesis. Here, no windowing and no overlap add procedure is applied, but the residual signal segments centered at the GCIs are simply cut or zero padded (symmetrically at both ends) to yield a segment length equal to the desired fundamental period. Since for most voiced speech signals the energy of the residual signal is concentrated at the glottis closure instants these segment length manipulations affect only little of the total signal energy. Also some of the artifacts caused by adding the overlapping signal segments in the LP-PSOLA algorithm are avoided with SRELP, however, other degradations are introduced, particularly if LP analysis results in a residual signal with the energy spread more widely over time. No frame length modifications are applied to unvoiced speech frames, avoiding the artificial periodicity artifacts encountered for such signals using PSOLA and LP-PSOLA.

Like LP-PSOLA, SRELP requires GCI determination and preferably pitch synchronous LP analysis. SRELP is implemented for concatenative synthesis in the freely available *Edinburgh*

Speech Tools for the *Festival* speech synthesis system[5], and is also used in the phoneme-to-speech front-end for the *Vienna Concept-to-Speech* system[6] [Kez95, Ran02].

2.4.3 Sinusoidal/harmonic-plus-noise modeling

For the sinusoidal model the speech signal is considered a sum of sinusoidal components. This relates to the model of (voiced) speech sounds being generated by a periodic source filtered by a linear vocal tract filter, both slowly time-varying. The number of sinusoidal components as well as frequency, phase, and amplitude of each component are commonly estimated frame-wise from a short-time Fourier transform or by analysis-by-synthesis of a recorded speech signal. During synthesis the signal is re-generated as the sum of the output of several sine signal generators [MQ86, Qua02]. Due to the non-stationary nature of speech, individual sinusoidal components do not only vary in amplitude and frequency over time, but new components may emerge (be born) or components may vanish (die).

In the original sinusoidal model [MQ86] the sinusoidal components comprise for both the periodic and the non-periodic signal component of speech signals (the presence of these two components is most evident in mixed excitation speech sounds, like voiced fricatives or breathy vowels). As such, the sinusoidal model can only be applied for speech coding and time scale modification.

The harmonic-plus-noise model [Ser89, LSM93] considers sinusoidal components with a harmonic relation, i. e., components with a frequency that is an integer multiple of the fundamental frequency, as the harmonic (periodic) signal component and the residual (difference between original speech signal and the harmonically modeled signal component) as the noise component. The residual is modeled either again by a combination of sinusoids or as a filtered time-variant random process using linear prediction and a (possibly sub-sampled) time-varying power estimate. Fundamental frequency modifications are achieved by changing the frequency of each sinusoid in the harmonically related signal component to integer multiples of the desired fundamental frequency (spectral re-sampling) [Sty01] (or by applying PSOLA to short-time synthesis frames [MC96], which, however, introduces the shortcomings of the PSOLA algorithm into sinusoidal modeling).

With prosodic manipulations, the amplitudes, frequencies, and phases of the individual sinusoidal components have to be readjusted appropriately for re-synthesis to attain the correct temporal trajectories and spectral envelope. The latter can be achieved, for example, by an estimation of the spectral envelope by LP analysis and an appropriate modification of the amplitudes of the harmonic components after spectral re-sampling [WM01, Bai02b, OM02], so that the spectral distribution of the synthetic signal is maintained.

For the adequate re-generation of the noise-like residual signal a pitch synchronous modulation of the random process is necessary [LSM93, SLM95, Bai02b]. To that end, the application of a parametric noise envelope for each fundamental period that is composed of a constant baseline and a triangular peak at the glottis closure instant is proposed in [SLM95]. Furthermore, the spectral content of the noise-like signal component can be shaped by an individual LP synthesis filter [SLM95, Bai02b]. Harmonic-plus-noise modeling is reported to yield synthetic speech of very high quality [SLM95, Sty01].

2.4.4 Comparison of prosody manipulation algorithms

The comparison of different prosody manipulation algorithms for concatenative synthesis is commonly performed by *perceptual ratings* of human subjects in terms of a mean opinion score. In [MAK$^+$93], e. g., it was found that, in terms of intelligibility, LP-PSOLA and SRELP

[5] http://www.cstr.ed.ac.uk/projects/festival/

[6] http://www.ai.univie.ac.at/oefai/nlu/viectos/

perform better than PSOLA on the full speech signal, while PSOLA in turn outperforms pulse-excited LP synthesis. The harmonic-plus-noise model is also reported to give better ratings for intelligibility, naturalness, and pleasantness than PSOLA [Sty01].

In the framework of the European Cost258 action ("The Naturalness of Synthetic Speech") an *objective comparison* of prosody manipulation algorithms for concatenative synthesis has been carried out [BBMR00, Bai02a].The objective rating of the prosody manipulation algorithms was based on a number of distance measures originally used for rating speech enhancement algorithms [HP98a]. A subset of these distance measures was applied to estimate the perceptual difference between natural speech samples and speech samples processed by the prosody manipulation algorithms with the task to gain the same prosody as the natural samples. As input for the prosody manipulation algorithms a natural speech sample with appropriate segmental content (same phoneme/word/sentence) but with "flat" prosody, i. e., nearly constant fundamental frequency and no stressing/accentuation/etc., was supplied.

It was found that some correlation exists between the different distance measures applied, and that about 90% of total variation in the rating of the prosody manipulation algorithms is captured in the first two components from a principal component analysis. Analysis of these first two components allows to distinguish between (a) the overall distortions (signal to noise ratio, SNR), and (b) the ratio between distortions in voiced regions and distortions in unvoiced regions of the signal, and thus to rate the individual algorithms in terms of their processing quality for voiced and unvoiced speech signals, respectively. In general it was discovered that all the prosody manipulation algorithms taking part in the test could only generate signals not closer to the human target signal than a noisy reference signal with an SNR of 20 dB [Bai02a].

2.5 Unit selection

The term *unit selection* denotes concatenative speech synthesis with selection of the segments to be concatenated from a large database of recorded speech signals at synthesis time. The selected "units", i. e., speech signal segments, are chosen to minimize a combination of *target* and *concatenation (or transition) costs* for a given utterance [Cam94, HB96]. The units to be concatenated can be fixed phonemic entities, for example single phonemes or syllables, or units of *non-uniform* size [Sag88, BBD01, PNR$^+$03], which complicates the selection task considerably.

The target costs are calculated by a measure of the difference between properties of a unit in the database and the target unit specified as input for the unit selection algorithm. If, for example, phonemes are considered as units, the phoneme identity, duration, mean F_0, formant frequencies, but also information about phonemic context, stress, or position in a word may be used to compute the target cost. The concatenation cost relates to the quality degradation due to joining segments with, e. g., mismatch in fundamental frequency or spectral content. A number of measures for such mismatches can be used, such as F_0 difference, log power difference, spectral or cepstral distance measures, formant frequency distances, etc. [KV98, WM98, CC99, KV01, SS01]. Minimizing the join costs is done by a path search in a state transition network (trellis) for the whole utterance with the target costs as state occupancy costs and the join costs as state transition costs (e. g., using the Viterbi algorithm) [HB96]. Since the criteria for the selection of units can be chosen for a specific purpose, unit selection may also be used to generate speech with different speaking styles or emotional content if the appropriate information is present in the database [ICI$^+$98].

In many unit selection synthesis systems *no or minimal signal processing* for prosody manipulations or smoothing is applied, since the optimum selection of units is supposed to yield units matching the desired prosody and with reasonable small discontinuities at the concatenation points. However, particularly for proper names not incorporated in the recorded database, the unit selection algorithm often has to select a series of short units with probably significant discontinuities [Stö02]. In this case, output speech quality suffers a degradation, which is especially annoying if the proper name is the segment of the utterance which carries information (like, e. g., the name of a station in a train schedule information system).

Nevertheless, unit selection synthesis is very popular at current, and several experimental and commercial systems have been built, like CHATR [BT94, HB96, DC97], which, together with the Festival speech synthesis system [BTC97, BC05], forms the basis for the AT&T 'Next-generation text-to-speech system' [BCS+99].

2.6 Challenges for speech synthesis

Despite the fact that most state-of-the-art speech synthesis systems are rated well in terms of *intelligibility*, many systems are judged worse concerning *naturalness*. The possible reasons for this discrepancy seem to be numerous and are not all investigated in detail yet.

In the European Cost258 action [KBM+02] a great effort was dedicated to the identification of reasons for this problem and to the proposition of possible remedies. As a main point for achieving natural synthetic speech the identification and reproduction of *natural dynamics* of the speech signal, as specified below, is considered. As noted above, for articulatory and formant synthesis the generation of adequate control parameter trajectories is a main factor determining the quality of the resulting synthetic speech signal. In data based concatenative or unit-selection synthesis this problem is partially resolved due to the utilization of recorded speech samples. Here, however, the necessary prosodic manipulations may degrade the natural signal quality. Moreover, the problem of concatenating recorded speech segments with mismatches in parameters, like spectral content, arises.

For the discrimination between natural and unnatural speech signals, a main issue appears to be the dynamic evolution of the signal parameters. These parameters can roughly be classified into the categories of supra-segmental features – including prosodic parameters, like F_0 trajectories and segmental duration, or formant frequency trajectories – and segmental features related to the speech waveform shape (sometimes denoted as features on the 'signal level') – like, e. g., spectral content, F_0 or amplitude variations on a short-time level, or noise modulation. The correct (natural) evolution of both supra-segmental and segmental features over time must be considered most important for naturalness of synthetic speech, and may be responsible for the limited success of articulatory and formant synthesis – where naturalness is determined by the appropriateness of the artificially generated control parameter trajectories, in comparison to concatenative or unit selection synthesis – where, to some extent, natural trajectories are reproduced from the recorded speech signals.

However, since the recorded segments may not comprise the desired prosody the rendering of *natural prosodic feature trajectories* is a major factor for perceived naturalness in concatenative synthesis, too. To a large part, this is a task for the so called 'higher-level' processing, i. e., the text or content analysis and phonetic pre-processing traditionally decoupled from the signal processing stage. Much effort has been dedicated to the generation of appropriate *supra-segmental prosodic parameters*, like F_0 trajectories and segment durations[7] from text input. However, concerning the fact that the natural segmental features are altered by the prosody manipulation algorithm, the increase in naturalness of concatenative synthesis when the 'correct' supra-segmental features for prosody are available to the synthesizer may be tied to a decrease in segmental quality due to (large scale) manipulations of prosodic features [JSCK03].

Nevertheless, as can already be seen from the remarks for the number of prosody manipulation algorithms above, we should aim at the target of *optimizing the non-prosodic parameters*,

[7] Amplitude trajectories are often not modeled explicitly for concatenative synthesis.

too. The fact that the incorporation of the *source-filter model* in the form of *linear prediction* results in better ratings for synthetic speech quality [MAK+93, Sty01] already points to the importance of *spectral content* and *precise formant frequency and bandwidth trajectories*. Also the *short-term variations of fundamental frequency and amplitude* have long been identified as an important cue for naturalness and should be re-generated appropriately. In the setting of the harmonic-plus-noise model the importance of *noise modulation* is encountered again, which has been an issue in formant synthesis for quite a time. Although the decomposition into harmonic and noise parts of the speech signal seems to be adequate, their interaction must not be neglected. Recent investigations into the nature of noise modulation [JS00a, JS00b] suggest, however, that modulation by one parametric amplitude envelope (as, e. g., proposed in [SLM95]) may not suffice to adequately model noise modulation for both vowels and voiced fricatives.

For unit selection synthesis the problems concerning segmental features are less significant. Especially when large units from the database can be used the natural segmental features will be simply reproduced. Here, however, and for concatenative synthesis, the problem of discontinuities at concatenation points arises. If only few concatenations of units occur the degradation of intelligibility and naturalness due to discontinuities may be negligible. For concatenative synthesis with a small inventory, however, and in specific but important cases (like, e. g., synthesis of texts including a large number of proper names) the smoothing of concatenations is of great importance for unit selection synthesis, too.

At concatenation points discontinuities in the signal and of the signal phase, as well as discontinuities of fundamental frequency, spectral content, signal amplitude, etc. can occur. Particularly for synthesis with an inventory with one instance of each segment (like common diphone or demi-syllable based concatenative synthesis), *pitch-synchronous processing* to ensure *signal phase coherence*, as well as interpolation of fundamental frequency, spectral content and signal amplitude is required. In text-to-speech systems the prosody generation module commonly provides a smooth fundamental frequency trajectory, hence no additional smoothing is necessary. For LP based synthesis algorithms, spectral content can be smoothed in a straightforward way by interpolation of the LP re-synthesis filter, preferably in the log-area-ratio or line-spectral-frequency domain [Ran99, WM00].

For unit selection synthesis smooth transitions at concatenation points can be facilitated by minimizing the join costs, i. e., by selection of good matching segments. However, since join costs are not the only criterion for the selection of units there always may occur concatenation points with fundamental frequency and spectral discontinuities. If no signal processing is applied the resulting discontinuities may seriously degrade naturalness. Many discontinuities mentioned above can easily be avoided with the glottal waveform generation methods used in model-based synthesis (sect. 2.3). With these signal generation methods, on the other hand, it is often difficult to re-generate variations depending on speaker identity, speaking style, or emotional state of the speaker. Particularly in signal generation models parameterized by only a small number of parameters (like the LF model), such variations can be captured by the model only to a rather limited extent.

Chapter 3

Nonlinear function models

For speech synthesis with the oscillator model, the correct identification of a nonlinear function $f(\cdot)\,;\,\mathbb{R}^N \rightarrow \mathbb{R}$ (as will be used in sect. 4.2, eq. 4.7 on page 50) by a *nonlinear function model* $\hat{f}(\cdot)$ is crucial. Since the nonlinear function of the oscillator model is applied in a high-dimensional space ($N \geq 3$), and thus is difficult to analyze and visualize, we will examine several nonlinear function models in a one-dimensional modeling task in this chapter. The analysis shall be, however, linked to the suitability of a nonlinear function model for speech prediction and for the application in the oscillator model.

The nonlinear function model should achieve a reasonably low prediction error on training and test signals. Moreover, it shall provide good generalization properties to ensure a generally stable behavior of the oscillator[1]. Several attempts to speech prediction and synthesis with different realizations for the nonlinear mapping function are reported in the literature. In most cases the oscillator model has been applied to the modeling of stationary sounds, e. g., sustained vowels. Here, a large enough number of training examples is available even for nonlinear function models with a large number of parameters.

Aiming at synthesis of non-stationary speech signals, the training of the nonlinear function has to be possible with speech samples on the time scale of the stationarity of the speech signal. For linear prediction, e. g., a speech signal is considered stationary on a time scale in the range of 10 to 20 ms, i. e., for only 160 to 320 samples (for a sampling rate of 16 kHz). This time scale is also the upper bound for stationarity when modeling the full speech signal by means of an stationary oscillator. To find a satisfactory nonlinear function model for a stable oscillator from a possibly small number of training data, we will account for the complexity of function models.

Furthermore, the nonlinear function model shall also promote the generation of transitions between successive models over time. To this end a parameterization is preferable where the model parameters derived from different sounds are related to each other so as to, e. g., facilitate updating of the model over time and model parameter interpolation.

Concerning the concept of modeling the oscillations excited by the vocal folds we have to consider that the speech signal will also consist of other excitations in general. Clearly, for unvoiced speech no excitation by vocal fold oscillations is present. For mixed excitation speech, e. g., voiced fricatives, the oscillatory signal is mingled with a strong noise-like signal component. But even for voiced speech signals like vowel signals a certain noise-like component is generally present in the speech signal. Thus, albeit we are dealing with clean signals recorded in a studio environment without ambient noise we have to make sure that the model training is robust with respect to the noise-like signal components.

In the following we will examine several nonlinear function models regarding their suit-

[1] For the meaning of “stable behavior” of the oscillator see sect. 4.2.

ability for speech prediction and for the oscillator model. We turn our attention specifically to the generalization properties of the nonlinear function models, as well as to the possibility to train a model with low number of training data in the presence of noise. The models will be tested for regression – i. e., fitting a curve from noisy data – on the one-dimensional function $y = \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ with additive Gaussian noise, so that the model behaviors can be illustrated graphically.

3.1 Radial basis function (RBF) networks

Radial basis function networks are, like multi-layer perceptrons (sect. 3.2.2), one kind of artificial neural networks (ANNs). The evolution of ANNs was inspired by the finding that the human brain computes in an entirely different way than a sequential digital computer. In an ANN complex nonlinear parallel computing takes place, and although one may doubt that an adequate model for the human brain can be built from ANNs, specific realizations of both RBF networks and multi-layer perceptrons have been proved to offer universal approximation capabilities for nonlinear functions [HSW89, HSW90, Hay94].

ANNs commonly consist of a number of similar *units (neurons, processing elements)* connected in the form of *layers*, and *weights* associated with the connections. The parameters of the ANN – i. e., the weights, but often also the parameters for the nonlinear functions at the units – are optimized to yield an approximation of a system function using examples of the input and corresponding output data of the system that should be modeled. A number of different algorithms for this *training* process are possible, depending on the network structure but also on the specific optimization criterion used.

ANNs have been applied to the modeling of the nonlinear function that governs the dynamics of speech in a low-dimensional phase space embedding for speech modeling, prediction, and synthesis in [Bir95, YH95, Bir96, Kub96a, BBK97, MM99, Man99, NPC99, BMM99, MM01, CMU02a, CMU02b].

RBF networks provide a global model for a nonlinear function with good local modeling if unimodal (e. g., Gaussian) basis functions are used. 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 an RBF. The network output is computed as the weighted sum of the units' outputs. The modeling properties are influenced by the number of centers, their positions, by the shape of the basis functions (e. g., the variance for a Gaussian basis function), and by the training algorithm used to determine the network weights.

The input-output relation of an RBF network is[2]

$$y = \hat{f}_{\mathrm{RBF}}(\boldsymbol{x}) = \sum_{i=1}^{N_c} w_i \, \varphi_i(\boldsymbol{x}) \quad , \qquad \mathbb{R}^N \rightarrow \mathbb{R} \quad , \tag{3.1}$$

operating on a N-dimensional input vector $\boldsymbol{x}$ to compute a scalar output y, with the basis functions $\varphi_i(\cdot)$, weighting coefficients w_i, and the number of basis functions and weighting coefficients N_c.

The most frequently used Gaussian radial basis function is defined by

$$\varphi_i(\boldsymbol{x}) = \exp\left(-\frac{1}{2} \frac{\|\boldsymbol{x} - \boldsymbol{c}_i\|^2}{\sigma_i^2} \right) \quad , \tag{3.2}$$

[2] 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 speech signals in general have zero mean, we do not make use of this bias term.

with the network center positions $\boldsymbol{c}_i$ and the variances σ_i^2 of the Gaussian basis functions. Other examples of basis functions can be found in the literature, e. g., in [Hay94]. An RBF network is a special case of kernel based function approximation with rotationally invariant kernels $\varphi_i(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{c}_i) = K(\|\boldsymbol{x} - \boldsymbol{c}_i\|)$. A schematic of a basic radial basis function network is depicted in fig. 3.1.

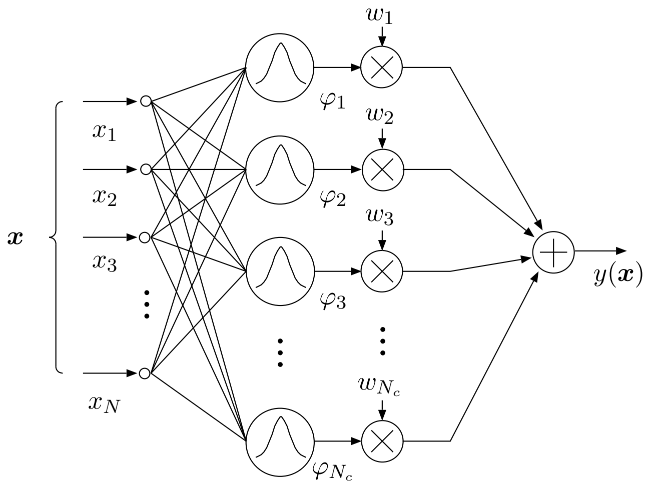


Figure 3.1: Radial basis function network

The network center positions $\boldsymbol{c}_i$ can be fixed *a priori* or depend on the training data. In the latter case the center positions are either set equal to the input training data, or a subset of the input training data, or are optimized during the training procedure (movable centers).

The widths of the basis functions should be chosen in a way such that the input space is appropriately covered by basis functions. This means that the width shall be related to a maximum distance between center positions [Hay94]. For centers depending on the input data and movable centers no maximum distance can be stated *a priori* and the choice of basis function widths relies on heuristics. If, however, the centers are fixed – for example on an equally spaced hyper-grid (fig. 3.2 on the following page) – the influence of basis function width on the coverage of the input space by the RBF network can be exemplified as in fig. 3.3 on page 23.

Here the basis function width is related to the maximum distance between adjacent network centers, which is the distance along the diagonal in the N-dimensional hyperspace

$$d_{\text{diag}} = d_{\text{grid}} \sqrt{N} \ , \tag{3.3}$$

with a spacing of the hyper-grid d_{grid} between centers in each dimension. The network output function for Gaussian basis functions with centers on a hyper-grid at unit intervals ($d_{\text{grid}} = 1$) in the range of $x \in [-3, 3]$ is calculated for unit weights. Since the centers are equally spaced the variances of the Gaussian basis functions can be chosen equal for all centers $\sigma_i^2 = \sigma_g^2$. The width d_{BF} of the Gaussian basis functions shall be defined as twice the standard deviation,

$$d_{\text{BF}} = 2\sigma_g \ . \tag{3.4}$$

As can be seen a small basis function width leaves ‘gaps’ in the output for unit weights over the input region covered by the centers (fig. 3.3, left column). This results in a high prediction error for unseen data, e. g., in a cross-validation procedure. A high prediction error for a basis function width below a certain threshold has been encountered, e. g., in [Man99] for

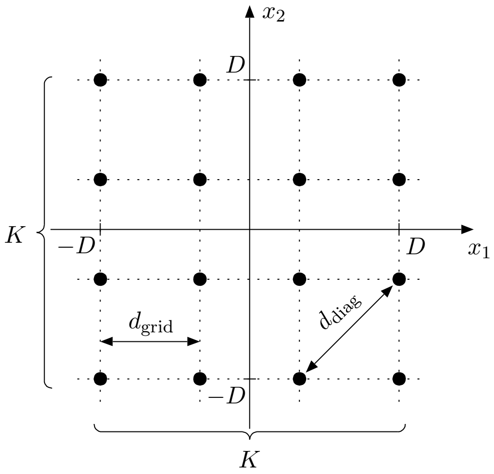


Figure 3.2: Two-dimensional example of RBF network centers (marked by the full circles) on equidistant hyper-grid. A number of $K = 4$ grid-lines are occupied by network centers. The position of the outer grid-lines D is chosen according to the maximum absolute input value. Distance between the grid-lines is $d_{\text{grid}} = 2D/(K-1)$. The diagonal distance between adjacent centers in the N-dimensional case is $d_{\text{diag}} = 2D/(K-1)\sqrt{N}$.

speech signal prediction and in [CMU02a, CMU02b] for predictions of signals from the Lorenz system. Too high a basis function width issues strongly overlapping basis functions and thus a high amount of interaction between different regions in input space. The network response for unit weights suffers a severe falloff towards the edges of the input region covered by network centers due to the missing influence from basis functions outside the input region (fig. 3.3, right column).

A reasonably flat network response is obtained with a basis function width equal to the distance between adjacent network centers along the diagonal, as depicted in fig. 3.3, center column. The choice of $d_{\text{BF}} = d_{\text{diag}}$ thus seems advantageous in terms of input space coverage. Considering that the problem of gaps in input space for small basis function width is less stringent with increasing input space dimension (cf. fig. 3.3, first column) the basis function width could be chosen somewhat smaller than d_{diag} for higher input dimension, still providing adequate coverage.

The problem of input space coverage can be relieved by normalizing the basis functions [MD89]. For Gaussian basis functions, defining

$$\varphi_{i,\text{norm}}(\boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2}\frac{\|\boldsymbol{x}-\boldsymbol{c}_i\|^2}{\sigma_i^2}\right)}{\sum_{j=1}^{N_c}\exp\left(-\frac{1}{2}\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{\sigma_j^2}\right)}\ , \tag{3.5}$$

the output of one basis function is divided by the sum of all basis function outputs. Thus the network output for unit weights will always sum up to unity. The resulting *normalized radial basis functions* $\varphi_{i,\text{norm}}$, however, are no more rotationally invariant, i. e., no *radial* basis functions any more. Nevertheless, normalized radial basis function networks have been proved to be slightly superior in terms of prediction error and less sensible to the choice of basis function width compared to non-normalized RBF networks in the prediction of noisy data from a nonlinear system [CMU02a, CMU02b] and they have successfully been used for speech synthesis in [Kub96a].

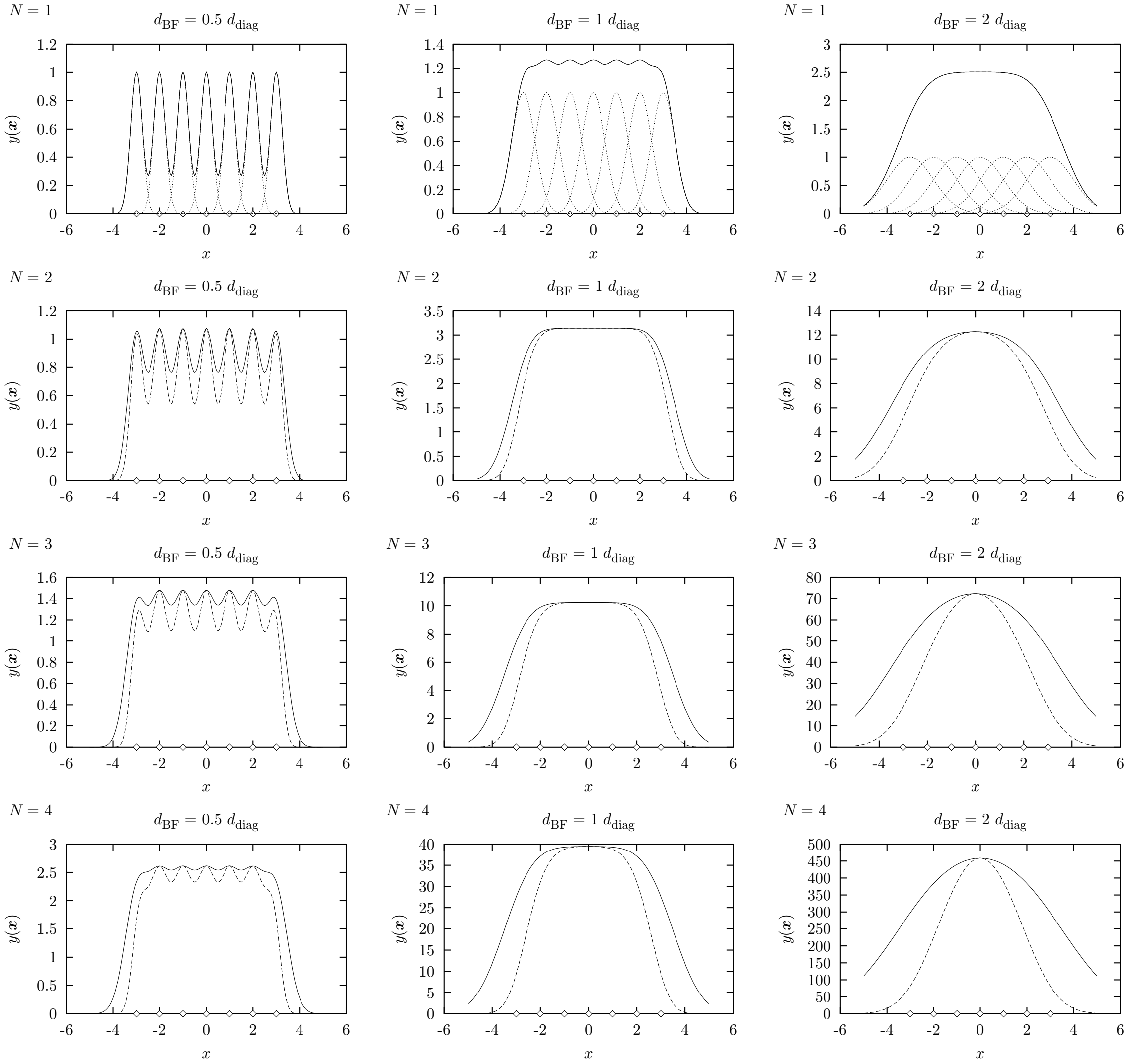


Figure 3.3: Choice of basis function width. For centers on a hyper-grid with unit spacing the network output y is shown for different basis function width d_{BF} and input space dimension N. Plotted are the network output functions of a Gaussian RBF network for unit weights (i.e., $w_i = 1$ for all i) along one axis of input space, $y(\boldsymbol{x}) = y([x, 0, \ldots, 0]^{\mathsf{T}})$ (solid line), and along the diagonal, $y(\boldsymbol{x}) = y([x, x, \ldots, x]^{\mathsf{T}})$ (dashed line). Center positions are indicated by the diamonds along the x axis. In the one-dimensional case (top row) the output of the individual basis functions is overlayed (dotted lines). RBF width is parameterized with the distance between centers along the diagonal in hyperspace, in this case of unit grid spacing $d_{\mathrm{diag}} = \sqrt{N}$. A small width of $0.5\, d_{\mathrm{diag}}$ results in an oscillating output (left column), especially for low dimension and along the diagonal, whereas a high width $2\, d_{\mathrm{diag}}$ results in a decay of the output function towards the edges of the input space region covered by the centers (right column). A reasonably flat response over the input space region covered by the centers is found for a width $d_{\mathrm{BF}} = d_{\mathrm{diag}}$ (center column).

To model a function given by example input-output pairs (training data) the appropriate RBF network weights w_i have to be found. Rewriting the network equation as a vector product yields

$$\hat{f}_{\mathrm{RBF}}(\boldsymbol{x}) = \sum_{i=1}^{N_c} w_i\, \varphi_i(\boldsymbol{x}) = \boldsymbol{w}^{\mathsf{T}} \boldsymbol{\varphi}(\boldsymbol{x}) \enspace , \tag{3.6}$$

with $\boldsymbol{\varphi}(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}), \ldots \varphi_{N_c}(\boldsymbol{x})]^{\mathsf{T}}$ and $\boldsymbol{w} = [w_1, w_2, \ldots w_{N_c}]^{\mathsf{T}}$. This equation is linear in the weights, and the solution for the unknown weights to minimize the mean squared prediction error on the training data

$$e^2 = \sum_{k=1}^{P} (\hat{f}(\boldsymbol{x}_k) - t_k)^2 \enspace , \tag{3.7}$$

with $k = 1 \ldots P$, P being the number of training samples, is

$$\hat{\boldsymbol{w}}_{\mathrm{MMSE}} = (\boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^{\mathsf{T}}\, \boldsymbol{t} \enspace , \tag{3.8}$$

with

$$\boldsymbol{\Phi} = \begin{bmatrix} \varphi_1(\boldsymbol{x}_1) & \cdots & \varphi_{N_c}(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{x}_P) & \cdots & \varphi_{N_c}(\boldsymbol{x}_P) \end{bmatrix} \enspace , \tag{3.9}$$

a $P \times N_c$ matrix composed of all basis functions outputs for all training input data vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_P$, and $\boldsymbol{t} = [t_1, \ldots, t_P]^{\mathsf{T}}$ a vector composed of the training output data (the targets). The expression on the right-hand side of eq. 3.8 contains the pseudo inverse $\boldsymbol{\Phi}^{\dagger}$ of matrix $\boldsymbol{\Phi}$ that can be computed as $\boldsymbol{\Phi}^{\dagger} = (\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathsf{T}}$ if $(\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi})^{-1}$ exists, i. e., if $\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}$ is not rank-deficient.

Thus, training – i. e., estimation of the weights – of a radial basis function network, with a priori fixed center positions and basis functions, is a linear problem – the weights are found by solving a system of linear equations –, in contrary to the training of other ANN realizations. The weights can be estimated in a one-step procedure and the resulting network has the *best approximation property*, meaning that for given input-output training examples, an RBF network with weights according to eq. 3.8 yields a better approximation of the *training output data* in terms of mean squared error (MSE) than with all other possible sets of network weights [PG90][3].

Generally, an RBF network achieves better prediction the higher the number of centers N_c, and yields perfect prediction at the positions of the training data for $N_c = P$. Here the weights can be found using the regular inverse of the now square matrix $\boldsymbol{\Phi}$ by $\hat{\boldsymbol{w}}_{\mathrm{MMSE}} = \boldsymbol{\Phi}^{-1}\, \boldsymbol{t}$. However, perfect prediction of the training data obviously implies over-fitting – i. e., modeling not only of the target function, but also of the concrete realization of additive noise in the training data – and thus bad noise robustness. In general, the number of training examples P should be considerably higher than the number of network parameters N_c. And although a higher number of centers suggests better modeling properties for a complicated function, for the prediction of noisy chaotic signals of infinite length, RBF networks have been proved to yield optimal prediction with a finite number of network centers [LLW01]. Simulation results also suggest that with a finite number of noisy training data the optimal number of network centers can be quite low, depending on the actual data generating system. But in general an optimum number of network centers for a given task has to be searched for, e. g., by cross-validation.

[3] For multi-layer perceptrons (MLP, cf. 3.2.2 on page 38) the best approximation property does not apply in general, and a training based on gradient descent has to be run several times until an appropriate model is found, because gradient descent training may merely converge to a local minimum of the error function.

In fig. 3.4 on the next page modeling of the sinc function by an RBF network, trained as described above, is depicted. In the noiseless case with training data covering the full range of centers (case (a)) a quite precise prediction of the target function is achieved with this network. In the noisy case (b) still a reasonably good prediction is achieved, although partial over-fitting of the noisy training data can be observed. Over-fitting is more severe in case (d), since here a smaller number of training data is used. In the case of incomplete training data (case (c) and (d), training data do not cover the full range of network centers) the network output function displays high peaks outside the training data range. This means that unreasonably large values are found for the respective network weights. This can either happen for centers that are far away from the input training data if the network tries to capture a small variation in the output training data, or if close centers receive large weights with different signs whose influence cancels at the position of the training data. This problem can be relieved using regularization where a preference for choosing small weights and a smooth function approximation is introduced (see section 3.1.2 below). The former problem can also be alleviated by pruning centers that are distant from all input training data prior to network weight learning.

3.1.1 Pruning centers distant from the training data

When training an RBF network on a speech signal embedded in phase space, often problems are encountered computing the pseudo-inverse in eq. 3.8. This is due to a high condition number, i. e., a high ratio of the largest over the smallest eigenvalue of the matrix $\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}$. The matrix is almost singular or *rank-deficient*, and instead of the inversion only a minimum norm solution can be searched for.

Even if the number of training samples and thus the number of rows in $\boldsymbol{\Phi}$ is increased the condition number of $\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}$ does not significantly change for stationary vowel signals [Man99]. Taking a closer look at the situation in phase space we see that the trajectory of a stationary vowel sound only fills a small subspace of $\mathbb{R}^N$ (cf. fig. 4.9 on page 57 or fig. 4.21 on page 73). The outputs of the basis functions corresponding to centers far away from this subspace will be close to zero for all training input data. Consequently, the corresponding columns in $\boldsymbol{\Phi}$ consist of small values only. In particular, if the centers are fixed a priori and distributed over the input space, for example on a hyper-lattice, a high condition number for $\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}$ (in the range of 10^{12}) is observed.

For centers located on a subset of the training input data, the condition number is somewhat lower (in the range of 10^9). However, if we want to maintain the same center positions for different speech sounds we have to deal with a set of a priori fixed centers. Here we can relieve the ill-conditioning by restricting the training process to centers in the vicinity of the actual trajectory and excluding centers far away from the trajectory from the training process. This is a simple kind of *network pruning*, where unnecessary parts of an ANN are removed. The effect on the network output function is exemplified in fig. 3.5 on page 27. Here the training data is limited to the interval $x \in [-3, 3]$ and the RBF centers are set equally spaced in $x \in [-5, 5]$. Training the full network on the sinc function may yield large weight values for some centers outside the range of the training data due to the network trying to capture small variations of the output training data. The resulting network output function thus displays large (negative) peaks in $x \in [-5, -3]$ and $x \in [3, 5]$, fig. 3.5 (a). If the centers outside the training data range are pruned before network training the network output function gracefully decays to zero for test input data outside the range of training input data, fig. 3.5 (b).

This kind of pruning reduces the condition number for speech signals to the range of 10^9 – the same gain as for taking network centers on a subset of the input training data. The computational load for the matrix inversion is reduced, too. One may start out with a high number of centers (several thousand) distributed in, say, the unit cube of phase space, and

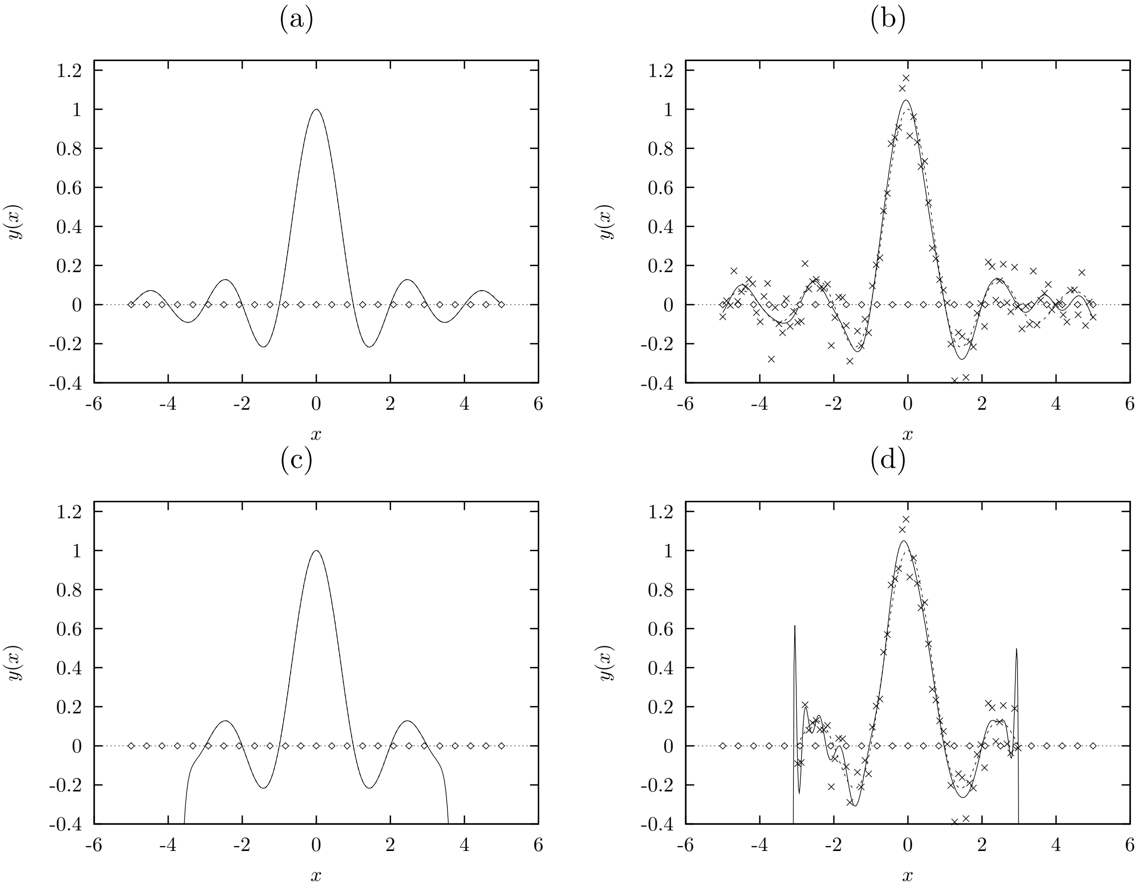


Figure 3.4: Modeling of the sinc function by an RBF network. The RBF network has $N_c = 25$ centers in $x \in [-5, 5]$ with Gaussian basis functions and a basis function width of $d_{\mathrm{BF}} = 0.41$ ($= d_{\mathrm{grid}} = d_{\mathrm{diag}}$, since $N = 1$). The sinc function is given (a) by $P = 100$ training samples equally spaced in $x \in [-5, 5]$ without noise, (b) with additive Gaussian noise and an SNR of 10 dB, and (c) for $P = 60$ training samples in $x \in [-3, 3]$ without noise and (d) with an SNR of 10 dB. In the plots the solid line represents the network output function, the dashed line is the target sinc function, crosses indicate the training samples in the noisy case, and the diamonds along the x axis represent network center positions. In the noiseless cases (a) and (c) the solid line (network output function) perfectly conceals the dashed line (target function) in the range of training data.

restrict the training, i. e., the matrix inversion to only a subset of the original centers (typically several hundred for speech signals).

Thus, the pruning of network centers with a certain minimum distance from *all* input training data both relieves the problem of ill-conditioning during network training and prevents the network weights from taking unreasonably large values resulting in unwanted peaks in the network output function. If it happens that we should need the pruned centers during non-stationary synthesis, e. g., to interpolate between different sounds, we can always consider the pruned centers still present in the network with a corresponding weight value of zero.

Alas, as will be shown in sect. 4.3.2, pruning does not suffice to find a good nonlinear function model for a practical (i. e., stable) oscillator, in general. A method that improves noise robustness, generalization and further reduces the condition number is regularization, described in the next section.

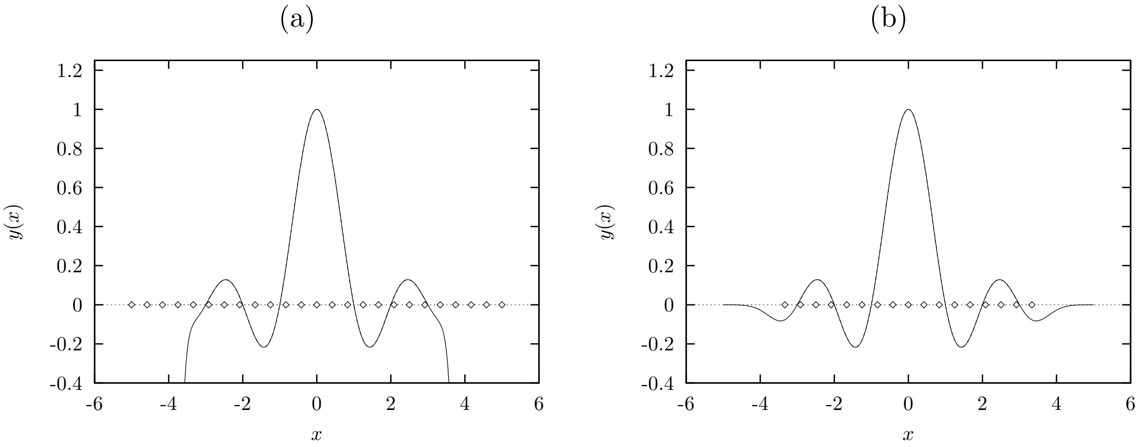


Figure 3.5: Effect of pruning distant centers of an RBF network before network training. For this example the target sinc function is provided with $P = 60$ training data points in the range $x \in [-3, 3]$. The RBF network is set up with $N_c = 25$ centers equally spaced in $x \in [-5, 5]$. (a) When training the full network some of the weights for centers outside $[-3, 3]$ receive large values and the network output function displays large peaks, too. (b) By pruning the network centers outside $[-3, 3]$ before network training the network output function gradually decays towards zero outside $[-3, 3]$.

3.1.2 Regularization

The estimation of the network weights in eq. 3.8 involves the inversion of the matrix $\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}$. In certain cases – particularly for *a priori* fixed centers (see above), or if some training input data are close to others (like for almost periodic voiced speech signals) – this matrix can be ill-conditioned. If so, the estimated weights can vary considerably for small changes in the training data (e. g., by noise). The problem of system identification is said to be *ill-posed* and regularization theory should be applied [TA77]. Regularization derives a well-posed approximation of a possibly ill-posed problem and prevents the RBF model from over-fitting noisy training data [PG89]. In particular, regularization improves the modeling of noisy signals from nonlinear systems with the oscillator model in the way that the dimension of the signal and the Lyapunov exponents are reproduced better than without regularization, which is shown for the modeling of the Lorenz system in [HP98b, Ran03].

Regularization of matrix inversion

Applying regularization to matrix inversion for weight estimation of an RBF network (or any other linear-in-the-weights model), the equation for the weights eq. 3.8 turns into [TA77]

$$\hat{\boldsymbol{w}}_{\text{reg}} = (\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi} + \lambda \boldsymbol{I})^{-1}\boldsymbol{\Phi}^\mathsf{T}\,\boldsymbol{t} \quad , \tag{3.10}$$

with λ being the *regularization factor*. With $\lambda = 0$ there is no regularization, increasing λ yields an increasingly well-posed problem, making the estimated values for the weights less sensitive to noise in $\boldsymbol{\Phi}$. The addition of the matrix $\lambda\boldsymbol{I}$ ensures a full rank of the matrix that has to be inverted, thus reducing its condition number with increasing values of λ.

Regularization of the network function

Regularization of nonlinear function models can also be achieved by posing a constraint on the nonlinear function model $\hat{f}(\cdot)$, e. g., that it should realize a *smooth* function. This is achieved by adding a term containing a differential operator $\mathcal{P}$ applied to $\hat{f}(\cdot)$ to the mean squared prediction error in eq. 3.7, yielding a cost function

$$H[\hat{f}] = \sum_{k=1}^{P} (\hat{f}(\boldsymbol{x}_k) - t_k)^2 + \lambda \|\mathcal{P}\hat{f}\|^2 \ , \tag{3.11}$$

where $\|\cdot\|$ is a norm on the function space $\hat{f}$ belongs to. For network centers equal to the training data (i. e., $\boldsymbol{c}_i = \boldsymbol{x}_i, N_c = P$) the following linear equation for the weights [PG89] is found:

$$(\boldsymbol{G} + \lambda \boldsymbol{I})\boldsymbol{w} = \boldsymbol{t} \ . \tag{3.12}$$

$\boldsymbol{G}$ is a matrix of Green's functions applied to the input training data, $\boldsymbol{G}_{ij} = G(\boldsymbol{x}_i, \boldsymbol{x}_j)$. For a specific choice[4] of the differential operator $\mathcal{P}$ the Green functions are multivariate Gaussians and an RBF network with Gaussian basis functions is obtained. The optimum weights for the network are found by solving eq. 3.12 for $\boldsymbol{w}$ with

$$\boldsymbol{G} = \begin{bmatrix} \varphi_1(\boldsymbol{x}_1) & \cdots & \varphi_P(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{x}_P) & \cdots & \varphi_P(\boldsymbol{x}_P) \end{bmatrix} \ , \tag{3.13}$$

a $P \times P$ matrix composed of the output of the Gaussians basis functions – or the Green functions – with variance σ as chosen in the differential operator $\mathcal{P}$, centered at the input training data positions, i. e.,

$$\boldsymbol{G}_{ij} = \varphi_i(\boldsymbol{x}_j) = \exp\left(-\frac{1}{2} \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2} \right) \ . \tag{3.14}$$

Generalized radial basis functions (GRBFs)

If we do not want the number of network centers N_c to be equal to the number of training samples P an extension to a number of $N_c < P$ arbitrary RBF center positions yields the *generalized radial basis function (GRBF)* network [PG89], with weights

$$\hat{\boldsymbol{w}}_{\mathrm{GRBF}} = (\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} + \lambda \boldsymbol{\Phi}_0)^{-1} \boldsymbol{\Phi}^{\mathsf{T}}\, \boldsymbol{t} \ , \tag{3.15}$$

with $\boldsymbol{\Phi}$ the $P \times N_c$ matrix in eq. 3.9, and an $N_c \times N_c$ matrix composed of the output of the radial basis functions applied to the network center positions:

$$\boldsymbol{\Phi}_0 = \begin{bmatrix} \varphi_1(\boldsymbol{c}_1) & \cdots & \varphi_{N_c}(\boldsymbol{c}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{c}_{N_c}) & \cdots & \varphi_{N_c}(\boldsymbol{c}_{N_c}) \end{bmatrix} \ . \tag{3.16}$$

Again, the amount of smoothing is determined by the regularization factor λ. For Gaussian basis functions the matrix $\boldsymbol{\Phi}_0$ has all ones on the diagonal and positive values smaller than one for the off-diagonal entries. Comparing eq. 3.15 to eq. 3.10 we see that the non-zero off-diagonal entries in $\boldsymbol{\Phi}_0$ make the only difference between regularization for matrix inversion and GRBFs.

[4] $\|\mathcal{P}\hat{f}\|^2 = \int_{\mathbb{R}^N} d\boldsymbol{x} \sum_{m=0}^{\infty} \frac{\sigma^{2m}}{m! 2^m} (P^m \hat{f}(\boldsymbol{x}))^2$ with $P^{2m} = \nabla^{2m}, P^{2m+1} = \nabla\nabla^{2m}$, [PG89, pp. 21ff]

On the one hand regularization avoids over-fitting of the training data and thus improves noise robustness. On the other hand the weights are also prevented from taking unreasonably high values in regions with no training data. Figure 3.6 depicts the network output function of a regularized RBF network and regression of the sinc function. The only effect in the noiseless case (a) is an under-estimation of the function peaks since the regularization implies a preference for smaller weight values (this can and should of course be avoided by reducing regularization in the low noise cases). In the noisy cases (b) and (d), however, the effect of smoothing by regularization can be observed, cf. fig. 3.4 (b) and (d). For incomplete training data (cases (c) and (d)) regularization results in a decay of the network output function towards zero for the regions not covered by the training data.

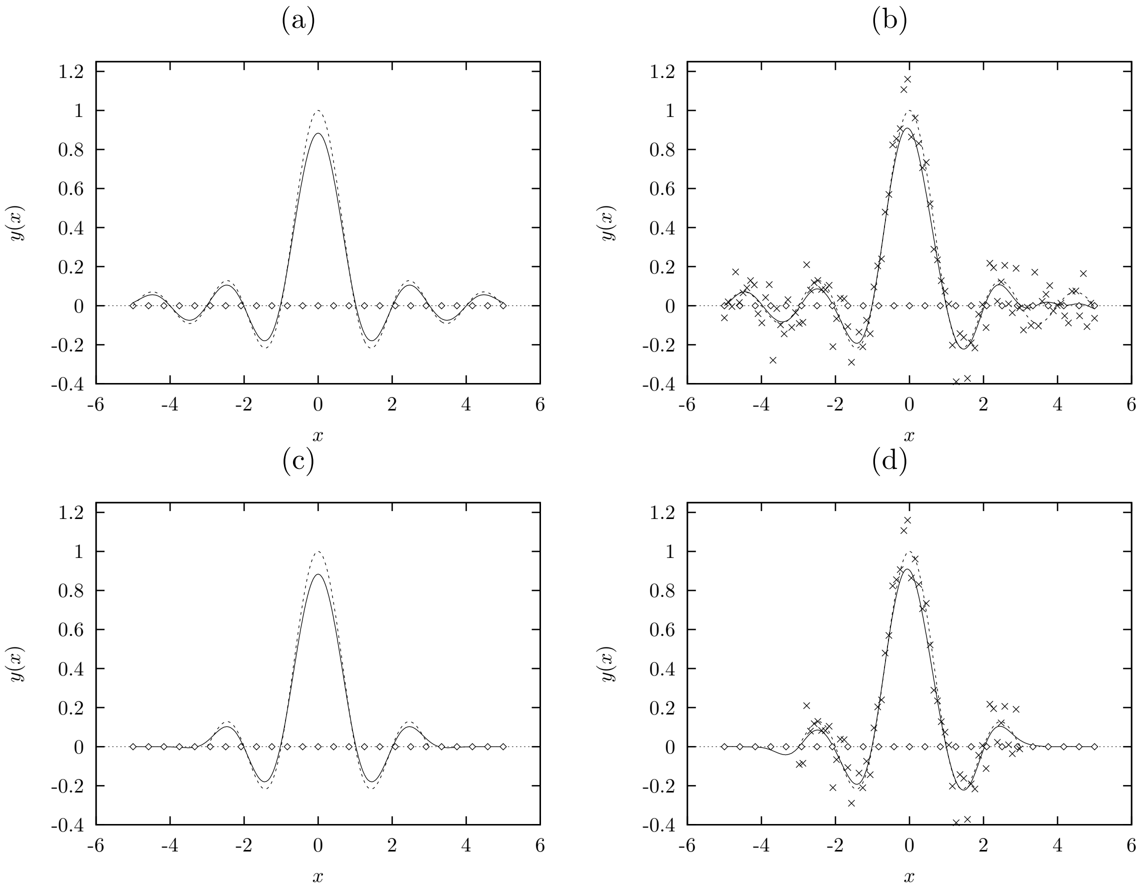


Figure 3.6: Modeling of sinc function by a regularized RBF network (eq. 3.10, $\lambda = 1$). Network parameters and all other experimental conditions are the same as in fig. 3.4 (a)-(d) on page 26. We can observe a general underestimation of target function peaks, smoothing of the network output function in the noisy case (cf. fig. 3.4, (b) and (d)), and a fast decay towards zero output for regions not covered by training data.

As we will see in sect. 4.3.2 on modeling of speech signals with the oscillator model, regularization is required to achieve stable oscillator behavior. However, the more regularization is applied the more of the detailed structure of the signal trajectories is smoothed out, and high-frequency components and cycle-to-cycle variations are no more reproduced. It is thus necessary to determine an appropriate amount of regularization, based on the nature of the signal to be modeled.

Cross-validation

The regularization factor λ should be chosen according to the noise variance in the training data. However, the noise variance is generally not known in advance. One way to find an optimal regularization factor is cross-validation on unseen training data. For cross-validation, a part of the training data is set aside for performance validation, and the value for the unknown model parameter – in our case the regularization factor – is chosen such that it optimizes the performance, e. g., by minimizing the prediction error on the validation set. To this end, network training and validation has to be performed several times with varying regularization factors.

Cross-validation is often run multiple times on different splits of the entire training data, for example in k-fold cross-validation [Sto74] by partitioning the training data set in a number of k subsets of equal size, and performing k cross-validation runs. In each run $k-1$ subsets of the entire data are used for training and the remaining subset for testing. The value of the model parameter that has to be optimized can then be chosen, e. g., as the mean of the optimal values from the k cross-validation runs. In this way the drawback of cross-validation – that a part of the training data has to be set aside for validation – can be relieved.

In any case, after finding the optimal parameter value, for the final network training all training data can be used. However, the value for the regularization factor is always determined using only a part of the available training data. On contrary, the two Bayesian algorithms introduced below use all available training data samples to determine the optimal regularization factor.

3.1.3 Bayesian learning of regularization factor and noise variance

In the Bayesian approach for model comparison[5] and optimization of regularization [Mac92a, Mac92c, Mac92b, Nea96] the unknown parameters of a nonlinear function model, which are the network weights for our RBF network, are considered random variables. A suitable *prior* distribution is chosen for the probability density function (pdf) of these parameters. To that end additional *hyper-parameters* for the characterization of the pdfs are introduced, which are regarded as random variables, too. The optimal weight values are then chosen by maximizing the *posterior* pdf for the model parameters – weights and hyper-parameters – given the training data samples. In the algorithms considered here, this is achieved by an iterative procedure in the manner of the expectation-maximization (EM) algorithm [DLR77, Moo96].

Here we consider two methods applicable to an RBF network. In this section, the simultaneous estimation of the weights, the regularization factor, and the noise variance in the training data is described [Mac92a]. The next section 3.1.4 deals with a closely related training algorithm known as the *relevance vector machine (RVM)* [Tip01], additionally allowing for the pruning of individual network centers. Here, only a short overview of Bayesian learning is given, a more detailed derivation of the Bayesian training process for RBF networks is given in App. B.

Generally, for the Bayesian approach to function model learning, we assume that the training output data are due to an additive noise model

$$t_k = f(\boldsymbol{x}_k) + \epsilon_k, \quad p(\epsilon) = \mathcal{N}(0, \sigma_n^2) \ , \tag{3.17}$$

with additive zero-mean Gaussian noise samples ϵ_k with variance σ_n^2, and the unknown nonlinear function $f(\cdot)$. For learning the parameters of an RBF network the function $f(\cdot)$ is parameterized as in eq. 3.1, i. e., we assume that the training data have been produced by

[5] In [Mac92a], besides the Bayesian optimization of network parameters and regularization, the Bayesian optimization of hypothesis $\mathcal{H}$ (network complexity), set of basis functions $\mathcal{A}$, and of the regularizer $\mathcal{R}$ is considered, too. Here, we restrict the description to one specific choice of $\mathcal{H}$, $\mathcal{A}$, and $\mathcal{R}$.

a nonlinear function that can be parameterized as an RBF network, and additive Gaussian noise.

The function parameters, in our case the network weights $\boldsymbol{w}$, are characterized by a prior pdf. For a fixed network structure – i. e., a fixed number of basis functions, fixed centers and widths – regularization is introduced by stating a preference for small weight values, yielding smoother network functions for a given target function, by choosing the prior pdf for the weights as a zero-mean Gaussian distribution with variance α^{-1}:

$$p(\boldsymbol{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{-\frac{N_c}{2}} \exp\left(-\frac{\alpha}{2}\|\boldsymbol{w}\|^2\right) \ , \tag{3.18}$$

introducing one additional hyper-parameter α. To complete the setting for the Bayesian model, we have a second hyper-parameter which is the variance of the additive noise σ_n^2 [Mac92a][6].

The Bayesian choice for the network parameters is given by the values of the parameters $\boldsymbol{w}$, α, and σ_n^2 maximizing the posterior pdf $p(\boldsymbol{w}, \alpha, \sigma_n^2|\boldsymbol{X}, \boldsymbol{t})$:

$$(\boldsymbol{w}, \alpha, \sigma_n^2)_{\mathrm{bay}} = \arg\max(p(\boldsymbol{w}, \alpha, \sigma_n^2|\boldsymbol{X}, \boldsymbol{t})) \ , \tag{3.19}$$

where $\boldsymbol{X}$ represents the collected input data vectors.

The maximization of eq. 3.19 cannot be accomplished analytically, however, an efficient iterative algorithm is available. The derivation of the algorithm is detailed in App. B. The important first step to derive the iterative algorithm is a decomposition of the posterior pdf for the unknown parameters in eq. 3.19 according to

$$p(\boldsymbol{w}, \alpha, \sigma_n^2|\boldsymbol{X}, \boldsymbol{t}) = p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{t}, \alpha, \sigma_n^2)\, p(\alpha, \sigma_n^2|\boldsymbol{X}, \boldsymbol{t}) \ . \tag{3.20}$$

The first iteration step is the computation of the parameters of the posterior pdf for the weights from known hyper-parameters, which is the first term on the right-hand side in eq. 3.20. Due to the choice of a Gaussian prior pdf for the weights (eq. 3.18), and the assumption of additive Gaussian noise, the posterior pdf for the weights is a multivariate Gaussian distribution, too, with means $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$:

$$\begin{gathered} p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{t}, \alpha, \sigma_n^2) = (2\pi)^{-\frac{N_c}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{w}-\boldsymbol{\mu})\right) \ , \\ \boldsymbol{\mu} = \frac{1}{\sigma_n^2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{t} \ , \qquad \boldsymbol{\Sigma} = \left(\frac{1}{\sigma_n^2} \boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{\Phi} + \alpha \boldsymbol{I}\right)^{-1} \ . \end{gathered} \tag{3.21}$$

The second iteration step is the update of the hyper-parameter values α and σ_n^2, according to equations maximizing the second term on the right-hand side of eq. 3.20. The derivation of the update equations used below, which follows [Mac92a], is given in detail in App. B.

The complete iterative algorithm for Bayesian optimization of weights and hyper-parameters – and thus of regularization – is summarized by the following equations:

$$\begin{aligned} \boldsymbol{\Sigma}^{(i)} &= \left(\frac{1}{\sigma_n^{2\,(i)}} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \alpha^{(i)} \boldsymbol{I}\right)^{-1} \ , \\ \boldsymbol{\mu}^{(i)} &= \frac{1}{\sigma_n^{2\,(i)}} \boldsymbol{\Sigma}^{(i)} \boldsymbol{\Phi}^T \boldsymbol{t} \ . \end{aligned} \tag{3.22}$$

[6] In some references the second hyper-parameter is denoted by β, representing the inverse of the noise variance: $\beta = \sigma_n^{-2}$.

$$\begin{aligned}\gamma^{(i)} &= N_c - \alpha^{(i)}\,\mathrm{Trace}(\boldsymbol{\Sigma}^{(i)}) \ , \\ \alpha^{(i+1)} &= \frac{\gamma^{(i)}}{{\boldsymbol{\mu}^{(i)}}^T \boldsymbol{\mu}^{(i)}} \ , \\ \frac{1}{\sigma_n^{2\,(i+1)}} &= \frac{\left\|\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{\mu}^{(i)}\right\|^2}{P - \gamma^{(i)}} \ . \end{aligned} \tag{3.23}$$

Here, $i = 1, 2, \ldots, N_i$ is the iteration index. For model learning the hyper-parameters are initialized with values according to the prior knowledge, i. e., some low value for $\alpha^{(1)}$ corresponding to a broad prior distribution of the weights $\boldsymbol{w}$ – indicating that we do not know anything about the correct values for the weights a priori – and a 'reasonable guess' for the noise variance $\sigma_n^{2\,(1)}$. The number of iterations N_i necessary for convergence of the Bayesian learning algorithm depends on the training data. For the examples in this thesis, the iterations were stopped if both the value of the hyper-parameter α and the value of γ change less than 1% in an iteration. For the example sinc function, the iterative algorithm converges within as few as three or four iterations.

The weight values maximizing their Gaussian posterior distribution are of course the means $\boldsymbol{\mu}$, and the Bayesian choice for the optimal weights thus is:

$$\hat{\boldsymbol{w}}_{\mathrm{bay}} = \boldsymbol{\mu}^{(N_i)} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \underbrace{\alpha^{(N_i)}\,\sigma_n^{2\,(N_i)}}_{\lambda_{\mathrm{bay}}}\,\boldsymbol{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{t} \ , \tag{3.24}$$

and the resulting nonlinear function model is

$$\hat{f}_{\mathrm{bay}}(\boldsymbol{x}) = \hat{\boldsymbol{w}}_{\mathrm{bay}}^{\mathsf{T}}\,\boldsymbol{\varphi}(\boldsymbol{x}) \ . \tag{3.25}$$

$\boldsymbol{\varphi}(\boldsymbol{x})$ is the vector of basis function responses to the input vector, as in eq. 3.6.

Note, that – relating to regularization by matrix inversion (eq. 3.10) – the product of the hyper-parameter values after convergence takes the role of the regularization parameter, $\lambda_{\mathrm{bay}} = \alpha^{(N_i)}\,\sigma_n^{2\,(N_i)}$. Thus, for an RBF network, the Bayesian optimization of the hyper-parameters yields an optimization of the regularization parameter directly comparable to cross-validation for the regularized RBF learning methods in the last section 3.1.2.

Due to the probabilistic modeling, the output value $y = \hat{f}_{\mathrm{bay}}(\boldsymbol{x})$ of the RBF network (eq. 3.6) for an arbitrary input vector $\boldsymbol{x}$ can be accompanied by an estimation of the according variance $\sigma_y^2(\boldsymbol{x})$ of the pdf for the output value y of the RBF network. The variance $\sigma_y^2(\boldsymbol{x})$ after Bayesian learning of noise variance and regularization factor for the RBF model is [Tip01]:

$$\sigma_y^2(\boldsymbol{x}) = \sigma_n^{2\,(N_i)} + \boldsymbol{\varphi}(\boldsymbol{x})^{\mathsf{T}}\,\boldsymbol{\Sigma}^{(N_i)}\,\boldsymbol{\varphi}(\boldsymbol{x}) \ , \tag{3.26}$$

using the final estimate for the noise variance σ_n^2 and for the weights covariance matrix $\boldsymbol{\Sigma}$. The standard deviation $\sigma_y(\boldsymbol{x})$ of the predicted value $y(\boldsymbol{x})$ represents the uncertainty of the prediction and can be visualized as *error bars*.

Figure 3.7 on the facing page gives examples for the application of Bayesian learning to the sinc function for different amount of Gaussian noise in the training data. In table 3.1 the values estimated for the noise level and the according regularization factors are listed (also for uniformly distributed noise samples added to the training data). As can be seen the estimated noise variance is in good agreement with the correct value and the regularization imposed on the network results in a reasonably good function approximation even for low SNR.

In fig. 3.8 on page 34 the network output function for incomplete training data is depicted. In this case the predicted standard deviation of the target function (indicated by the 'error bar' functions $\hat{f}_{\mathrm{bay}}(x) \pm \sigma_y(x)$) grows considerably for regions not covered by the training data.

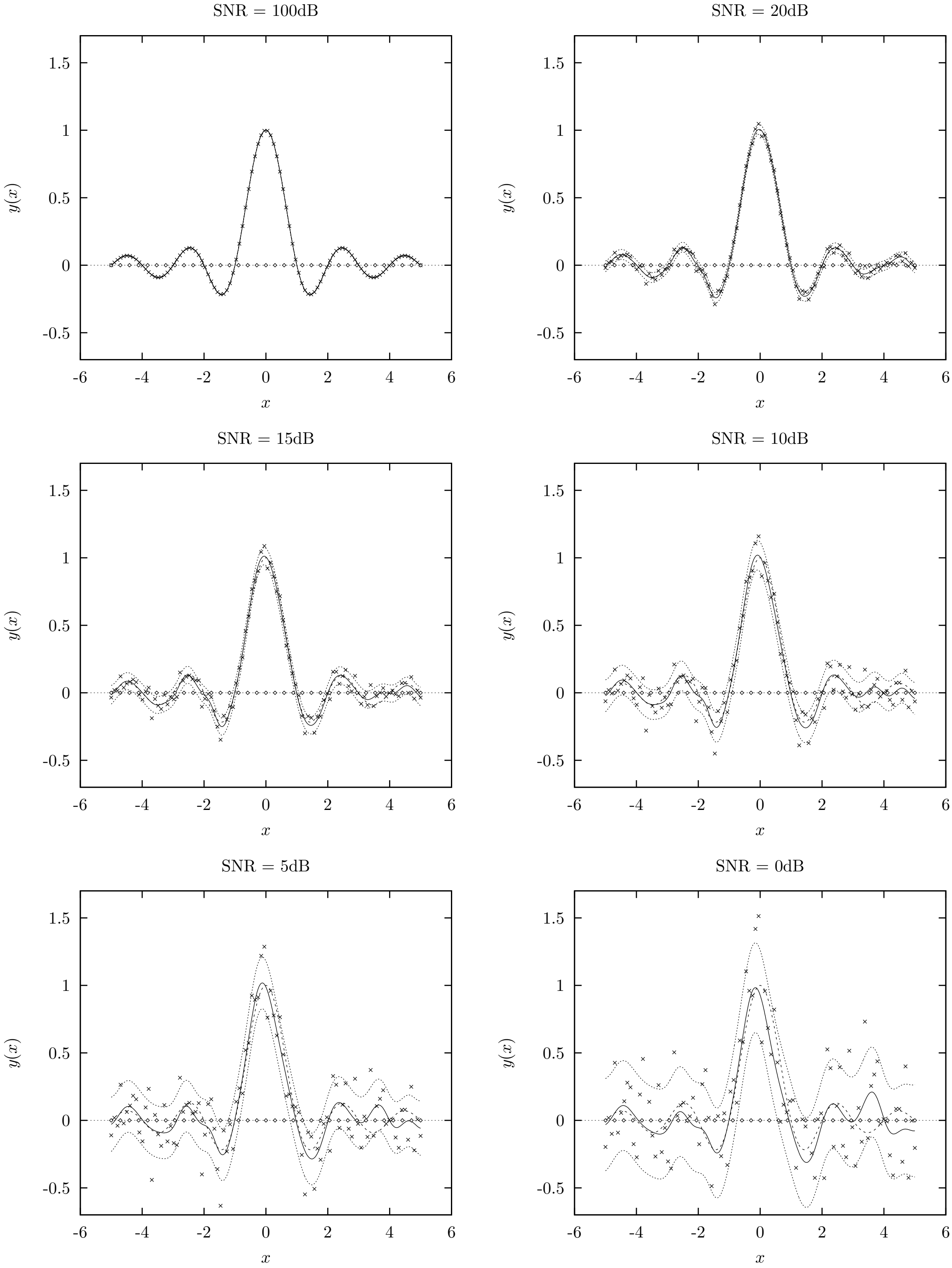


Figure 3.7: Bayesian learning of the sinc function for different noise levels. Shown are the target function (dashed line), the network output $\hat{f}_{\rm bay}(x)$ (solid line) along with the predicted 'error bars' $\hat{f}_{\rm bay}(x) \pm \sigma_y(x)$ (dotted lines), and the training data (crosses). The positions of network centers are indicated by the diamonds along the x axis. Network and training parameters are $P = 100$, $N_c = 35$, and Gaussian basis functions with $d_{\rm BF} = 0.5$.

Table 3.1: Parameters estimated by Bayesian learning of the sinc function. Training data are corrupted by additive Gaussian or uniform noise with variance σ^2. The estimated noise variance σ_n^2 does agree well with the actual noise variance and the amount of regularization is automatically increased with increasing noise level. The resulting network output functions for the Gaussian noise case are depicted in fig. 3.7 on the preceding page.

	Actual noise variance	Gaussian noise, estimates		Uniform noise, estimates	
SNR (dB)	σ^2	σ_n^2	$\lambda = \alpha\,\sigma_n^2$	σ_n^2	$\lambda = \alpha\,\sigma_n^2$
100	9.70×10^{-12}	4.04×10^{-7}	1.55×10^{-5}	4.04×10^{-7}	1.55×10^{-5}
20	9.70×10^{-4}	1.19×10^{-3}	3.85×10^{-2}	8.50×10^{-4}	2.58×10^{-2}
15	3.07×10^{-3}	1.84×10^{-3}	5.27×10^{-2}	2.90×10^{-3}	8.70×10^{-2}
10	9.70×10^{-3}	1.12×10^{-2}	2.84×10^{-1}	1.00×10^{-2}	2.53×10^{-1}
5	3.07×10^{-2}	3.05×10^{-2}	6.08×10^{-1}	2.39×10^{-2}	5.41×10^{-1}
0	9.70×10^{-2}	1.01×10^{-1}	1.64	9.23×10^{-2}	1.60

This indicates that we can not be sure whether predicted output values in these regions are correct. In general, the resulting network output function may as well display large peaks outside the region of training data, like the RBF network without regularization in fig. 3.4 (c) and (d). Pruning of centers distant from all input training data (as depicted in sect. 3.1.1 on page 25) should thus also be performed before Bayesian network training.

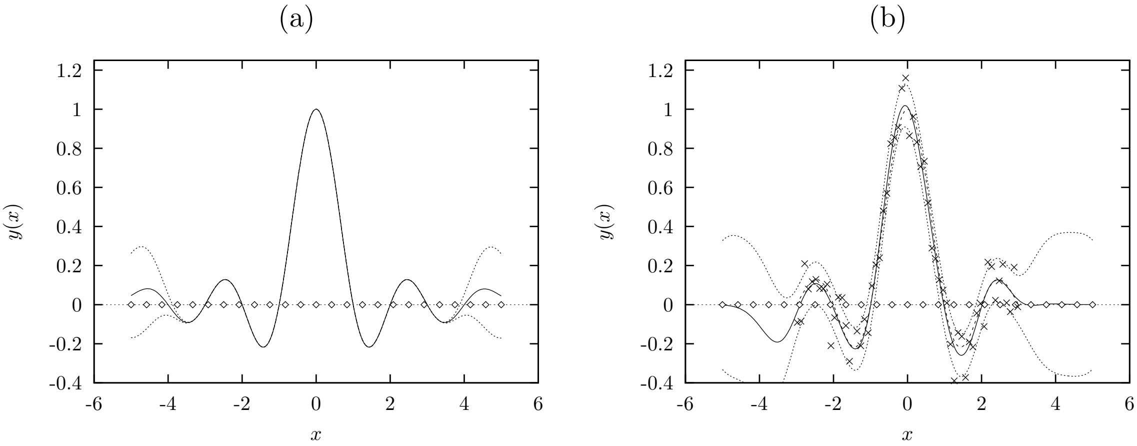


Figure 3.8: Modeling of the sinc function with incomplete training data ($x \in [-3, 3]$) by an RBF network with Bayesian learning. (a) noiseless case, (b) noisy case with an SNR of 10 dB. Note the larger spread of the error bars $\hat{f}_{\text{bay}}(x) \pm \sigma_y(x)$ (dotted lines) in the regions not covered by training data.

3.1.4 The relevance vector machine (RVM)

The relevance vector machine [Tip01] is a training algorithm also based on Bayesian estimation of the RBF network weights. Again the additive noise model (eq. 3.17) is assumed as the source of the training data. Deviating from the algorithm described in the previous section, an individual Gaussian prior pdf is chosen for *each network weight:*

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N_c} p(w_i|\alpha_i) \ , \quad p(w_i|\alpha_i) = \mathcal{N}(0, \alpha_i^{-1}) \ , \tag{3.27}$$

with hyper-parameters α_i, $i = 1, 2, \ldots, N_c$, determining the variance of $p(w_i|\alpha_i)$. A small value for α_i relates to a large variance of the prior pdf for w_i, which again should be the initial setting, as we do not know anything about the weights a priori. A large value for α_i means that the prior pdf $p(w_i|\alpha_i)$ as well as the posterior pdf $p(w_i|\boldsymbol{X}, \boldsymbol{t}, \alpha_i, \sigma_2^2)$ for this weight is concentrated around zero. In this case, the weight w_i almost certainly is close to zero.

Similar to the learning process depicted in the last section, the estimation of the network weights is accomplished in an iterative procedure [Tip01], by first computing the parameters of the weights' posterior distribution

$$\begin{aligned} \boldsymbol{\Sigma}^{(i)} &= \left(\frac{1}{\sigma_n^{2\,(i)}} \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi} + \boldsymbol{A}^{(i)} \right)^{-1} \ , \\ \boldsymbol{\mu}^{(i)} &= \frac{1}{\sigma_n^{2\,(i)}} \boldsymbol{\Sigma}^{(i)} \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{t} \ , \end{aligned} \tag{3.28}$$

with $\boldsymbol{A}^{(i)} = \mathrm{diag}([\alpha_1^{(i)}, \alpha_2^{(i)}, \ldots \alpha_N^{(i)}])$. Again, small values for the initial values of the hyper-parameters $\alpha_k^{(1)}$ are chosen (i. e., a flat prior distribution for the weights is assumed), and after evaluating eq. 3.28 the hyper-parameters are updated according to

$$\begin{aligned} \alpha_k^{(i+1)} &= \frac{\gamma_k^{(i)}}{(\mu_k^{(i)})^2} \ , \qquad k = 1 \ldots N_c^{(i)} \ , \\ \frac{1}{\sigma_n^{2\,(i+1)}} &= \frac{\left\| \boldsymbol{t} - \boldsymbol{\Phi} \boldsymbol{\mu}^{(i)} \right\|^2}{N_c - \sum_k \gamma_k^{(i)}} \ , \qquad \text{with} \\ \gamma_k^{(i)} &= 1 - \alpha_k^{(i)} \Sigma_{kk}^{(i)} \ , \qquad k = 1 \ldots N_c^{(i)} \ . \end{aligned} \tag{3.29}$$

Again, $i = 1, 2, \ldots, N_i$ is the iteration index, μ_k is the k^{th} element of the mean vector $\boldsymbol{\mu}$, Σ_{kk} is the k^{th} element on the diagonal of the covariance matrix $\boldsymbol{\Sigma}$, and γ_k are additional intermediate parameters equivalent to γ in eq. 3.23.

While iterating eqs. 3.28 and 3.29 some α_i tend to ∞, meaning the pdf of the corresponding weight is concentrated around zero and so the corresponding basis function can be pruned without impairing network performance. Iterations are continued with the corresponding values μ_i removed from $\boldsymbol{\mu}$, the according rows and columns deleted from $\boldsymbol{\Sigma}$ and with $N_c^{(i)}$ reduced by the number of pruned basis functions. In our implementation of the RVM training process iterations are stopped if the number of centers does not change during 10 iterations. After the iteration the weights for the remaining centers are set to the mean values of their posterior pdf:

$$\hat{\boldsymbol{w}}_{\mathrm{RVM}} = \boldsymbol{\mu}^{(N_i)} \ . \tag{3.30}$$

The nonlinear function model provided by the RVM is

$$\hat{f}_{\mathrm{RVM}}(\boldsymbol{x}) = \hat{\boldsymbol{w}}_{\mathrm{RVM}}^\mathsf{T} \boldsymbol{\varphi}'(\boldsymbol{x}) \ , \tag{3.31}$$

with $\hat{\boldsymbol{w}}_{\mathrm{RVM}}$ the $N_c^{(N_i)}$-dimensional vector of weights remaining in the network after N_i iterations, and $\boldsymbol{\varphi}'(\boldsymbol{x})$ the vector of the remaining basis function's responses to the input vector.

The RVM thus represents the nonlinear function by a *sparse RBF network*, comparable to support vector machines [Vap95]. Only centers that are relevant to the modeling of a function are used for regression[7]. Error bars for the predicted output values can be computed by the same formula as for the Bayesian training (eq. 3.26 on page 32), with the vector $\boldsymbol{\varphi}(\boldsymbol{x})$ and the matrix $\boldsymbol{\Sigma}$ now only containing values corresponding to the centers not pruned during RVM training iterations.

In fig. 3.9 on the facing page, the modeling of the sinc function by the RVM is depicted (cf. also [Tip01]). In the first four subplots (a)-(d) the same network and training parameters as for the previous modeling tests (see caption of fig. 3.4 on page 26) were used. Here we observe that in the noiseless cases (a) and (c) nearly all of the network centers are used for the RVM regression. In the noisy case (b) and (d), somewhat more of the centers are pruned. For incomplete training data (case (c) and (d)) also the outlying centers are used for the regression and the problem of large weight values and peaks in the network output function (cf. fig. 3.4 on page 26) persists.

For the choice of a larger basis function width, fig. 3.9 (e)-(h), a larger number of basis functions are pruned, due to the higher interaction between basis functions for different centers. Outlying centers, however, are still included in the final RVM. This calls for *a priori* pruning of centers distant from the training input data for the RVM, or for the use of network centers equal to (a subset of) the input training data as described in [Tip01].

The regression behavior of the RVM on the sinc function generally resembles the behavior of the RBF network with Bayesian learning of regularization factor and noise variance, with the exception that for a large enough basis function width the smooth target function is represented by a smaller number of basis functions.

3.2 Other models

We will give a very brief overview of other realizations of the nonlinear function that have also been used for the oscillator model or closely related models.

3.2.1 Locally constant (nearest neighbor) approximation

A straightforward approach to modeling the nonlinear function is the construction of a locally constant approximation based on the training data [KK94]. The function $f(\boldsymbol{x})$ is approximated by a piecewise constant function with the function value corresponding to the output target of the nearest training sample in input space:

$$y = \hat{f}_{\text{loc}}(\boldsymbol{x}) = t_i \quad , \quad i = \arg\min_{k=1\ldots P}(\|\boldsymbol{x} - \boldsymbol{x}_k\|) \quad , \quad \mathbb{R}^N \to \mathbb{R} \quad , \tag{3.32}$$

with $\boldsymbol{x}$ the N-dimensional input vector, $\boldsymbol{x}_k$, $k = 1 \ldots P$ the input vectors of the training data, and t_k the corresponding outputs (targets), the vector norm is the Euclidean $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^{N} x_i^2}$. The $\arg\min$ function picks the index k that minimizes its argument.

Thus, the dynamics in phase space are represented by a number of vectors each corresponding to a certain region of phase space and the respective outputs. Predictions are made by finding the appropriate prototype vector for the region containing the current input vector and returning the corresponding output value.

In fig. 3.10 on page 38, the manner a locally constant model approximates the sinc function is exemplified. The output function is piecewise constant, and if enough training samples are available a reasonably good approximation of a given function can be achieved (fig. 3.10 (a)). For noisy training data (fig. 3.10 (b)) the noise is preserved by the locally constant model, and

[7] RVM learning can also be applied for minimizing the number of filter taps in adaptive filters [KKP03].

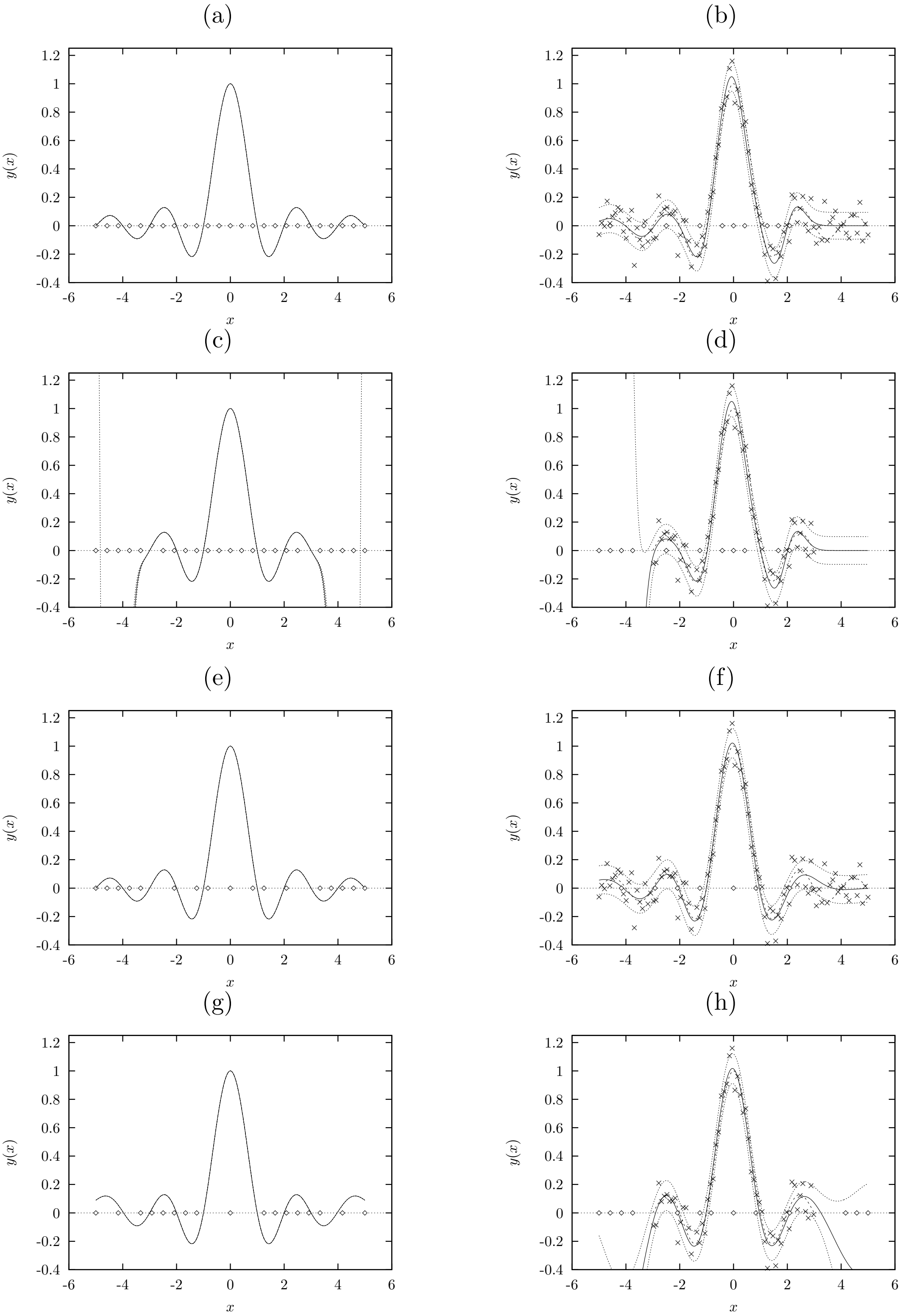


Figure 3.9: RVM modeling of sinc function. The centers finally used by the RVM are indicated by the diamonds on the x axis. Network parameters for case (a)-(d) are the same as given for fig. 3.4 on page 26, for case (e)-(h) the width of basis functions is doubled. The network output function (solid line) is accompanied by the 'error bars' calculated in eq. 3.26 (dotted lines). The number of centers is reduced more if the width of the basis function is chosen larger. In the case of training data not covering the range of centers (cases (c), (d), (g), and (h)), some outlying centers are not eliminated in the training procedure and large weights are assigned occasionally, resulting in peaks in the network output function.

within the range of training input data the function model will reproduce a similar amount of noise in the output as is present in the training data. This may constitute a drawback in terms of generalization. However, it is advantageous if we want the oscillator model to reproduce both voiced and unvoiced excitation. For incomplete training data (fig. 3.10 (c) and (d)) the locally constant model implicitly does a constant extrapolation to regions not covered by the training data.

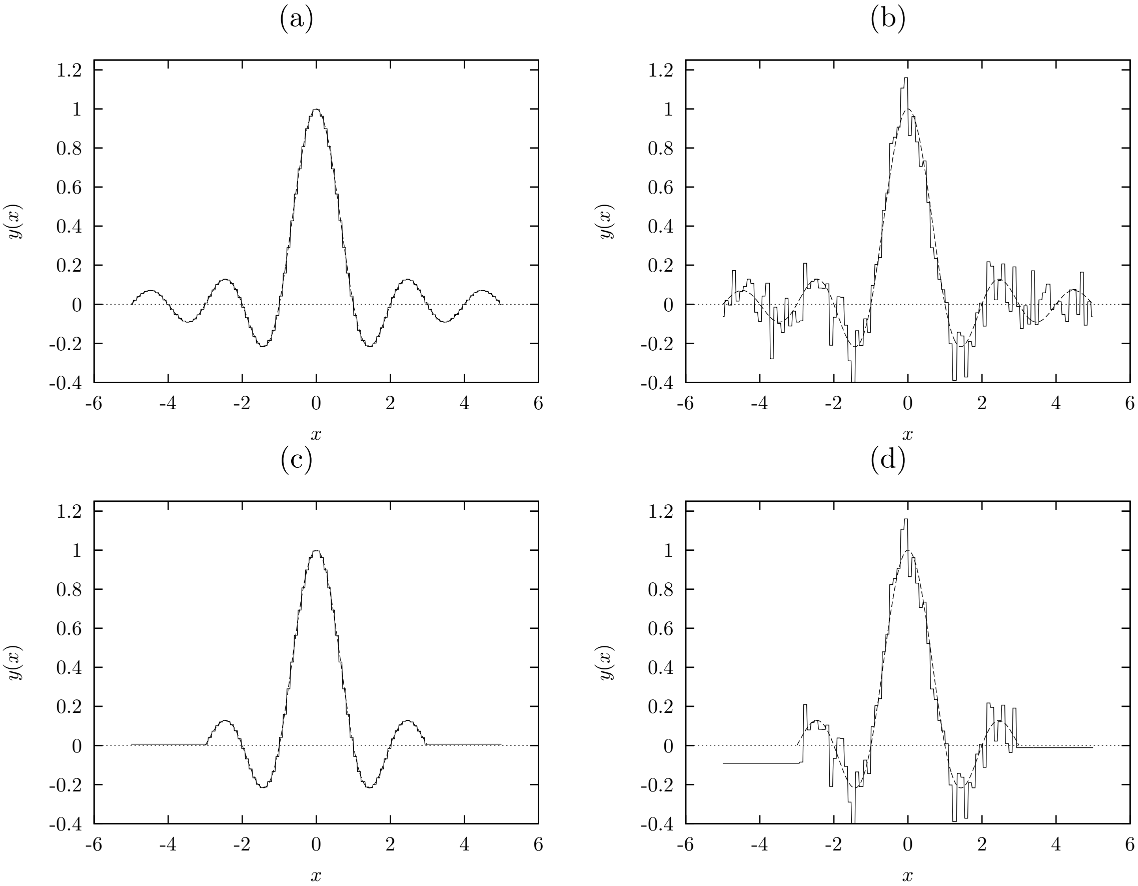


Figure 3.10: Example of function approximation by the locally constant model. Modeling of the sinc function a) using 100 training samples in the range $x \in [-5, 5]$ without noise, b) with an SNR of 10dB, and c) using 60 training data samples in $x \in [-3, 3]$ without noise and (d) with an SNR of 10dB.

Using a recorded speech signal for model learning, the approximation of $f(\cdot)$ is constructed from the state transitions in phase space observed for the speech signal. These transitions are, e. g., stored in a codebook and for synthesis the nearest neighbor of the current state in the codebook is accessed for prediction. The locally constant model with a frame-wise adaptive codebook has been used for time scale modification of speech signals [KK94].

3.2.2 Multi-layer perceptrons

Multi-layer perceptrons (MLPs) are ANNs more closely related to biological functions in the brain than RBFs. At the nodes of the network a nonlinear function related to the threshold for firing of a neuron (activation function) is applied to a weighted sum of the previous nodes' outputs. A fully connected MLP with scalar output is defined by the equation

$$y = \hat{f}_{\mathrm{MLP}}(\boldsymbol{x}) = g_l\left(\sum_{i=1}^{N_l} w_i^l\ g_{l-1}\left(\sum_{j=1}^{N_{l-1}} W_{j,i}^{l-1}\ \ldots\ g_1\left(\sum_{k=1}^{N_1} W_{k,j}^1\, x_k\right)\right)\right) \ . \tag{3.33}$$

Here, l is the number of network layers, N_m the number of units in layer m, the matrices $\boldsymbol{W}^m = [W_{j,i}^m]$ and the vector $\boldsymbol{w}^l = [w_i^l]$ are the weights at the connections from unit i in layer $m-1$ to unit j in layer m, and $g_m(a)$ is the activation function at the units of layer m.

An activation function often used is

$$g(x) = \frac{2}{1+\exp(-2cx)} - 1 \ , \qquad c > 0 \ , \tag{3.34}$$

which resembles a threshold function for $c \rightarrow \infty$, but is continuous and differentiable, which is essential for MLP training.

The network parameters are the weights between network layers $\boldsymbol{W}^m$, $m = 1, \ldots l-1$ and $\boldsymbol{w}^l$, the parameter(s) of the activation functions, and commonly also a bias vector for each layer to shift the individual activation functions along the input axis.

MLPs with as few as two layers and smooth activation functions are capable of approximating an arbitrary function (MLPs are said to be universal function approximators) [HSW89, HSW90]. There is, however, no constructive rule to determine the structure and size of an MLP to achieve a certain amount of approximation accuracy.

Training of the MLP parameters has to be performed in an iterative procedure, the most commonly used training algorithm is back-propagation [Hay94]. Here, the search for the optimal network parameters is performed by gradient descent on the error function surface, with random initialization of the parameters. Thus, the solution finally found may only represent a *local minimum* of the error function instead of the desirable *global minimum*. Commonly, the training process is performed for several parameter initializations and the network with minimum error will be used.

In fig. 3.11 on the following page again the regression for the sinc function is depicted. For these examples, the training process had to be performed several times until an MLP with a reasonably low modeling error was found. One can see that both in the noiseless cases (a) and (c) and for additive noise (b) and (d) the network function reasonably well fits the original function. With incomplete training data (c) and (d), though, the network output function tends to large (negative) values for $|x| > 3$. This does not happen for all training runs, though, so one can manually choose a network with a suitable network output function for incomplete training data. Moreover, the network output is always limited by the maximum of the activation function in the output layer. However, the output of an MLP may basically display arbitrary output (in the range allowed by the output activation function) for an input in regions not covered by the input training data due to the lack of an error criterion for these regions.

The transition and interpolation between different models (e. g., for different speech sounds) using MLPs is hindered by the missing relation between MLP parameters for different models. Even for the same training data, an MLP model may internally be represented by completely different parameterizations. Due to the usually identical activation functions in one layer, and due to the full interconnection matrix between the layers, the N_m units in layer m can be permuted *ad libitum*, so there is no clear ordering of one layer's units. Moreover, different initializations of the network parameters may result in equal solutions in terms of prediction error (at local minima of the error function) with completely different parameter values. Thus, non-stationary synthesis (e. g., transitions between different phonemes) may only be performed by switching between different nonlinear function models.

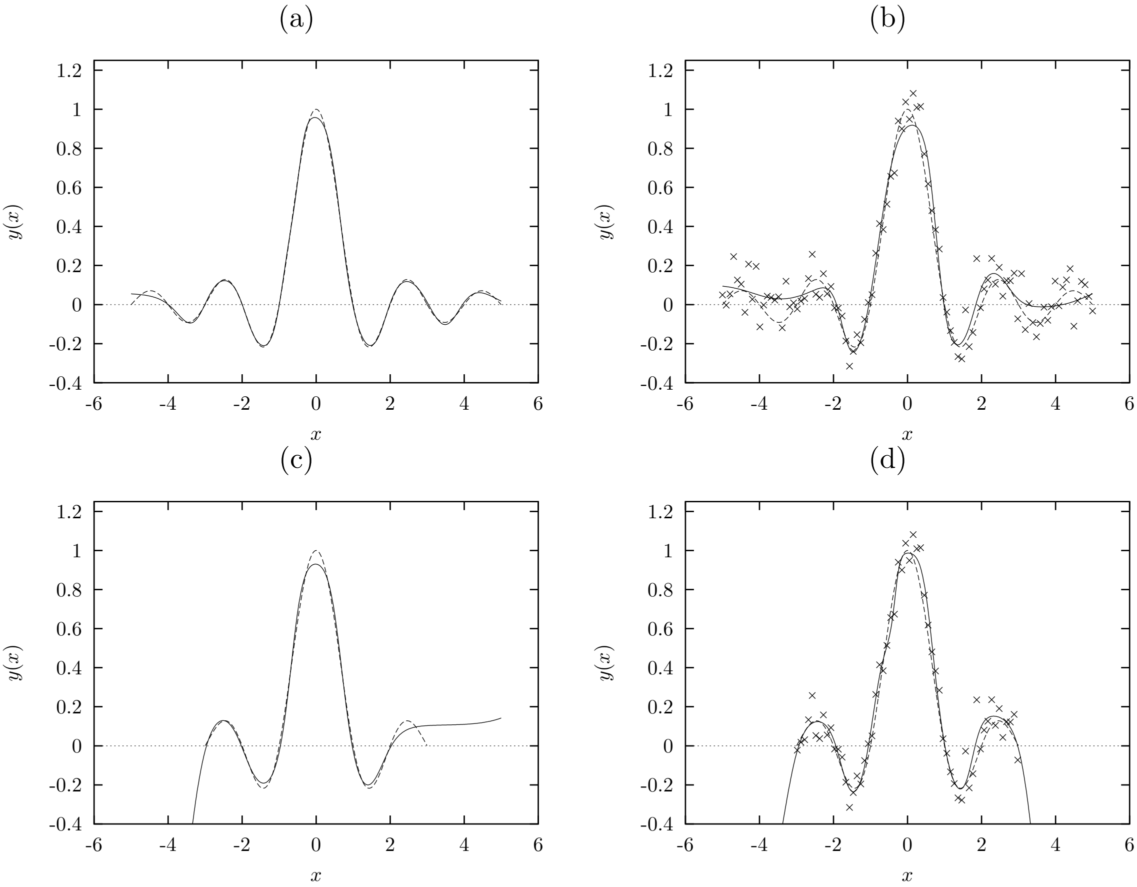


Figure 3.11: Example of function approximation of the sinc function by a multi-layer perceptron, (a) using 100 training samples in the range $x \in [-5, 5]$ without noise, (b) with an SNR of 10 dB, (c) using 60 training data samples in $x \in [-3, 3]$ without noise, and (d) with an SNR of 10 dB. Again, dashed lines show the target function, solid lines the network output function, and crosses the training examples in the noisy cases. The MLP architecture is 1-9-1 (input, hidden, output units) resulting in 28 network parameters, and $g_i(\cdot) = \tanh(\cdot)$. Network training has been performed using the Levenberg-Marquardt back-propagation algorithm limited to 30 epochs. The network was trained with several different initializations and examples with "good" approximation properties are depicted here.

3.2.3 Multivariate adaptive regression splines (MARS)

Another approximation for a nonlinear function are multivariate adaptive regression splines [Fri91], which have been used for nonlinear sub-band synthesis of speech [HK98]. Here, the input space is repeatedly partitioned and each sub-region is approximated by a product of linear spline function. The partitioning threshold is adapted to minimize the squared error in each stage. During the repeated partitioning the spline functions from earlier stages are used as basis for the next approximation step (up to a specified number of maximum interaction).

Function approximation by MARS results in a continuous output function, consisting of products of linear spline functions in the components of the input vector for each of the final partitions of input space. The one-dimensional example for the sinc function is depicted in fig. 3.12. If a high enough number of splines is used the target function is approximated well (fig. 3.12 (a)). In the presence of noise the MARS model still captures the general picture (fig. 3.12 (b)), suggesting good noise robustness. For the case of incomplete training data (fig. 3.12 (c),(d)) the modeling by linear splines results in a linear extrapolation for regions not covered by the input training data. Thus, the absolute value of the model output may attain large values for an input outside the training data, especially if the maximum absolute output value in the training set is reached at the border of the region of input training data.

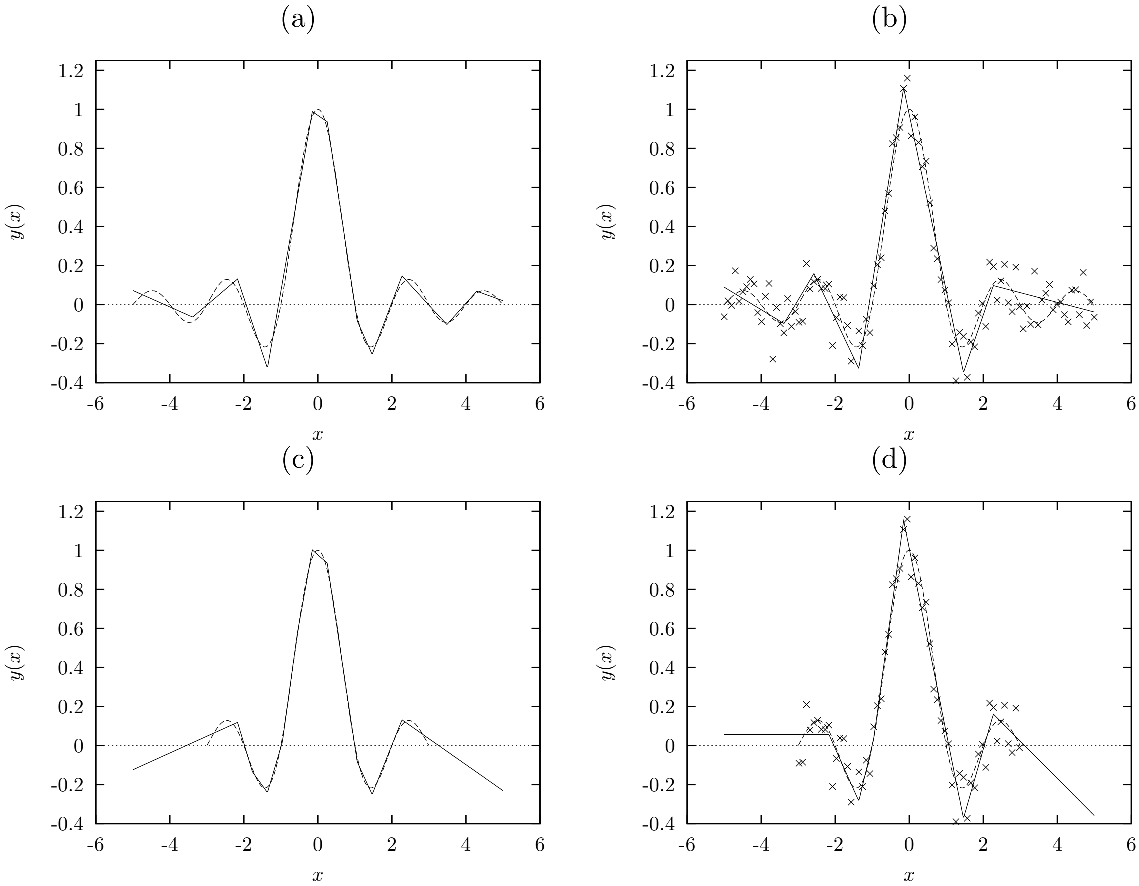


Figure 3.12: Regression of the sinc function by a MARS model. The training was limited to a maximum interaction level of 3 and a maximum number of basis functions equal to 20. As in the previous examples, case (a) is for 100 training samples in $x \in [-5, 5]$ without noise, case (b) with additive Gaussian noise with an SNR of 10 dB, and case (c) for 60 training samples limited to $x \in [-3, 3]$ without noise and (d) with an SNR of 10 dB.

Due to the adaptive partitioning depending on the training data, the partitioning thresholds and spline coefficients will be different for different target functions and training data. A direct relation between MARS models for different sound signals can thus not be easily established.

3.3 Conclusion

In this section a number of candidate models for the nonlinear function in the oscillator model were presented and tested in a simple one-dimensional regression task with noisy and incomplete training data. The aim was to test their usability for the oscillator model, and to find the reasons for some of the problems that are reported in the literature and became apparent in our work.

The most commonly used indicator for a "good" nonlinear function model is the modeling error. A summary of the modeling errors of the candidate models with approximately the same number of parameters for the sinc function is given in table 3.2. Here we can see that the best performance, i. e., the lowest prediction error is achieved by models using continuously differentiable functions, namely the RBF and MLP networks, on the noiseless training data. The worse performance of the locally constant model and the MARS model on noiseless training data is due to the function approximation by a finite number of constant or linear splines, respectively. In the case of noisy training data, the locally constant model displays a modeling error in the range of the noise level in the training data, i. e., the *locally constant model preserves the noise* of the training data (see also fig. 3.10 on page 38), whereas – for the reasonably large number of training data in the examples – *all other models suppress the noise* to some extent.

Table 3.2: Modeling error for regression of the sinc function of candidate nonlinear function models for different amount of Gaussian noise in the training data. The numbers are associated with the conditions reported in the plots in the last sections, for complete training data (100 samples for $x \in [-5, 5]$). The modeling error is computed for the noiseless target sinc function for 500 points in $x \in [-5, 5]$ and normalized relative to the signal level.

	SNR of training data		
	∞	20 dB	10 dB
Model	Relative modeling error (dB)		
RBF network without regularization	−55.29	−25.85	−15.83
RBF, reg. matrix inversion ($\lambda = 1$)	−22.11	−21.72	−17.43
GRBF ($\lambda = 1$)	−18.54	−18.67	−16.35
RBF, cross-validation	−55.34	−26.85	−15.62
	($\lambda = 10^{-3}$)	($\lambda = 0.1$)	($\lambda = 1$)
Bayesian RBF	−55.29	−26.29	−16.99
	($\lambda = 1.6 \times 10^{-5}$)	($\lambda = 1.8 \times 10^{-2}$)	($\lambda = 0.14$)
RVM	−55.18	−26.10	−16.46
Locally constant (nearest neighbor)	−25.76	−19.94	−10.93
MLP, backprop.	−27.38	−19.91	−15.82
MARS	−20.65	−19.84	−14.48

The basic RBF network, the regularized RBF network with cross-validation, Bayesian learning, as well as the RVM perform similarly in terms of modeling error, with a very good function approximation in the absence of noise and a modeling error about 6 dB below the noise level in presence of noise in the training data. The regularized RBF network with a fixed regularization factor $\lambda = 1$ exhibits an almost constant modeling error, regardless of the noise level, both for regularization of matrix inversion and GRBFs. This is primarily due to the preference of lower weight values induced by regularization, which leads to a general underestimation of function peaks (cf. fig. 3.6 on page 29). When using cross-validation or the Bayesian algorithm to determine the regularization factor less regularization is applied in the lower noise cases, leading to a better function approximation.

However, the ratings of the different function models in terms of modeling error may not be sufficient to prove the suitability for use in the oscillator model [HP98b, CMU02a]. For other preconditions it is not easy to give quantitative measures. We will, however, try to give some qualitative criteria here.

As we have previously noted the training data for speech signals in a multi-dimensional phase-space embedding may only fill a small region of phase space. So, we have to deal with *incomplete training data* during nonlinear function training. To explore the suitability of the candidate models for such a task we have restricted the training process for modeling the sinc function to a limited region of the input variable $x \in [-3, 3]$ and observed the model output function for $x \in [-5, 5]$.

Generally, all model output functions for regression in $x \in [-3, 3]$ resemble the results stated above. However, for inputs outside the region of training data, different behavior is observed. The locally constant model results in a constant extrapolation of the output value corresponding to the nearest training input value (fig. 3.10 on page 38 (c)). This can be considered favorable for the use in the oscillator model since the nonlinear function output cannot grow beyond the range of training output data and an arbitrary state in phase space should quickly converge to the region of the actual signal trajectory during synthesis.

For the basic RBF network with fixed centers on a hyper-grid it was observed (fig. 3.4 (c)) that the network output function may display large peaks in the region not covered by the training data, due to the assignment of large weights to the corresponding centers. This problem could be relieved either by restricting the center positions to the region in input space covered by the training data (sect. 3.1.1) or by regularization (sect. 3.1.2). In both cases the network output function decays to zero output outside the training data region. When applied in the oscillator model this means that, for each state outside the training data, the system trajectory would be pulled to the origin. This seems problematic if the origin of state space itself is not covered by the training data. However, if basis functions with infinite support (like Gaussians) are used, the system trajectory will also approach the original trajectory, even if the origin is not near to the training data (cf. [Man99]). By preventing high peaks in the network output function we shall also prevent an oscillator behavior with large amplitude intermediate pulses as observed in [Man99, fig. 6.11] for the RBF network without regularization (cf. sect. 4.3.2).

The Bayesian RBF network and the relevance vector machine may, similarly to the RBF network without regularization, assign large weight values to centers away from the input training data, especially in the case of noiseless training data when regularization is automatically reduced (e. g., fig. 3.9 on page 37, subplot (c)). Thus, the center positions for the Bayesian network training should be restrained to the input space region covered by the training data, either by choosing the centers equal to (a subset of) the training data as in [Tip01] or by pruning centers away from the training data before network training.

For the MLP the network output function may attain arbitrary values in the range of the activation function in the output layer for regions not covered by the training data. The MARS model may assign splines with nonzero slope at the borders of regions covered by training data. Thus, both have to be considered dangerous to the stability of the oscillator model.

Using a relatively small number of training data examples as compared to the number of nonlinear function parameters – like in the examples 100 or 60 training samples compared to about 30 parameters[8] – we have to consider the problem of over-fitting noisy training data and of bad generalization. As we have seen this might be a concern for the basic RBF network. However, both *a priori* pruning, and thereby reducing the number of network parameters, and regularization mitigates this problem for smooth target functions. Also the MLP and the MARS model can be considered to give good interpolation of a smooth target function in the noisy case.

[8] With the exception of the locally constant model, which uses all training samples as 'parameters'.

Concerning the aim of associating a sequence of nonlinear function models for different sounds (different training data), only the RBF model with *a priori* fixed centers and basis function widths can provide such relations. Here a clear correspondence between the weights for different training data is given according to the position of the RBF centers in input space. This correspondence can also be retained if different centers are pruned for different training data in the RVM, by assigning zero weight values to the pruned centers.

To conclude, for the realization of the nonlinear function in the oscillator model, the RBF network with *a priori* fixed center positions and basis functions widths seems to be the favorable choice. We will apply pruning of network centers that are more than a specific maximum distance away from all input training data to ensure a good numerical conditioning for matrix inversion during training, and to prevent peaks in the network output function. Furthermore, we will use regularization to further promote good generalization and robustness of the network output function for noisy training data. We will, in particular, make use of automatic determination of the regularization factor, by cross-validation or Bayesian learning of the noise variance and regularization factor.

Chapter 4

Speech synthesis with the oscillator model

In this chapter we examine modeling and re-synthesis of speech signals by a nonlinear oscillator. The oscillator model used in our work is based on a nonlinear predictor for time series. In sect. 4.1 we present considerations for optimal time series prediction. The oscillator model is introduced in sect. 4.2. In sect. 4.3 we examine the application of the oscillator model using the various nonlinear function models presented in Chapter 3 to stationary full speech signals. In sect. 4.4 we argue for combining linear prediction and the oscillator model to achieve a higher number of successfully re-synthesized signals. Finally, in sect. 4.5 we present some examples how to generate speech signals with varying fundamental frequency and transitions between different speech sounds by model interpolation.

4.1 Time series prediction

Prediction of scalar time series $\{x(n)\}$ refers to the task of finding an estimate $\hat{x}(n+1)$ of the next sample $x(n+1)$ based on the knowledge of the history of the time series, i. e., the samples $x(n), x(n-1), \ldots$. A common approach is *linear prediction* [MG76], where the estimate is based on a linear combination of N_{LP} past samples

$$\hat{x}_{\mathrm{LP}}(n+1) = \sum_{i=0}^{N_{\mathrm{LP}}-1} a_i \, x(n-i) \ , \tag{4.1}$$

with the prediction coefficients $a_i, i = 0, \ldots, N_{\mathrm{LP}}-1$. The linear prediction approach, however, does not account for nonlinear dependencies in the time series. Introducing a general nonlinear function $f(\cdot) : \mathbb{R}^N \to \mathbb{R}$ applied to the vector

$$\boldsymbol{x}(n) = [x(n), x(n-M), \ldots, x(n-(N-1)M)]^{\mathsf{T}} \tag{4.2}$$

of past samples, we arrive at the *nonlinear prediction* approach

$$\hat{x}_{\mathrm{NP}}(n+1) = f(\boldsymbol{x}(n)) \ . \tag{4.3}$$

The vector series $\{\boldsymbol{x}(n)\}$ is called an *time-delay embedding* of the system underlying the time series $\{x(n)\}$ in N-dimensional space. Here we already include the option of using a vector $\boldsymbol{x}(n)$ of non-contiguous past samples (instead of the latest N samples $[x(n), x(n-1), \ldots, x(n-(N-1))]^{\mathsf{T}}$) from the original time series in the embedding by introducing an *embedding delay* M that can be chosen $M \geq 1$. The choice of $M > 1$ has the advantage of increased memory length (number of past samples in the delay line) for a fixed embedding dimension (which relates to complexity of the nonlinear function model, see below). A schematic of this nonlinear predictor is given in fig. 4.1.

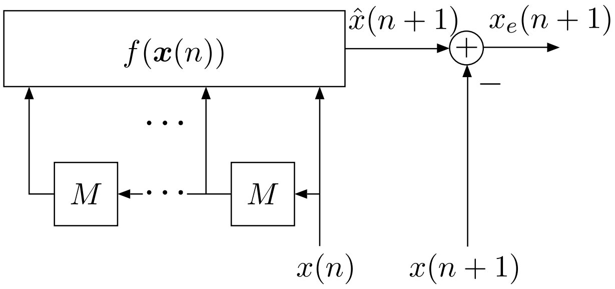


Figure 4.1: Nonlinear predictor for scalar time series

4.1.1 Embedding dimension

The choice of embedding dimension N, i. e., the dimension of the embedding vector $\boldsymbol{x}(n)$, is based on either the dimension of the original system that gave rise to the signal (which is often unknown) or on an estimate of the – possibly fractal – dimension of the attractor on which the signal evolves. The dimension of the embedding determines the dimension of the input space of the prediction function $f(\cdot)$ in our oscillator, and thus is most often related to the complexity of the nonlinear function model for $f(\cdot)$. Furthermore, the higher the dimension of the input space for a parametric nonlinear function model is, the more sparsely this input space will be filled with training data: This is known as the so-called "curse of dimensionality". Hence, the lowest possible embedding dimension should be searched for.

An embedding theorem going back to [Whi36] says, that a d-dimensional smooth manifold has a diffeomorphism in $\mathbb{R}^{2d+1}$ generated by a 'generic smooth map'. Thus, almost any d-dimensional manifold can be embedded in another space of dimension $N = 2d + 1$ by an invertible (i. e., one-to-one) mapping.

For dynamical systems with only a scalar observable (like a speech signal), Takens [Tak81] proved an analogous theorem for the system attractor generated by a time-delay embedding (eq. 4.2). According to Takens' theorem the attractor dynamics of the original nonlinear system that gave rise to a scalar time series can be fully reconstructed from a time-delay embedding (eq. 4.2) if the embedding dimension is

$$N \geq 2\,d + 1 \quad , \tag{4.4}$$

depending on the dimension d of a manifold that can hold the attractor of the observed signal, i. e., d being an integer larger than the *fractal dimension* d_f of the observed signal attractor. This embedding theorem is valid even for a chaotic system. Thus, we can always embed a signal with a fractal dimension of, e. g., $d_f = 2.1$ in a time-delay phase space of dimension $N \geq 2 \cdot \lceil 2.1 \rceil + 1 = 7$, regardless of the actual dimension of the system state vector that gave rise to the signal (which may be $d_{\mathrm{orig}} = 100$). Takens' theorem is valid for arbitrary embedding delay M, noise-free observations of the steady state attractor, and if the time series is observed over an infinite time interval.

A little more relaxed is the embedding theorem by Sauer et al. [SYC91]. The steady-state attractor of an unknown dynamical system can be fully reconstructed in an N-dimensional embedding if

$$N > 2\,d_f \quad , \tag{4.5}$$

with d_f being the fractal dimension of the attractor. According to this theorem, an embedding dimension of $N = \lceil 2 \cdot 2.1 \rceil = 5$ suffices for a signal with fractal dimension $d_f = 2.1$.

The fractal dimension can be estimated indirectly from an observed scalar time series, e. g., using the box-counting dimension or correlation dimension for the attractor observed in an embedding phase space [ABST93]. However, some definitions for the fractal dimension, like the Hausdorff dimension may *not* be used with this embedding theorem [SYC91].

For voiced speech signals a correlation dimension of between one and two has been found in [BK91], values below three in [Tow91, BM94], but also higher values up to four have been stated in the literature. This means an embedding dimension N between three and nine will be necessary for the reconstructed phase space to represent the dynamics of a voiced speech signal. However, in [BK91] a saturation of the redundancy of the state space embedding with optimized embedding delay (see sect. 4.1.2 on the next page) for sustained vowel signals at all dimensions $N \geq 3$ has been found, indicating that a reconstruction dimension of three suffices for these signals if the embedding delay is chosen appropriately. For the application of the oscillator model with a locally constant nonlinear function model [KK94], it has been found that stationary speech can be reproduced well by a low-dimensional embedding $N = 4$, but for transient signals a higher embedding dimension is necessary.

In general, the above embedding theorems ensure a reconstruction of the original system dynamics, and they are sufficient, but not necessary conditions [HP98b], meaning that a suitable choice of the embedding delay M may allow to reconstruct a signal in a phase space of a lower dimension N than the embedding theorems require. A way to find a sufficient embedding dimension, if the embedding delay M is chosen already, is to look at the number of false neighbors for the points $\boldsymbol{x}(n)$ on the trajectory in phase space [KBA92]. A false neighbor is a nearby point $\boldsymbol{x}(k)$, $\|\boldsymbol{x}(k) - \boldsymbol{x}(n)\| < \epsilon$ whose trajectory significantly diverges from the trajectory of $\boldsymbol{x}(n)$. Increasing the embedding dimension N the number of false neighbors for a low-dimensional signal gets lower and should reach a minimum at the dimension which is sufficient for the embedding. In fig. 4.2 the fraction of false neighbors for some sustained vowel signals from male and female speakers are plotted as a function of N. As we see the number of false neighbors is close to zero for all signals for $N \geq 4$, suggesting that an embedding dimension of $N = 4$ suffices for sustained vowel signals.

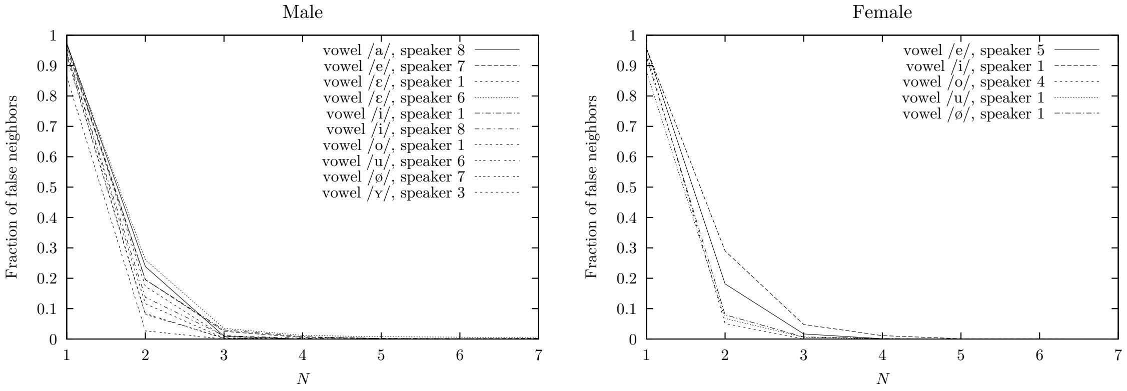


Figure 4.2: Fraction of false neighbors as a function of embedding dimension N for sustained vowel signals from male and female speakers. Embedding delay was chosen $M = 13$ for male, and $M = 11$ for female speakers (at a sampling rate of $f_s = 16$ kHz). The number of false neighbors is calculated by the function `false_nearest` from the nonlinear time series analysis package TISEAN [HKS99].

4.1.2 Embedding delay

The choice of embedding delay M should be guided by properties of the time series, too. The samples in $\boldsymbol{x}(n)$ should be chosen so as to serve as independent coordinates in reconstruction space. The choice of $M > 1$ prevents the usage of adjacent, and thus possibly highly correlated samples from the time series in the embedding. If, however, M is taken too large, the samples $x(n)$ and $x(n - M)$ will be disconnected, or statistically independent, especially for chaotic time series. Also, for time series with harmonic components, the embedding delay should be smaller than half the fundamental period [SYC91].

The choice of an appropriate intermediate value for M to yield uncorrelated samples in $\boldsymbol{x}(n)$ can be based on the auto-correlation function, e. g., by choosing M equal to the time lag of the first zero crossing of the auto-correlation, thus yielding linearly uncorrelated samples in $x(n)$ and $x(n - M)$. Minimizing linear dependence, however, might not be optimal, in particular for the embedding of signals coming from nonlinear systems (cf. [Ber97, App. B]). A better way is to look at the mutual information (MI) between the samples in two time series $\{x(n)\}$ and $\{x(n - L)\}$ as a function of lag L and to choose the embedding delay M at the lag L at which the first minimum of the mutual information occurs [Fra89, Ber98].

For sustained vowel signals the first minimum of the mutual information typically occurs for a delay between 0.5 and 1 ms (i. e., for a lag L between 8 and 16 samples at 16 kHz sampling rate). Mutual information as a function of delay is depicted for a number of sustained vowel signals spoken by male and female speakers taken from the speech signal database used to test our model (see sect. 4.3) in fig. 4.3. In several cases the 'first minimum' is not very pronounced. It is, however, clear that a choice of $M > 1$ is advantageous in any case.

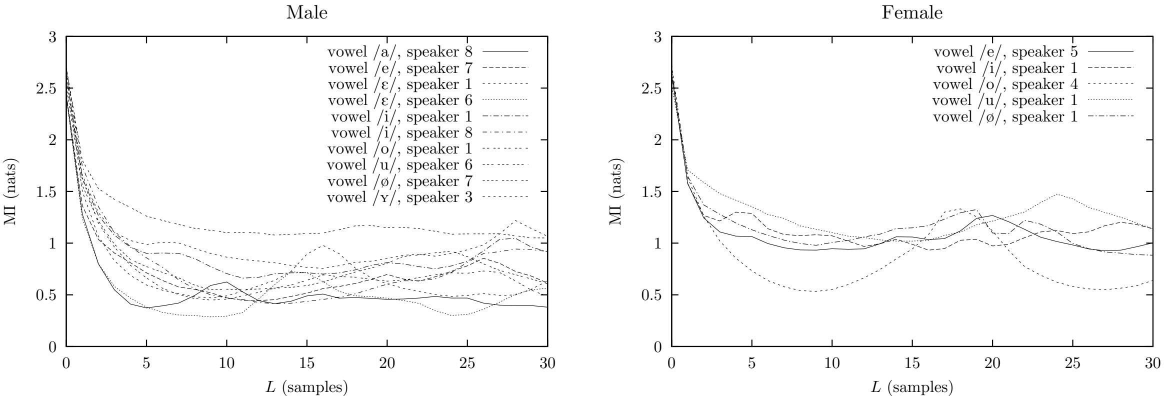


Figure 4.3: Mutual information (MI) as a function of time lag L for sustained vowel signals from male and female speakers. Signal sampling rate is 16 kHz. Mutual information is calculated using the function `mutual` in the nonlinear time series analysis package TISEAN [HKS99] with a number of 16 partitions.

In spite of the above rules for choosing embedding dimension and delay it has been found that the optimum values for the embedding parameters may only be found in a trial and error approach by varying N and M until an optimum performance in terms of an error criterion, like mean squared error for signal prediction is achieved. The optimum parameter values for prediction – i. e., parameter values that minimize the mean squared prediction error – may, however, not comprise the optimal setting for the use of a nonlinear predictor in the oscillator model [CMU02a]. Here the primary concern is to 'unfold' the trajectory in phase space to avoid intersections that cannot be discerned otherwise [SYC91]. Moreover, in order to keep a fixed structure of the predictor in the oscillator model for synthesis of transitions between different sounds we need to find a compromise embedding in common for all speech sounds.

4.1.3 Non-uniform embedding

Concerning the signal embedding the question arises, if the uniform embedding in a phase space constructed from equidistant samples from the scalar signal is optimal [JM98]. To test for a possible gain in nonlinear prediction based on a general choice of the single embedding delays we consider a 'non-uniform' embedding constructed of non-equidistantly spaced past samples from the signal as input of the predictor by building the vector $\boldsymbol{x}(n)$ according to

$$\boldsymbol{x}(n) = [x(n), x(n-M_1), x(n-M_2), \dots, x(n-M_{N-1})]^\mathsf{T} \quad , \quad M_1 < M_2 < \dots M_{N-1} \in \mathbb{N} \ . \tag{4.6}$$

We compute the prediction gain bound from the mutual information function [Fra89] by the method in [BK94, Ber97] for a one sample ahead prediction for all possible choices[1] of M_i in the range $M_i < 50$ with embedding dimension $N = 4$ for 10 sustained vowel signals from several male speakers, and for 5 sustained vowel signals from female speakers ($f_s = 16$ kHz).

The main finding is that the prediction gain for the optimum non-uniform embedding is in any case not more than 1 dB higher than for the optimum uniform embedding for each individual signal, the average increase in prediction gain is only 0.61 dB. Also, the maximum difference between the highest and the lowest achievable prediction gain for all embeddings with $M_i < 50$ is only 5.5 dB for male signals (mean difference is 3.8 dB, the overall mean prediction gain is 29.8 dB). For vowel signals from female speakers a qualitatively similar behavior is observed. Figure 4.4 shows the ranges of prediction gain bound found for the individual signals.

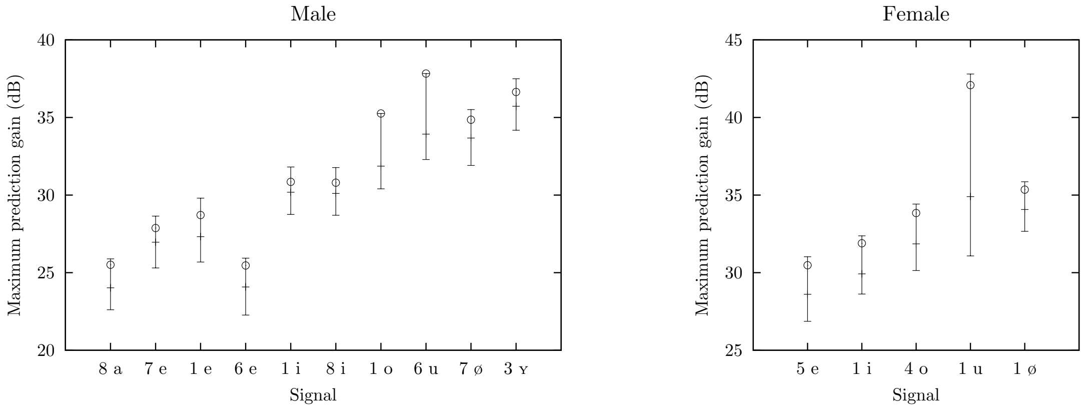


Figure 4.4: Span of prediction gain bounds for sustained vowel signals from male and female speakers for the prediction of the next future sample based on a non-uniform, four-dimensional embedding with all combinations of M_i in the range $M_i < 50$. The tic marks on the vertical lines indicate the highest, mean, and lowest prediction gain bound found for a four-dimensional embedding, the circles show the highest prediction gain for a uniform embedding. Signals are labelled with the number of the speaker and the vowel symbol.

In many cases the highest prediction gain is attained for low embedding delays. For non-equidistant embeddings, the delay $M_1 = 1$ is found for six of the ten male signals, in two cases we even find $M_1 = 1, M_2 = 2, M_3 = 3$. Likewise, the optimal uniform embeddings are found for $M = 1$ in three cases, for nine of the ten signals the optimal uniform embedding is found with an embedding delay $M \leq 8$.

This leads us to the conclusion that a non-uniform embedding optimized for the one sample ahead prediction gain does not greatly improve prediction performance. For the oscillator

[1] A number of 18 424 different embeddings

model this optimization regarding prediction gain may even be harmful: As mentioned above, for the oscillator model the embedding parameters should be chosen as to unfold the signal attractor in phase space, which is not the case if low embedding delays M_i are used, as it is the case for the non-uniform embeddings with the highest prediction gain for many speech signals.

It has to be noted that a low-dimensional embedding can also be derived from a higher-dimensional embedding ($M = 1$, N large) by linear combination of the high-dimensional embedding vector components [SYC91], e. g., by singular value decomposition [BMM99]. In [Man99] singular value decomposition embedding for full speech signals has been investigated and it was shown that it leads to attenuation of frequencies above 1 kHz in the re-synthesized signals. Besides, the singular value decomposition has to be performed for each signal, i. e., the decomposition matrix will be different for different training signals, in general[2].

4.2 The oscillator model

Feeding the predicted sample of the nonlinear predictor fig. 4.1 to the delay line – rather than a sample from the original signal – we obtain the *oscillator model* [Kub95, PWK98]. The oscillator model is an *autonomous* nonlinear deterministic dynamical system, i. e., it can generate output signals without an excitation signal. As opposed to linear prediction (eq. 4.1), which represents a finite impulse response (FIR) filter that can be inverted[3] to yield a purely recursive infinite impulse response (IIR) LP synthesis filter which has to be excited by, e. g., a pulse train or a noise signal (cf. formant synthesis, sect. 2.2), the oscillator model does not need any driving input signal.

The system equation of the discrete time oscillator model (cf. eq. 4.3) is

$$y(n+1) = f(\boldsymbol{y}(n)) \ , \tag{4.7}$$

with the oscillator output signal $y(n)$, and the vector $\boldsymbol{y}(n)$ representing a time-delay embedding of the *output* signal $y(n)$, as in eq. 4.2. A schematic of the model is depicted in fig. 4.5.

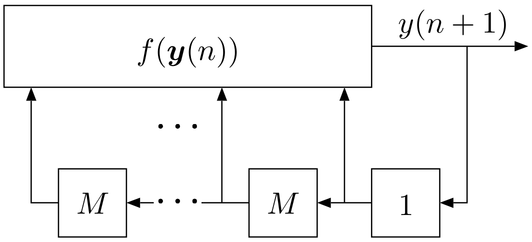


Figure 4.5: Autonomous oscillator model for signal synthesis

For synthesis of speech signals the embedding parameters N and M have to be chosen as described above, and a nonlinear function model, e. g., one of the models described in Chapter 3, is used as the realization of the nonlinear function $f(\cdot)$, with its parameters learned from a recorded speech signal. The oscillator model has been applied for generation and modification of speech signals, with several different realizations of the nonlinear function [KK94, Bir95, Kub96a, HK98, BMM99, NPC99, MM01, RK01, Ran01, Ran03, LZL03].

We have noted in the introduction that it is still a challenge to come up with a *stable* realization of the oscillator model for *successful re-synthesis* of speech signals. For the scope of

[2] One can of course try to find an optimal decomposition matrix for *all* signals of a speaker or a database.

[3] If the LP filter is minimum phase.

this thesis we use the following notion of stability and successful re-synthesis: A stable oscillator shall denote an oscillator that successfully re-generates an *output signal similar to the training signal* in the way that – for voiced speech signals used for training – stable oscillations are generated without severe deviations from the training signal in waveform shape, amplitude, and fundamental frequency. In particular, successful re-synthesis of voiced speech signals necessitates that the oscillator output signal does not converge to a fixed point, or diverge, nor displays intermittent large peaks. For vowel signals successful re-synthesis also shall imply that the vowel can be clearly identified perceptually.

4.3 Comparison of nonlinear function models

The oscillator model with different realizations of the nonlinear function model as described in Chapter 3 will now be tested for speech signals. Here we use a database of sustained vowel signals from 9 female and 11 male speakers recorded in a studio environment. The database consists of 9 vowels (and 3 nasals/1 liquid) uttered by each speaker in an artificially sustained manner. Speakers were asked to utter each vowel in the most stationary way, i. e., to keep fundamental frequency as well as spectral properties as constant as possible. The length of the recorded signals varies between 0.26 and 1.71 secs for male and between 0.22 and 1.71 secs for females speakers. The original sampling rate of 48 kHz was reduced to 16 kHz, i. e., to the sampling rate for 'wide-band' speech commonly used in high-quality speech synthesis, by linear phase low-pass filtering and down-sampling. The minimum length recordings thus provide a number of 4160 (male) and 3520 (female) signal samples for training.

4.3.1 Choice of embedding and nonlinear function model structure

Obviously, the large number of parameters – embedding parameters N and M, number of parameters for the nonlinear function (network centers, shape, position and width of the basis functions for RBF networks), number of training samples – would allow for individual optimization of our model for different speech sounds. However, due to the high dimensionality of the parameter space it seems improbable to find the 'best' parameter combination for every signal. Moreover, since we aim at using the oscillator model for general purpose synthesis we will try to *keep constant as many parameters as possible* for all different sounds. Thus, we will first determine a set of *fixed model parameters* that are used *for all vowel signals* in our database.

Regarding the choice of embedding dimension, a dimension of $N = 4$ seems to be justified by the method of false nearest neighbors (fig. 4.2) for sustained vowel signals, in general. Optimization of embedding delay according to mutual information, however, may yield different values for individual phonemes (cf. fig. 4.3). The value used here, $M = 13$, is a compromise for signals from male speakers. We found that both a lower value ($M = 10$) and a higher value ($M = 16$) result in a smaller number of successfully re-synthesized signals. For female speakers the optimum embedding delay for vowel signals might be somewhat smaller, however, for most of the following examples the same embedding delay $M = 13$ as for male speakers was used.

Regarding the RBF-network based models, we favor the utilization of fixed network center positions on a hyper-grid (cf. fig. 3.2 on page 22) for all signals, to achieve our aim of obtaining related parameters of models for different speech sounds. In general the outer grid-lines of the hyper-grid are set to $D = 1$, and training signals are normalized to have a maximum absolute amplitude of one. Concerning the limited number of training data, using $K = 5$ grid lines per dimension, yielding $N_c = K^N = 625$ network centers, is the largest possible choice – otherwise the number of network weights would exceed the number of training samples.

Gaussian basis functions are used in most cases in the literature. Informal experiments with triangular or raised cosine shaped basis functions showed a worse performance of the oscillator model. Also the use of (non-radial) factorized raised cosine basis function [SCAA01] was found inferior. Normalized radial basis functions (eq. 3.5) were reported to achieve good performance in the oscillator model in [Kub96a, CMU02a, CMU02b]. In informal experiments, however, the number of vowel signals that could be stably re-synthesized using normalized radial basis functions with centers on a hyper-grid without regularization, i. e., trained according to eq. 3.8, was below the number found for regularized RBF networks using cross-validation[4]. The Gaussian basis function thus seems to be the favorable choice.

The influence of basis function width on the prediction error for speech signals has been investigated in a number of publications (e. g., [MM01, CMU02a]). Generally, a certain minimum basis function width has been found necessary to achieve acceptably low prediction error, and the prediction error is virtually constant or slightly increases for increasing basis function width beyond an optimum value. The considerations in sect. 3.1 (cf. fig 3.3) that the basis function width should be tied to the spacing of the center grid are confirming these findings. The basis function width will thus be set to $d_{\mathrm{BF}} = d_{\mathrm{diag}}$ (see eq. 3.3 and fig. 3.2) in the following if not stated explicitly otherwise.

4.3.2 Need for regularization

If the nonlinear function $f(\cdot)$ is approximated by an RBF network with a priori fixed center positions we first can confirm findings in [Man99, YH01, MM01]:

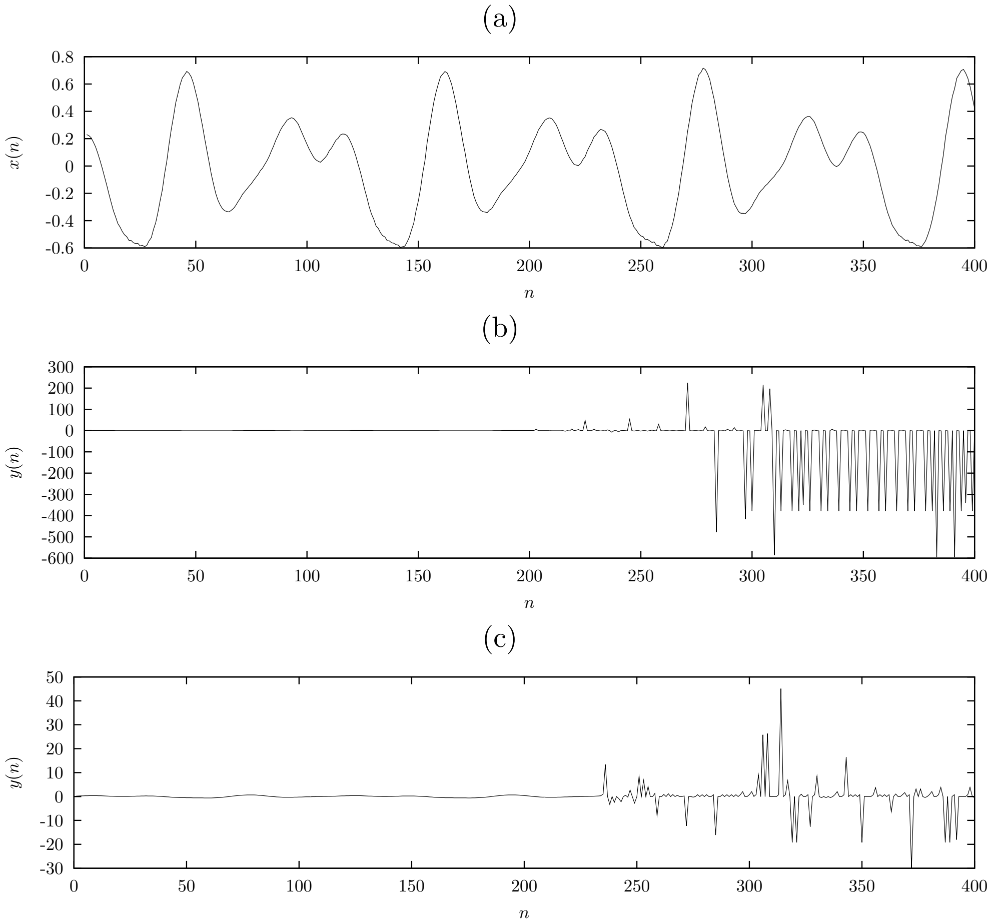
1. Using an RBF network without regularization often yields large amplitude intermittent peaks (spikes) in the oscillator output signal, fig. 4.6 (b). This can already be explained by the finding that some network weights might take large values – especially in the regions of input space not covered by training data – to model small variations in the training output data (cf. fig. 3.4 on page 26 in sect. 3.1). Thus, regularization is necessary to yield stable synthesis with an RBF network used as nonlinear function model in the oscillator.

2. When varying the amount of regularization for an RBF network, for some of the full vowel signals, the oscillator output signal properties could be adjusted in the following way: For high λ the output signal is commonly periodic with little high-frequency components. Reducing λ more and more high-frequency components are generated, and the signal may also display cycle-to-cycle variations. Oscillator output signals for varying regularization factor are depicted in fig. 4.7.

Ad 1: The height of the intermittent peaks can be reduced by excluding centers with a certain minimum distance from all input training data from the learning process, i. e., by the center pruning method described in sect. 3.1.1. The somewhat improved output signal is depicted in fig. 4.6 (c). Nonetheless, the generated signal is no satisfactory approximation for the original signal at all. In addition to regularization the simple pruning method, however, is a first step to reduce the height of undesired peaks in the network output function if fixed RBF centers are distributed in input space and the training input data do not cover the whole region occupied by the centers.

Ad 2: The possibility to adjust the output signal quality for one specific training signal by means of varying the regularization factor seems to be a great convenience at a first glance. If the amount of higher-frequency components can be controlled by the regularization factor, one might even think of generating synthetic speech signals with varying spectral properties

[4] A similar finding is stated in [Man99].


(a)

(b)

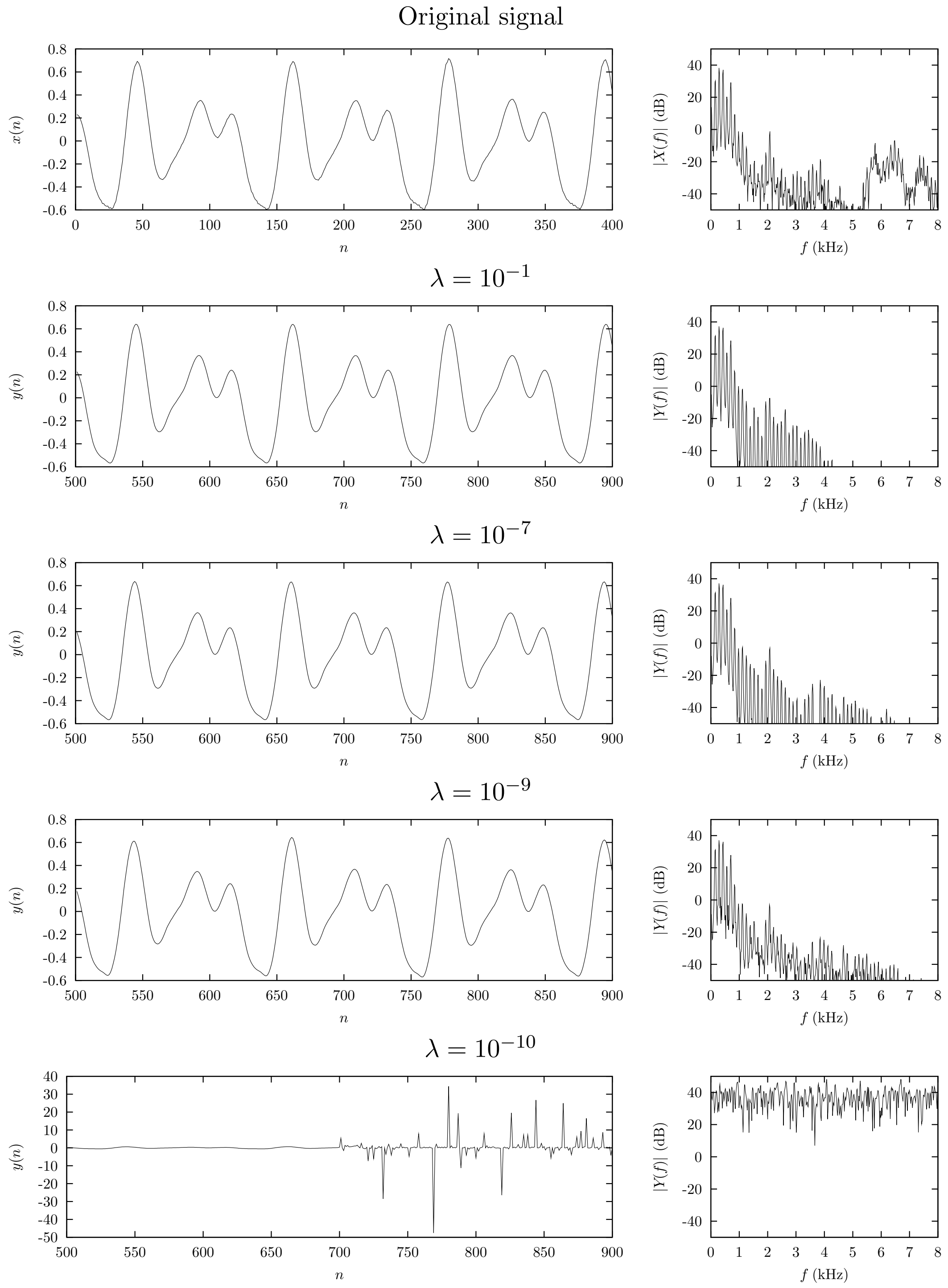
(c)

Figure 4.6: Typical behavior of an oscillator model using an RBF network without regularization for the nonlinear function model. (a) Original signal, (b) full network training, (c) pruning of distant centers. Embedding parameters are $N = 4$, $M = 13$. Network parameters are $K = 5$ ($N_c = 625$), Gaussian basis functions with $d_{\mathrm{BF}} = 2\,\sigma_g = d_{\mathrm{diag}}$ (cf. eq. 3.4 on page 21). Training is performed on $P = 3000$ signal samples. The distance threshold for pruning centers is set to $d_{\mathrm{prun}} = 2\,\sigma_g$, resulting in pruning of 144 network centers for this signal.

from one training signal, e. g., to produce different speaking styles [Ter02b]. From fig. 4.7, however, we see that by varying the regularization factor – though yielding a differing spectral distribution – the high-frequency components of the full speech signal above 5 kHz cannot be re-synthesized faithfully by the oscillator model. If regularization is decreased below a certain value the oscillator becomes unstable producing similar output as without regularization, rather than reproducing the high-frequency components correctly.

Moreover, the properties of the *training signals* require an individual amount of regularization for different sounds or speakers. Thus, the manual optimization of the regularization factor would be a time-consuming task. Automatic determination of the regularization factor by cross-validation or the Bayesian training algorithm is, therefore, a pre-requisite if the oscillator model shall be applied to a large number of different speech signals.

For the signals in fig. 4.7 regularization was applied according to eq. 3.10, i. e., by regularizing matrix inversion. Comparing this way of regularization to GRBF networks, eq. 3.15, little difference in performance was found. The mean square prediction error on the training data as a function of regularization parameter λ of both methods for the signal (vowel /o/) of



Original signal

$\lambda = 10^{-1}$

$\lambda = 10^{-7}$

$\lambda = 10^{-9}$

$\lambda = 10^{-10}$

Figure 4.7: Oscillator output signals for varying regularization factor λ. On the left-hand side the time domain signal is depicted, and on the right-hand side the DFT spectrum computed from 1000 waveform samples. Note the change in y axis scaling in the time signal plot for the case $\lambda = 10^{-10}$. Embedding and network parameters are the same as for the synthetic signals in fig. 4.6, regularization is applied according to eq. 3.10 (regularization of matrix inversion).

figures 4.6 and 4.7 is depicted in fig. 4.8 (a). As can be seen, the prediction error increases with the amount of regularization similarly for both methods within a range of 0.3 dB. Also, the perceptual quality of stably re-synthesized signals is indistinguishable. The condition number of the inverted matrix, fig. 4.8 (b), however, differs by a factor of about 10^3, with the regularization of matrix inversion displaying a lower condition number than for the GRBF approach.

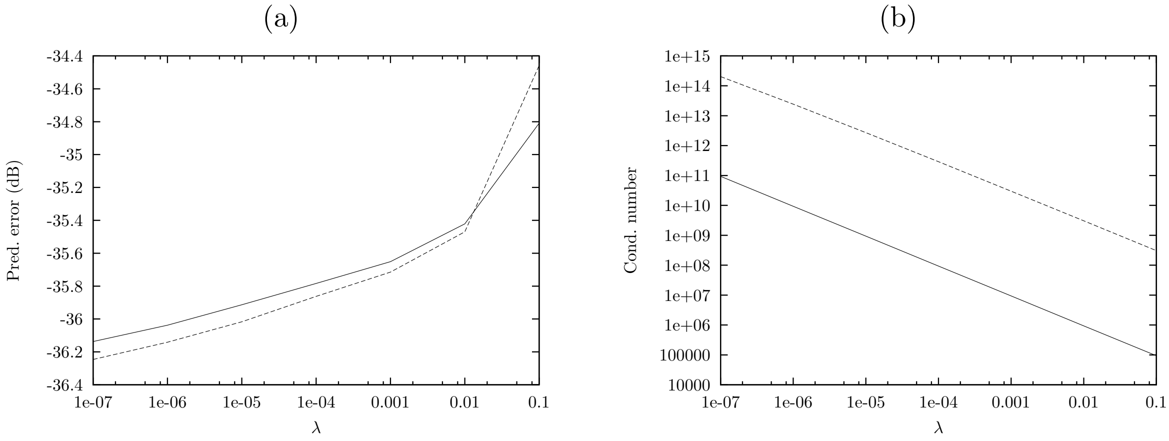


Figure 4.8: (a) Mean squared prediction error and (b) condition number of the inverted matrix, as a function of regularization factor λ for regularization of matrix inversion (solid lines) and GRBFs (dashed lines). Signal and model parameters as in fig. 4.6.

Recalling that the only difference between regularization of matrix inversion and by GRBFs is the presence of matrix $\boldsymbol{\Phi}_0$ instead of the identity matrix, and that $\boldsymbol{\Phi}_0$ has all ones on the diagonal and smaller (positive) values on the off-diagonal elements, this comes with no surprise. Especially for not too high basis function widths – as in our case $d_{\mathrm{BF}} = d_{\mathrm{diag}}$ (cf. pp. 21f) – most of the off-diagonal elements are close to zero (for the network used in the examples 92% are smaller than 0.1). Thus, the two methods for regularized RBF network training seem to be almost equally powerful, with the exception that for GRBFs a higher condition number will sooner lead to numerical problems during matrix inversion when the amount of regularization is decreased.

4.3.3 Determining the regularization factor

We will now look at automatic determination of the regularization factor for RBF network training, either by cross-validation or by Bayesian network training.

Using cross-validation on the prediction error to find an optimal regularization factor λ for RBF training requires to set aside a part of the training data for validation. As has been previously noted, this is troublesome for speech signals that are only short-term stationary in general, i. e., there is only a limited number of training samples available for one sound.

Applying a simple cross-validation procedure to GRBF training, with a randomly chosen cross-validation set comprising 10% of the training data ($P = 3000$, so 2700 samples were used for training and 300 for cross-validation), we found that for vowel signals the resulting regularization factor varies over a wide range. Examples for the resulting regularization factor λ_{xv} from cross-validation for vowel signals from one speaker for full network training (using all centers) are given in table 4.1. λ values for cross-validation were varied in decade steps over the range $\lambda \in [10^{-12}, 10^2]$.

The Bayesian algorithm for learning noise variance and regularization factor (sect. 3.1.3) yields a regularization factor λ_{bay} correlating to some degree with the one found by cross-validation, but generally assigns a higher value (in contrary to the results for the one-dimensional task in Chapter 3, cf. table 3.2 on page 42). Values found for λ_{bay} for vowels from one speaker are also stated in table 4.1. The correlation coefficient between λ_{xv} and λ_{bay} on a logarithmic scale is $c(\ln \lambda_{\text{xv}}, \ln \lambda_{\text{bay}}) = 0.546$ for all male vowel signals in the database. The geometric mean of the quotient of λ_{bay} and λ_{xv} is $\text{gmean}(\frac{\lambda_{\text{bay}}}{\lambda_{\text{xv}}}) = \exp\left(\text{mean}(\ln \frac{\lambda_{\text{bay}}}{\lambda_{\text{xv}}}\right)) = 1.76 \times 10^4$. Generally speaking the Bayesian algorithm assigns a regularization factor four orders of magnitude higher than cross-validation does.

For the male vowel signals in our database, the mean number of iterations in the Bayesian training process to achieve less than 1% variation in the hyper-parameters α and γ is 19.3.

Table 4.1: Regularization factor found by cross-validation λ_{xv} and Bayesian training λ_{bay} for several vowels from one male speaker for full network training. Embedding parameters were $N = 4, M = 13$, network size $K = 5$ ($N_c = 625$), using Gaussian basis functions with width $d_{\text{BF}} = d_{\text{diag}}$.

Vowel	/a/	/e/	/i/	/o/	/u/	/ø/	/ɛ/	/ʏ/
λ_{xv}	1×10^{-5}	1×10^{-5}	1×10^{-10}	1×10^{-4}	1×10^{-3}	1×10^{-9}	1×10^{-6}	1×10^{-10}
λ_{bay}	4.7×10^{-3}	1.7×10^{-4}	1.5×10^{-5}	1.0×10^{-3}	2.3×10^{-3}	2.3×10^{-6}	4.7×10^{-4}	6.8×10^{-6}

The computational complexity of both the Bayesian training and cross-validation mainly lies in the matrix inversion (eq. 3.10 or 3.15 for cross-validation, and eq. 3.22 for Bayesian learning), which is of order $\mathcal{O}({N_c}^3)$. For the Bayesian training one matrix inversion is required per iteration, whereas for cross-validation one matrix inversion is needed for each value of λ, multiplied by the number of validation sets if k-fold cross-validation is used. Furthermore, the network output has to be computed for all training data in each iteration in the Bayesian case, and for the training data in the validation set for each λ value for cross-validation. The nonlinear part (eq. 3.9), however, has to be calculated only once for each training signal, and the computation of the network output for the training input data reduces to a vector-matrix multiplication. The complexity is thus dominated by the matrix inversion, and determined by the number of network centers, as well as the number of iterations in Bayesian training, or the number of cross-validation sets times number of λ values of interest for cross-validation.

The search for the optimal regularization factor over the range $\lambda \in [10^{-12}, 10^2]$ using cross-validation thus requires a comparable computational complexity when using one validation set and one λ value per decade – namely 15 matrix inversions – as does the Bayesian training with 19.3 iterations on average. If, however, we increase the number of λ values for the cross-validation procedure to get a finer grained result, or if we search for the regularization factor using cross-validation with more than one validation set, the computational complexity for cross-validation exceeds the one for the Bayesian training considerably.

Re-synthesis of male vowel signals from our database yields a number of only 9 successfully[5] re-synthesized signals for regularized training according to cross-validation, and 16 for Bayesian training (of 88 vowel signals from male speakers in total). For vowel signals from female speakers we yield comparable numbers. The low number of successfully re-synthesized signals is somewhat discouraging. In [Man99, MM01] it was stated that all of a number of vowel sounds could be stably re-synthesized if the regularization factor found by cross-validation was increased. Considering that the regularization factor from Bayesian training generally is

[5] Cf. definition of successful re-synthesis in sect. 4.2.

higher than the one found by cross-validation, the model based on Bayesian training should have been more successful.

Many of the re-synthesized signals display large-value intermittent peaks (spikes) like the signal in fig. 4.7 for $\lambda = 10^{-10}$. Recalling that RBF networks without or with a small amount of regularization often yield large peaks for input values outside the region of training data (cf. fig. 3.4 (c),(d)) and that the trajectory of a vowel signal in phase space does not cover the phase space occupied by network centers, the pruning of centers should be advantageous. When applying network pruning (sect. 3.1.1) we observe a reduction in amplitude of the intermittent spikes – but no significant change in general behavior: The number of successfully re-synthesized male vowels does not change for Bayesian learning and even decreases for cross-validation. Also, if regularization is increased by taking $\lambda = 100\lambda_{\text{bay}}$ and re-training the RBF network using regularized matrix inversion (eq. 3.10) the number of stably re-synthesized signals increases only slightly. Both pruning and increasing regularization seems to provide no general solution to the problem of instabilities during oscillator synthesis.

Looking at the number of successfully re-synthesized signals as a function of vowel identity, table 4.2, we see that some vowels can be reproduced more easily than others. Of course the chosen embedding and network parameters may not comprise the optimum for all vowel signals. However, again, we want to maintain a certain choice of parameters for all signals. Also, the reason could be that some of the recorded signals are not as stationary as others. However, the numbers in table 4.2 related to the phase-space representations of some vowels in fig. 4.9 suggest that signals with a smoother trajectory in phase space (vowels /o/ and /u/) are more likely to be reproduced adequately than signals displaying a detailed fine-structure (vowels /a/, /e/, and /i/). In sect. 4.4 we will show that inverse filtering of the speech signals yields signals that can be modeled with greater ease for all vowels.

Table 4.2: Number of successfully re-synthesized vowel signals from male speakers applying the oscillator model to full speech signals, for regularization according to cross-validation and Bayesian learning, and by increasing the regularization factor from Bayesian learning. For each of 11 male speakers one signal of each vowel was used for training, the total number of training signals for the vowels listed being 88.

Vowel	/a/	/e/	/i/	/o/	/u/	/ø/	/ɛ/	/ʏ/	total
Cross-validation	0	0	0	2	7	0	0	0	9
Bayesian	0	0	0	6	8	2	0	0	16
$\lambda = 100\lambda_{\text{bay}}$	0	0	1	6	8	1	0	1	17

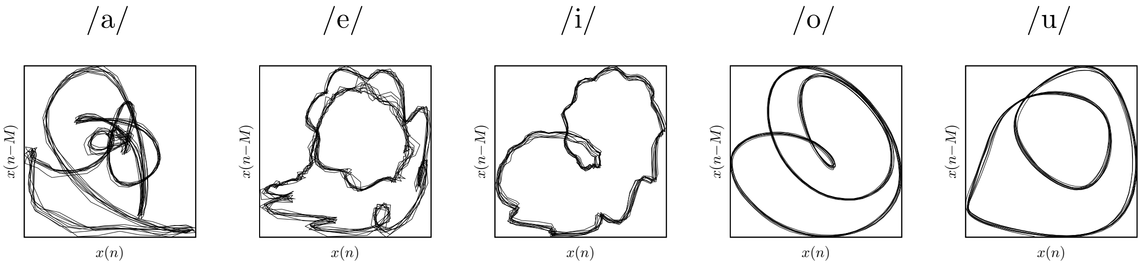


Figure 4.9: Two-dimensional projections of the phase-space representation of vowel signals from one speaker.

Problems synthesizing, e. g., the vowel /i/ have, however, been encountered also using other methods:

> Da eben hier die Rede von dem Selbſtlauter i iſt, ſo muß ich doch eine luſtige Anekdote hier erzählen. In *** kam ein vornehmer durch prächtige Equipage und Ordensbänder ausgezeichneter Herr zu mir, und bat mich ihn meine ſprechende Maſchine, die dazumal noch ziemlich mangelhaft war, hören zu laſſen, beſonders verlangte er, daß ich ihm die Vokale in der gewöhnlichen Ordnung herſagen möchte. Ich entſchuldigte mich, und ſagte ihm, daß mir noch das i fehlte, das ich alles Nachforſchens ungeachtet noch nicht habe ausfinden können. Ey, ſagte er, wie können ſie doch in einer Stadt wie *** wo es an Künſtlern aller Arten wimmelt, hierwegen in Verlegenheit ſeyn; ſollte Ihnen denn hier nicht jemand gleich ein i machen können?
>
> Wolfgang van Kempelen [vK70, footnote pp. 199f].

Apparently, finding a mechanical structure to reproduce the vowel /i/ was more difficult than for other vowels (≫*. . . still missing the i, which I could, despite all inquiry, not find yet.*≪). Here we also encounter a first indication for the difficulty to explain the distinction between a system model and a signal reproduction system: ≫*Ay, he said, how but could you therefore be at a loss in a town like ***, swarmed with artists of all kind; shall not someone here be able to make you an i right away?*≪.

4.3.4 Performance using other nonlinear function models

A brief summary of the performance of the oscillator model for full speech re-synthesis when using the other nonlinear function models described in Chapter 3 is given.

RVM

The RVM training algorithm (sect. 3.1.4) automatically provides both regularization and pruning. As compared to the Bayesian algorithm for regularization only, the RVM may provide an RBF network with a significantly lower number of parameters.

As opposed to [Tip01] – who is using the training data points as network centers – we apply the RVM algorithm with RBF centers on a fixed hyper-grid for the oscillator model. As for the RBF based models before, we use an $N = 4$-dimensional embedding space, embedding delay $M = 13$, and a number of $K = 5$ grid lines per dimension, i. e., starting out with $N_c = 625$ network centers, and a width of the Gaussian basis functions $d_{\text{BF}} = d_{\text{diag}}$. Training is again performed on a number of $P = 3000$ samples from the vowel signals[6].

During the iterations in RVM training, the number of basis functions (or centers) is greatly reduced. In our implementation of the algorithm the hyper-parameters are initially set to $\alpha_i^{(1)} = 0.1, {\sigma_n^2}^{(1)} = 1$, and a basis function $\varphi_i(\cdot)$ is pruned if the corresponding hyper-parameter α_i exceeds a value of 10^6. For vowel signals this leads to a reduction of network centers during training iterations as depicted in fig. 4.10.

The training process is terminated if the number of network centers N_c does not change during 10 iterations. This leads to a mean number of 60.2 iterations, and 86.6 network centers used in the RVM on average for all male signals in the database (including nasals and liquids).

Interestingly, the signals re-generated successfully by the oscillator model with the RVM as nonlinear function model are essentially the same signals that have been re-generated successfully by the RBF network with Bayesian regularization (listed in table 4.2). Also, the

[6] Using the training data points as centers would lead to a 3000×3000 matrix $\boldsymbol{\Sigma}^{-1}$, the inversion of which is not only impractical on normal scale computers, but which is also most certainly near singular due to the *almost periodic* nature of voiced speech.

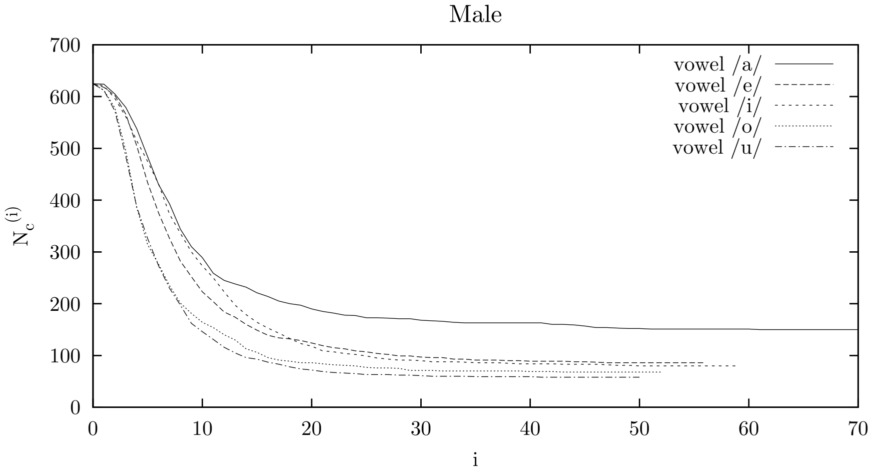


Figure 4.10: Number of network centers $N_c^{(i)}$ used in the RVM as a function of iteration index i in RVM training for some vowel signals from one speaker.

prediction error (on the training data) and the test error (on an unseen test set) is only slightly higher (by less than 1 dB) for the RVM than for the Bayesian trained RBF network. Thus, the RVM seems a valuable option for speech signal prediction with reasonably lower complexity compared to RBF networks with a priori fixed centers, giving almost equal performance as Bayesian regularized RBF networks. However, the number of full speech signals that can be successfully re-generated is equally low as for Bayesian regularization of the full RBF network.

And, as opposed to the one-dimensional regression example in sect. 3.1.4 where the final number of network centers could be reduced by increasing basis function width, for modeling vowel speech signals no significant changes can be observed if the basis function width is increased to $d_{\mathrm{BF}} = 2\, d_{\mathrm{diag}}$. Neither the number of network centers finally used for the RVM, nor the number of iterations, or of successfully re-synthesized vowel signals changes substantially.

As we have seen from the one-dimensional regression example in fig. 3.9 the RVM with a priori fixed centers – like the other RBF based models – may yield large amplitude peaks in regions of input space where no training data are available. Large amplitude intermittent peaks are produced when modeling some vowel signals using the RVM, too.

MLPs

For the speech signal modeling task using an MLP as nonlinear function model, a three-layer MLP (i. e., an MLP with a linear input layer, one nonlinear hidden layer, and one nonlinear output layer) was used, since this structure is proved to have universal approximation capabilities [HSW89]. The same embedding parameters ($N = 4$, $M = 13$) as for the previous models are applied. Hence, the structural parameters of the MLP specified so far are (cf. eq. 3.33): $l = 3, N_1 = 4, N_3 = 1$. We can still choose the nonlinearities $g_l(\cdot)$ in the hidden and the output layer and we can adjust model complexity by the number of nodes in the hidden layer N_2.

The nonlinearities were chosen according to eq. 3.34 (tansig function), with an additional individual bias term for the nodes input. Concerning model complexity it was found that acceptable oscillator behavior can be achieved using an MLP with less adjustable parameters[7] than used in the RBF networks. An MLP with $N_2 = 15$ nonlinear nodes in the hidden layer

[7] Parameters of the MLP that are optimized in the training process.

– leading to a total of 91 adjustable parameters (weights and biases) – already may lead to good results.

Training is performed using back-propagation and the Levenberg-Marquardt algorithm [Mar63] limited to 20 iterations (epochs) in the first case. For individual training runs on the same network structure we find a variety of possible output signal behaviors of the oscillator model as depicted in fig. 4.11 (b)-(e). Different oscillator behavior here is due to the different random initializations of the adjustable network parameters.

With the exception of the network resulting in the signal depicted in fig. 4.11 (b), which has a mean squared prediction error on the training data of about $-18\,\mathrm{dB}$, all other networks in the example (fig. 4.11 (c)-(e)) showed approximately the same training error of about $-30\,\mathrm{dB}$. Hence, the training error cannot be used as a performance criterion regarding stability of the oscillator. Also, the attempt to avoid over-training by increasing the training error target or reducing the number of training iterations (early stopping) does not result in a generally stable behavior of the oscillator. Thus, stability of the oscillator model using an MLP as nonlinear function realization has to be manually inspected for each training signal.

Locally constant approximation

An oscillator with the locally constant model as nonlinear function representation yields stable re-synthesis in any case. Stationary speech signals can be reproduced in a low-dimensional embedding space ($N = 4$), high-frequency components are also reproduced well since the model preserves the noise (cf. sect. 3.3), and even unvoiced phonemes can be reproduced acceptably [KK94]. A time-varying codebook of training data and higher embedding dimension even allows for reproduction and time-scaling of non-stationary speech signals [KK94][8].

MARS

Using a MARS model as realization of the nonlinear function, it has been found that in a 4-dimensional phase-space embedding essentially *no stable re-synthesis* of full speech signals is possible. The oscillator output tends to $+\infty$ or $-\infty$ in all investigated cases. Due to the MARS model assigning splines with nonzero slope at the borders of the region of input training data (cf. sect. 3.2.3, and fig. 3.12 on page 41), an occasional large oscillator output value results in a diverging signal.

In a higher-dimensional embedding ($N = 6$) is was possible to re-generate stable speech signals using the MARS model in rare cases. Most of the oscillator signals still diverge. It has to be noted that the algorithm used (Fortran program by J. Friedman [Fri91]) seems to prevent the construction of models with a high number of parameters from little training data ($P = 3000$ in our case). Thus it was not possible to extend model training above a number of 20 basis functions (101 parameters).

What is more, the oscillator model using MARS has been shown to be able to re-generate simpler oscillatory signals. In [HK98] three oscillator models are used in parallel to re-generate speech signals by using each oscillator for one speech formant. Training signals are derived by bandpass filtering of the original speech signal with the bandpass center frequencies at the formant frequencies. The output signal of each bandpass filter is almost sinusoidal, and with some hand-tuning stable re-synthesis of the bandpass signals could be achieved using oscillators with the nonlinear function realized by MARS. Using several oscillators in parallel, however, introduces the problem of mutual phase synchronization [HK98].

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

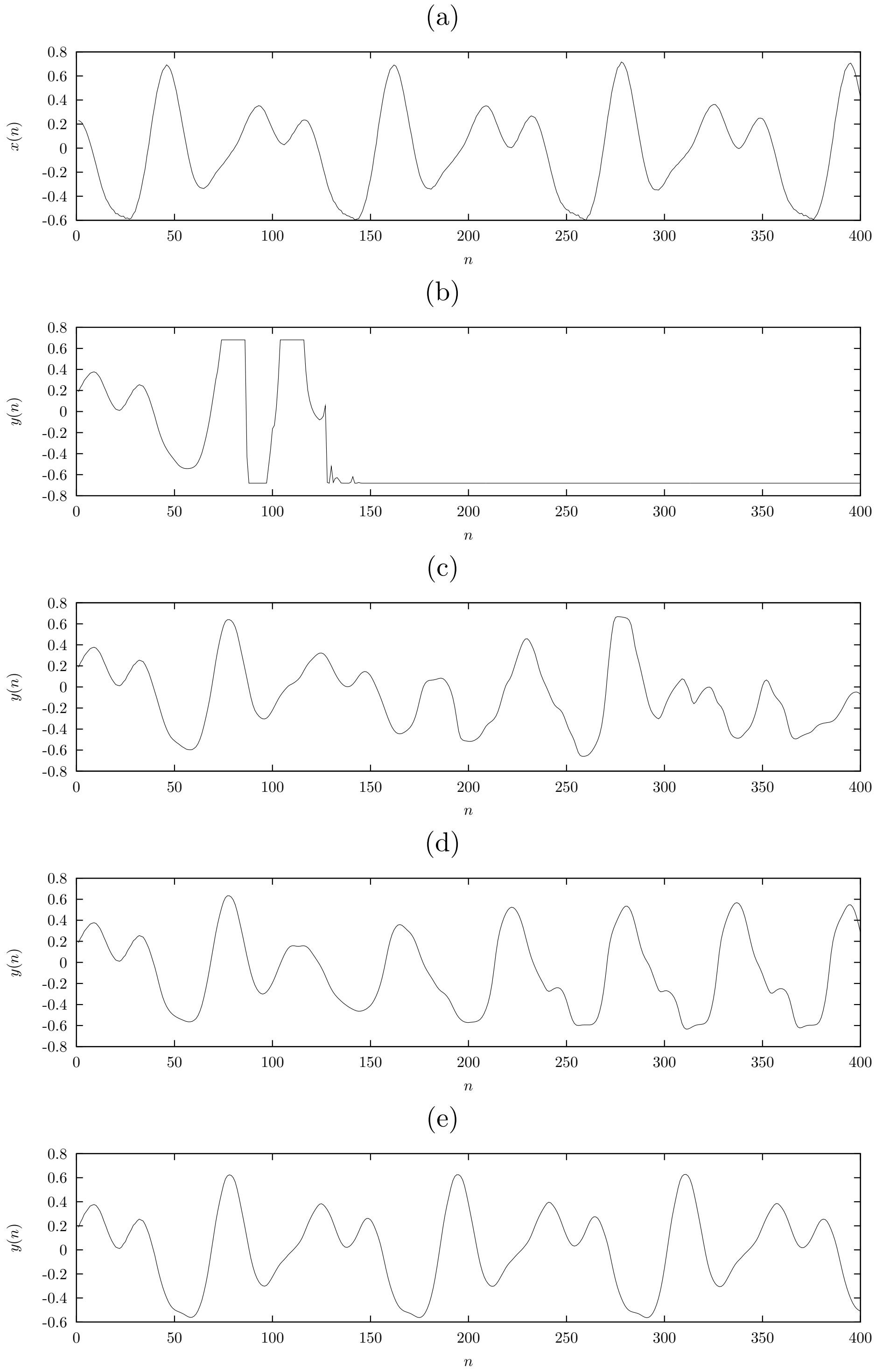


Figure 4.11: Time domain signals of a sustained vowel signal (/o/, male speaker). (a) original signal, and (b)-(e) synthetic signals generated by an oscillator using an MLP as nonlinear function model. All signals (b)-(e) are generated using the same network structure and training process (but different random initializations for the network weights).

4.3.5 Properties of re-synthesized signals

As opposed to the low number of successfully re-synthesized signals from the sustained vowel database stated above, it is found that for specific signals re-synthesis is possible, irrespective of the nonlinear function model. For example, the vowel signal /o/ in fig. 4.12 (a) could be re-synthesized using a regularized RBF network trained according to cross-validation or Bayesian learning, the RVM, an MLP, and MARS models. Time signals and DFT spectra of the synthetic signals are given in fig. 4.12 (b)-(f). The respective phase-space representations for the original signal and the re-synthesized signals are depicted in fig. 4.13.

An embedding delay of $M = 13$ was used for all models. The RBF based models, the RVM, and the MLP are applied in an $N = 4$-dimensional embedding. For the RBF models network centers on a hyper-lattice with $K = 5$ grid-lines per dimension are used. This results in a number of $N_c = 625$ centers for the RBF networks. Cross-validation assigns a regularization factor of $\lambda = 10^{-4}$ (λ was stepped in decades in the range $[10^{-12}, 10^{1}]$), whereas Bayesian training yields $\lambda = 1.0 \times 10^{-3}$ after $N_i = 13$ training iterations. The RVM using a basis function width $d_{\text{BF}} = 2d_{\text{diag}}$ (twice as high as the other RBF networks) could be applied with only $K = 4$ grid-lines, i. e., starting with $N_c = 256$ centers, which were reduced to a number of 64 centers used in the RVM after $N_i = 44$ training iterations. The MLP comprised three layers, with 15 nodes in the hidden layer, and tansig nonlinearities used in the hidden and the output layer. The MARS model with 20 basis functions had to be applied in an $N = 6$-dimensional embedding space to achieve stable re-synthesis.

From fig. 4.12 we can see that the fundamental waveform shape of the re-synthesized signals matches well with the original waveform shape, visible also in the phase-space representations in fig 4.13. Also the low frequency formants – important for the perceived vowel identity – are produced. However, for all models the spectral envelope clearly differs from the one of the original signal. Spectral ‘troughs’ between formants are less ‘deep’, and above about 4 kHz the spectra of the synthetic signals commonly are rather flat, whereas the original signal still holds some higher formants. These attributes have to be considered general properties of *full speech signal* synthesis using a *low-dimensional oscillator model.*

Thus, besides the problem of stability, the oscillator model applied to full speech signals also results in dissatisfying reproduction of wide-band speech signals. In the next section we will show how spectral reproduction – and stability – can be improved if we fall back upon linear prediction as a model for the vocal tract.

4.4 Combined LP and oscillator model

In this section we will reason for combining the oscillator model with inverse filtering by linear prediction (LP). Inverse filtering and LP filtering is introduced, and the means for successfully modeling a ‘glottal signal’ are depicted. However, let us start with a quote about the voiced speech source for vowels from [vK70]:

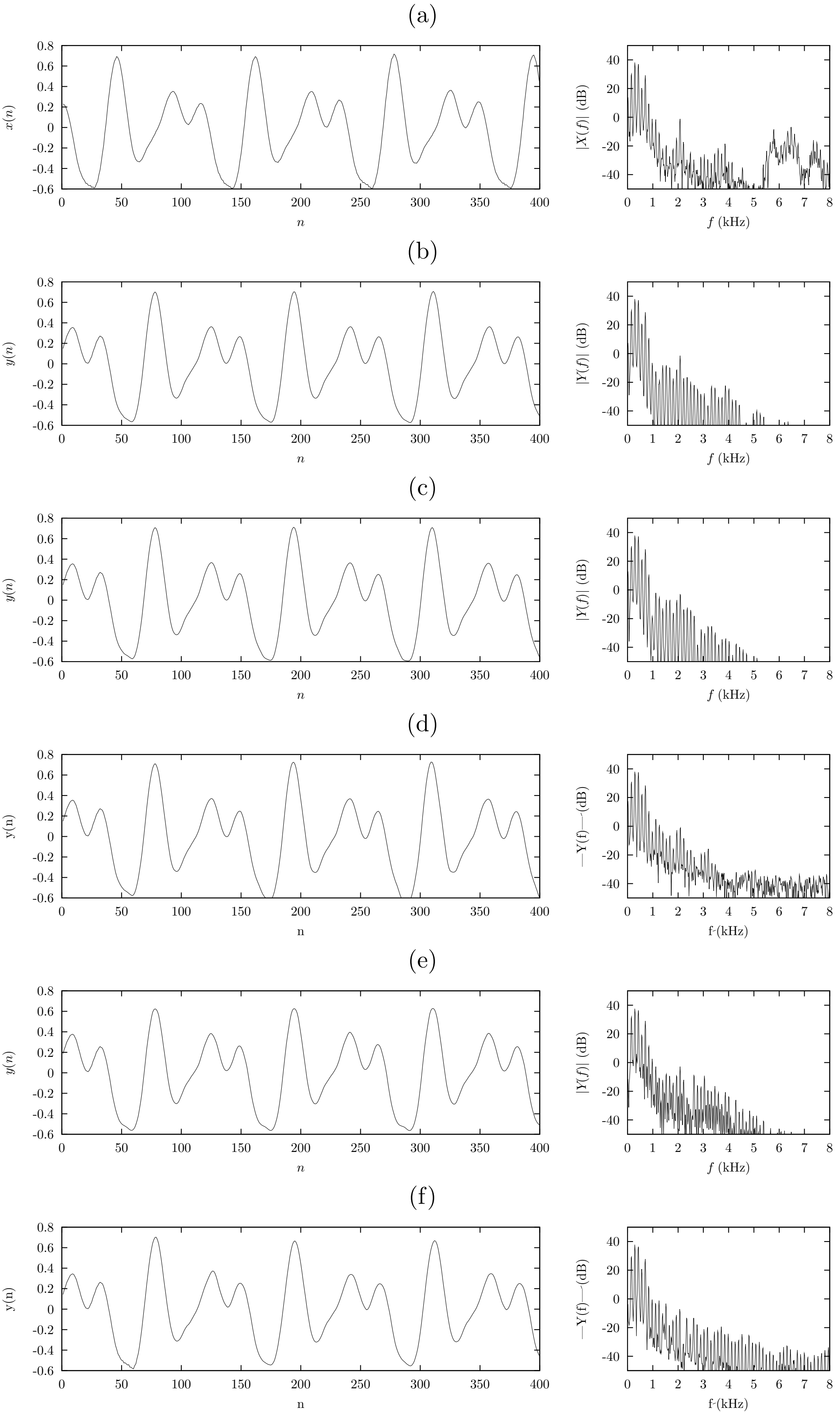


Figure 4.12: Time domain signals and DFT magnitude spectra of a sustained vowel signal (/o/, male speaker). (a) original signal, (b)-(f) signals re-generated by the oscillator model using (b) a regularized RBF network with cross-validation, (c) a Bayesian regularized RBF network, (d) an RVM, (e) an MLP, and (f) a MARS model for the nonlinear function.

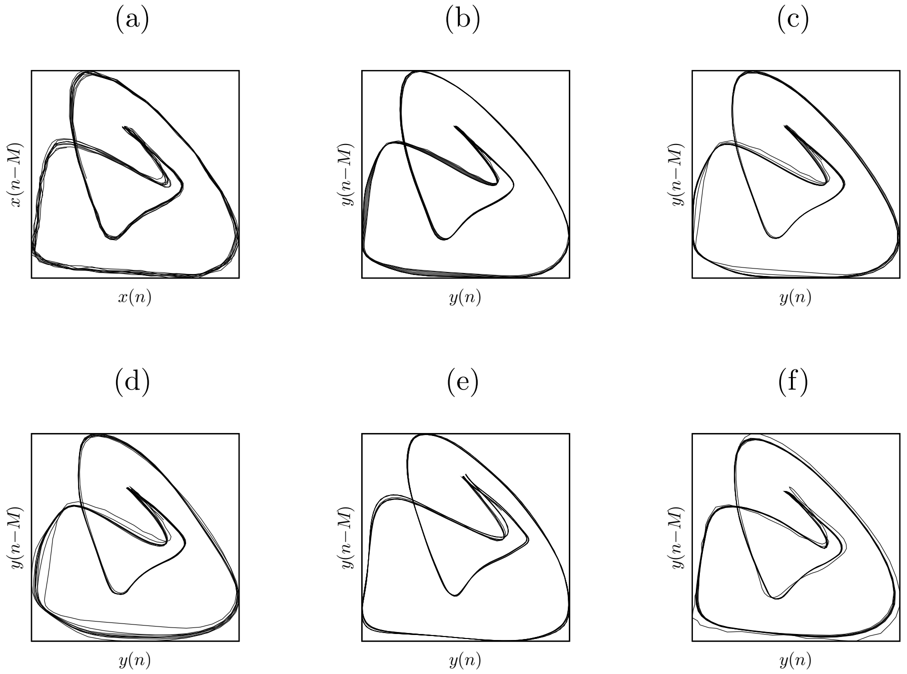


Figure 4.13: Two-dimensional phase-space representations of the signals in fig. 4.12.

1) Die Stimmritze tönet bey dem einen wie bey dem anderen immer gleich, und zwar bey geſchloſſener Naſe.

2) Die Stimme wird, ſo wie ſie aus der Kehle kömmt, durch die Zunge gleichſam wie durch einen Kanal gerade den Lippen zugeleitet. Je mehr ſich die Zunge bey dieſer ihrer Verrichtung, beſonders mit ihrer hintern Hälfte hebt, oder niederlegt, je enger oder weiter wird dieſer Kanal: je enger oder weiter dieſer iſt, je verſchiedener iſt der Laut.

3) Die weitere oder engere Oeffnung des Mundes vollendet endlich den Laut ganz, und gibt ihm ſeine Reinigkeit. (…)

Ein Selbſtlauter iſt alſo ein Laut der Stimme, der durch die Zunge den Lippen zugeführt, und durch ihre Oeffnung herausgelaſſen wird. Der Unterſchied zwiſchen dem einen und dem anderen Selbſtlauter wird durch nichts anderes zuwege gebracht, als durch den weiteren oder engeren Durchgang, den entweder die Zunge, oder die Lippen, oder beyde zuſammen der Stimme geſtatten.

Wolfgang van Kempelen [vK70, § 105/106, pp. 189f]. Von den Lauten oder Buchstaben: Von den Selbstlautern.

What Wolfgang van Kempelen depicts here is that the voiced speech source ≫*is the same for all vowels*≪, and that the ≫*differences between vowels are made up by different positions of tongue and lips*≪. Though this may seem a very simplified point of view today, it is the basis of source-filter modeling and estimation of the glottal source signal by inverse filtering.

4.4.1 Inverse filtering and source-filter modeling

Inverse filtering of speech refers to the task of determining an estimate of the glottal source signal from the full speech signal. The most common approach to inverse filtering is linear prediction (LP). LP filtering is utilized in a broad range of speech transmission systems, for speech analysis, in speech recognition and in many speech synthesis algorithms (see Chapter 2).

Application of inverse filtering by linear prediction in combination with the oscillator model may be questioned since the nonlinear prediction used in the oscillator can theoretically always outperform a linear predictor in terms of mean squared prediction error [Kub95]. In [TNH94] and [FMV97], e. g., nonlinear prediction by Volterra series or by an MLP is shown to yield a higher prediction gain than linear prediction on speech signals. Here, however, the nonlinear models are applied to the same vector of past samples as the LP filter (i. e., $M = 1, N = N_{\mathrm{LP}} = 10$), and thus the nonlinear models have a quite high-dimensional input space (i. e., complexity) and even so have a short memory.

In [BBK97] a nonlinear predictor of considerably reduced complexity is shown to be able to outperform the linear predictor in a *long-term prediction* task. Improvements in prediction gain by using a combination of nonlinear prediction by an RBF network and linear prediction were found in [Tow91, PMdM99].

In our view, the relation between the linear prediction filter and the influence of the vocal tract on the acoustic glottal signal justifies the use of additional LP filtering for speech synthesis. The nonlinear system modeling task is shifted from modeling the full speech signal to modeling an estimate of the output signal of the physical oscillator in speech production, the acoustic glottal pressure signal. We will see that, for vowel signals which could not be successfully re-synthesized by the oscillator model in the full speech signal domain, LP inverse filtering and the application of a simple low-pass filter yields signals that are easier to model – since they display a simpler structure in phase space than the full speech signals, in general. Moreover, the signals for different vowels are more similar in the LP residual domain than the full speech signals. Thus, an LP front-end shall favor the use of the oscillator model for non-stationary speech modeling, e. g., for the generation of transitions between phonemes.

The benefit of using LP in addition to nonlinear modeling for speech signals can also be argued for from a nonlinear system theoretical point of view: Considering a speech production model, the oscillatory source signal of voiced and mixed excitation signals is filtered by the vocal tract [Fan70], which is commonly considered an infinite impulse response (IIR) filter. IIR filtering may change the properties of the attractor of a nonlinear system [ABST93]. It shall thus be advantageous to compensate the influence of the vocal tract on the source signal using LP inverse filtering before oscillator training. This has been advocated in [NPC99], and also in our work [RK01] the better reconstruction of high-frequency components has been proved, suggesting a more accurate modeling of the nonlinear dynamics of the speech signal. The argument of the IIR vocal tract filter changing the dimensionality of the speech source signal is, however, not applicable, if the assumptions for LP are valid, i. e., if the vocal tract filter is modeled as a *purely recursive* IIR filter [MG76]: Since the LP inverse filter is an FIR filter, it cannot change signal dimensionality [ABST93], so the same must be true for the complementary IIR synthesis filter, i. e., the vocal tract filtering.

In this section we will give a brief overview of linear prediction analysis and its application in speech synthesis. Different algorithms for estimation of the LP filter coefficients and different filter structures, as well as some elaborate inverse filtering algorithms are briefly referenced. The main concern, however, is the usability of LP for our purpose of speech synthesis using the oscillator model, thus to yield a 'glottal signal' that can be modeled using the oscillator model to re-generate speech signals better than if the oscillator model is applied to full speech signals.

4.4.2 Linear prediction of speech

Linear prediction of speech aims at finding a linear finite impulse response (FIR) filter of length N_{LP} for a given speech signal, commonly with the aim to yield a filter output signal (residual) with minimum energy, or – as an equivalent criterion – with maximally flat (white) spectrum [Mak75, MG76]. As such it can be used to find an estimation of the vocal tract filter and of the acoustic glottal source signal, if the vocal tract is considered an all pole infinite impulse response (IIR) filter. An all pole IIR filter correlates with a physical model of the vocal tract composed of a series of equal length uniform tubes with varying cross section areas [MG76]. Specifically for vowels, the uniform tube model constitutes a good approximation of reality.

Starting with a speech signal, the 'inverse' FIR filter performs a linear prediction for the next sample based on the current and past signal samples, eq. 4.1. The common aim of minimizing the energy of the inverse filter output signal, the residual signal, leads to equations for the filter coefficients based on an estimate of the auto-correlation of the input speech signal. Depending on how the estimate is computed from the input signal the *auto-correlation* method or the *co-variance* method for LP analysis are distinguished, both providing filter coefficients for the direct form filter structure.

Based on the equivalence of a tube model for the vocal tract and the LP (synthesis) filter, other realizations of the discrete-time LP analysis and synthesis filters more closely related to vocal tract physics are available in the form of lattice filters [MG76]. Filter coefficients can as well be estimated using the lattice filter structure by partial correlation (PARCOR) analysis [ISK$^+$72, MC81, HM84]. The actual realization of the LP analysis and synthesis filters does not depend on the LP analysis procedure since the filter coefficients can be converted between direct form and lattice filter coefficients (using the Levinson-Durbin recursion). However, different behavior of the direct form and the various lattice filter realizations concerning transients can be observed if the filter coefficients are switched [Kub86, Ran99].

Linear prediction is used in the coding algorithms of most digital voice telephony systems, but it also provides the basis of many speech synthesis algorithms [OS76, Str78, SA79, Oli80, Hei82, Ata83, ČBC98, Ran02] (see also Chapter 2). LP analysis methods specially suited for speech analysis [Alk92] and synthesis [Ans97, AKM98, Pea98, FNK$^+$98, FNK$^+$99] have been developed.

Particularly specific algorithms for determining the glottal source signal based on LP analysis [WMG79, Hes83, Wok97] may seem useful for our purpose. Unfortunately, these algorithms are not invertible for synthesis.

As the relation of LP to the uniform tube model has been pointed out, we should also mention analysis/synthesis algorithms based on the simulation of acoustic wave propagation in the tube model (for recent work see, e.g., [EMPSB96, SL00, SL01]). Due to the more appropriate physical modeling, these algorithms can easily include effects not modeled by standard LP, e. g., dissipation, additional sources along the vocal tract, or interaction between the vocal tract and glottal source signal.

The uniform tube model LP may constitute no good approximation of natural human speech production for some sounds like, e. g., nasals. Here the mouth is closed and the air pressure waves primarily emerge from the nose. Acoustic tube models including the nasal cavity have been developed (e. g., [Coo93, SL03]), however, model parameters are difficult to estimate, and the algorithms are not applicable for inverse filtering of the full speech signal yet.

Summing up, it may be said that linear prediction is a proven and valuable tool for speech processing. The LP synthesis filter represents the uniform tube model of the vocal tract. The LP analysis filter must be minimum phase to ensure a stable LP synthesis filter. We have to bear in mind that this might pose problems for speech sounds other than vowels because, e. g., the all-pass part of a nasal signal will still be present in the LP residual signal, or for unvoiced speech sounds like fricatives which are not excited at the glottis. However, for vowel signals a representation of the vocal tract filtering by linear prediction and modeling by a combination of linear prediction and the oscillator model shall clearly be advantageous.

4.4.3 Estimation of the 'glottal signal'

Combining linear prediction and the oscillator model for speech modeling is not straightforward since LP inverse filtering of speech signals yields a signal that can hardly be identified by the oscillator model [Kub95]. The phase-space representation of a vowel signal (vowel /o/) and its LP residual signal in fig. 4.14 may illustrate the problem: Identifying the signal dynamics of the residual signal by the nonlinear function in the oscillator model can be considered as difficult as visually identifying the signal trajectory in the phase-space representation. Hence, the oscillator model will certainly fail to reproduce a residual signal like the one depicted in fig. 4.14 (b).

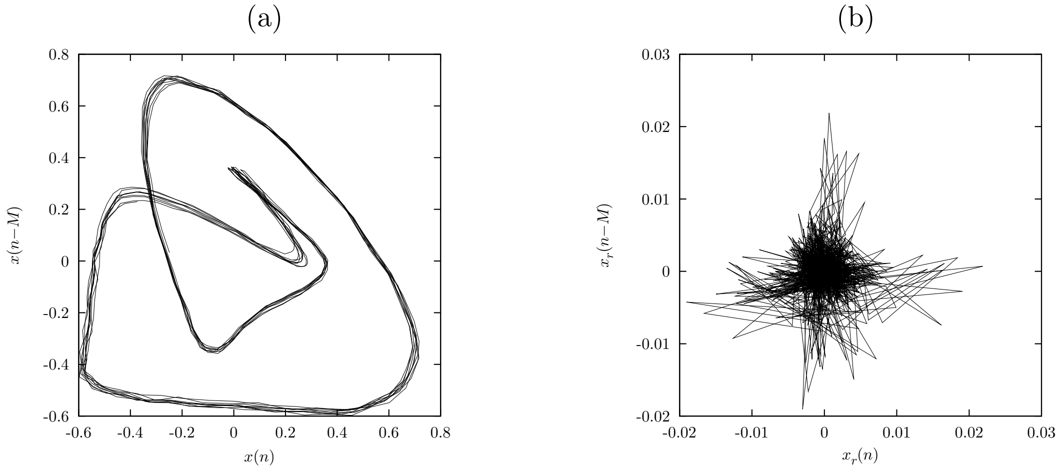


Figure 4.14: Two-dimensional projection of the phase-space representation of (a) the full speech signal $x(n)$ of vowel /o/ and (b) the corresponding LP residual signal $x_r(n)$.

The problem is that LP inverse filtering yields a residual signal with minimum energy, or equivalently with a maximally flat (white) spectrum. This energy minimization is a benefit for speech coding, leading to a considerable reduction of the bit rate. For our speech modeling task, however, it is obviously not an optimum approach.

In speech production models [Fan70] the voiced source signal is considered to have a spectral slope of -12 dB per octave above a frequency of 100 Hz. Lip radiation introduces a high-pass filtering with a slope of $+6$ dB per octave. Assuming that the vocal tract filtering does not alter the overall spectral slope, the full speech signal of voiced speech sounds – when recorded in the acoustic far-field – should display a total spectral slope of -6 dB per octave. For adequate inverse filtering the total spectral slope of -6 dB per octave should be removed from the signal before LP analysis and whitening by the LP inverse filtering, and an additional low-pass filtering has to be applied to the LP residual signal to yield a spectral slope of -12 dB per octave in the estimated 'glottal signal'.

Compensating the spectral slope of the full speech signal is done by applying a high-pass *pre-emphasis* filter before LP analysis. For discrete time signals this is commonly done by computing $x_{\mathrm{em}}(n) = x(n) - k_{\mathrm{em}} x(n-1)$, i. e., by applying a first order FIR filter

$$H_{\mathrm{em}}(z) = 1 - k_{\mathrm{em}} z^{-1} \quad . \tag{4.8}$$

To achieve a high-pass characteristic the coefficient k_{em} has to be chosen $k_{\mathrm{em}} > 0$. Proposed values are in the range $[0.9, 1]$ for 8 kHz sampling rate [MG76], for example $k_{\mathrm{em}} = 0.94$ [O'S87]. Using the pre-emphasis filter only for LP analysis (i. e., parameter estimation) and not before inverse filtering yields a residual signal with a falling spectral slope of -6 dB per octave, as in fig. 4.16 (b). Yet another filtering with -6 dB per octave is missing to achieve the characteristic of -12 dB per octave for the 'glottal signal'.

A simple effective low-pass filtering can be achieved by applying a one-pole recursive filter

$$H_{\mathrm{lp}}(z) = \frac{1-k_{\mathrm{lp}}}{1-k_{\mathrm{lp}}z^{-1}} \ , \tag{4.9}$$

also called 'lossy integration'. This filter has a gain of 0 dB at low frequencies and a falling spectral slope of -6 dB per octave above the cutoff frequency for $0 < k_{\mathrm{lp}} < 1$. For $k_{\mathrm{lp}} \in (0.9, 1)$ the filter frequency response displays the desired slope of -6 dB per octave in the frequency range of interest as depicted in fig. 4.15. To yield an estimation for the glottal signal this low-pass filter is used in series with the LP inverse filter. An example for the low-pass filtered LP residual signal (without pre-emphasis) is depicted in fig. 4.16 (c).

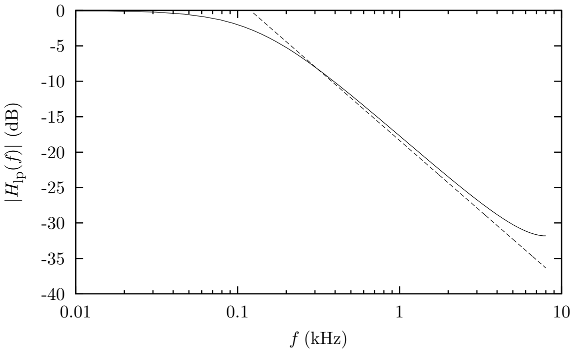
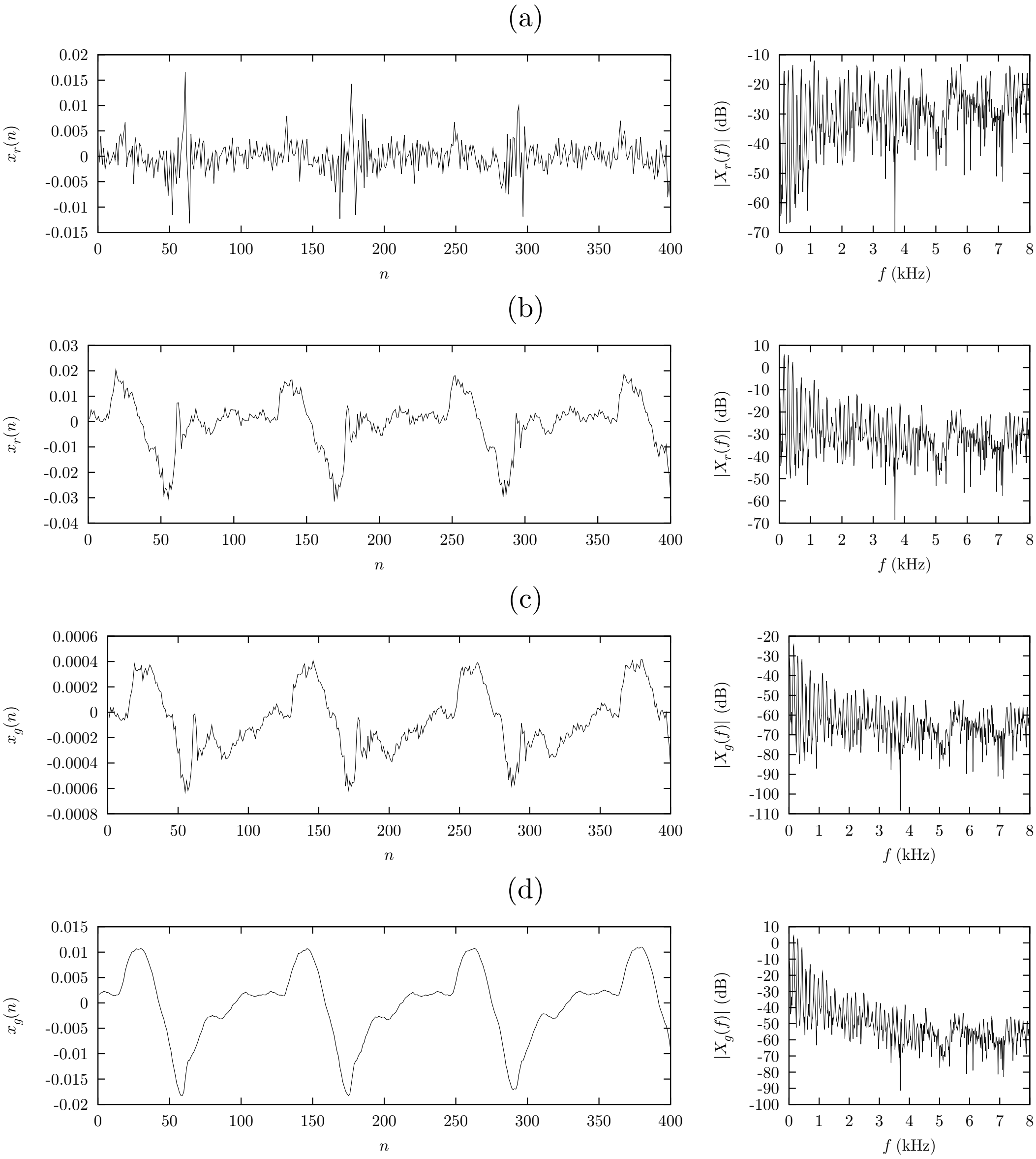


Figure 4.15: Frequency response $|H_{\mathrm{lp}}(f)|$ of a one-pole low-pass filter (solid line) with a pole at $z_p = k_{\mathrm{lp}} = 0.95$ for a sampling frequency of 16 kHz. The dashed line displays a slope of -6 dB per octave.

Combining pre-emphasis and low-pass filtering an estimated glottal signal as in fig. 4.16 (d) is found. Note that lower values for both k_{em} and k_{lp} were used than for the cases of pre-emphasis and low-pass filtering only. The time signals in fig. 4.16 (b-d) well resemble the basic form of the acoustic source signal in the Liljencrants-Fant model (sect. 2.3, fig. 2.2 on page 10).

The property that the signals from LP inverse filtering with pre-emphasis and/or low-pass filtering may be better suited for re-generation by the oscillator model than the LP residual signal without pre-emphasis may be deduced from the phase-space representations in fig. 4.17. Compared to the residual signal of LP without pre-emphasis, case (a), in all other cases (b-d) the trajectory of the signal in phase space can be better identified by the eye, suggesting a less complex signal structure and a more effective modeling by the oscillator.

Concerning the complexity of modeling a signal we can also look at the number of false neighbors in phase space as a function of embedding dimension N. In fig. 4.18 the fraction of false neighbors is plotted for the vowel /o/ as a function of N for the LP residual signal and the low-pass filtered residual signal from LP inverse filtering with pre-emphasis. For the full speech signals of vowels (cf. fig. 4.2) the number of false neighbors approaches zero for $N = 4$. For the LP residual signals without pre-emphasis fig. 4.18 (a), however, the decay is much slower, and a value close to zero is reached not until $N = 7$. Thus, though the FIR LP inverse filtering cannot change the dimension of the signals [SYC91, ABST93], LP inverse filtering emphasizes the higher-dimensional dynamics present in the speech signal, making the signal modeling difficult. Then again, the signals yielded by LP inverse filtering with pre-emphasis and subsequent low-pass filtering fig. 4.18 (b) display a similar characteristic as the full speech signal, indicating that they can be modeled in a low-dimensional embedding space.


(a)

(b)

(c)

(d)

Figure 4.16: Time domain signals and DFT magnitude spectra of an inverse filtered vowel signal (vowel /o/ from a male speaker) using (a) LP analysis without pre-emphasis, (b) LP analysis with pre-emphasis ($k_{\mathrm{em}} = 0.94$), (c) LP analysis without pre-emphasis and low-pass filtering ($k_{\mathrm{lp}} = 0.98$), and (d) LP analysis with pre-emphasis and low-pass filtering ($k_{\mathrm{em}} = 0.75, k_{\mathrm{lp}} = 0.9$).

It can be argued that the low-pass filtering by a recursive, i. e., IIR filter can change the dimension of the signal [ABST93]. Especially a recursive low-pass filtering with a pole close to $z = 1$, which has a long impulse response and nonlinear phase may seem untoward. However, using the same argument as for the IIR vocal tract filter (cf. sect. 4.4.1) it can be shown that a purely recursive low-pass filter does *not* change signal dimension. Moreover, using a purely recursive low-pass filter makes the analysis filtering *invertible*, meaning that the influence of the low-pass filtering can be compensated for during synthesis, which is, for example, not the case if, e. g., a Butterworth low-pass filter with (nearly) linear phase is used, as in [NPC99]. Besides, the one-pole low-pass filter structure is (recursively) used for the phase-space embedding of time series in the 'Gamma model' [dP92, HP98b].

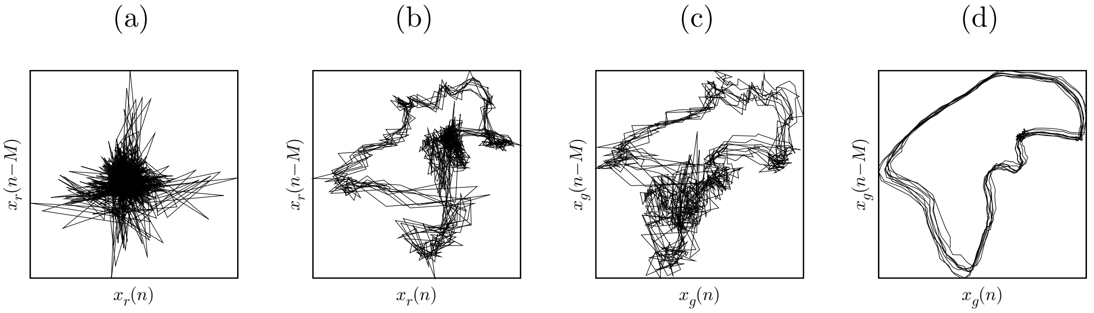


Figure 4.17: Two-dimensional projection of the phase-space representation of the signals in fig. 4.16.

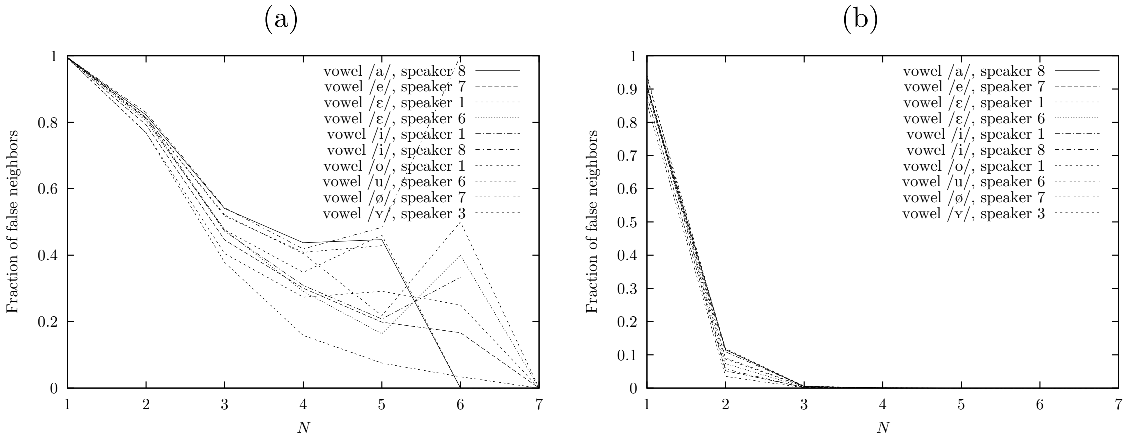


Figure 4.18: Fraction of false neighbors as a function of embedding dimension N for (a) the LP residual signal, and (b) the low-pass filtered residual signal from LP with pre-emphasis for several vowels from male speakers (cf. fig. 4.2 for false neighbors in the full speech signal).

A schematic of the proposed combination of LP inverse filtering and the oscillator model is depicted in fig. 4.19. The recorded full speech signal $x(n)$ is filtered by the pre-emphasis filter $H_{\mathrm{em}}(z)$ to yield the signal $x_{\mathrm{em}}(n)$. LP analysis is performed on $x_{\mathrm{em}}(n)$ and provides the coefficients for the LP inverse filter $A(z)$. The LP residual signal $x_r(n)$ is filtered by the low-pass filter $H_{\mathrm{lp}}(z)$ to yield the estimated glottal signal $x_g(n)$. The oscillator model is trained on $x_g(n)$, i. e., the parameters for the nonlinear function are learned from the trajectory of the time delay embedding $\boldsymbol{x}_g(n)$ drawn from $x_g(n)$. At the bond between analysis and synthesis the signal is parameterized by a number of $N_c + N_{\mathrm{LP}}$ parameters: The RBF network weights and the LP filter coefficients. In the synthesis stage the oscillator model acts as an autonomous signal generator and the low-pass filtering and LP inverse filtering is compensated for by applying the complementary filters, i. e., a high-pass with transfer function $1/H_{\mathrm{lp}}(z)$ and the LP synthesis filter $1/A(z)$.

4.4.4 Synthesis of stationary vowels

The combined LP and oscillator model is applied to signals from the sustained vowel database (see sect. 4.3 on page 51). Assuming stationarity, both LP analysis and nonlinear function learning is performed for a number of $P = 3000$ training samples (136 ms). An LP filter of order $N_{\mathrm{LP}} = 18$ was used, with a pre-emphasis factor $k_{\mathrm{em}} = 0.75$ (sampling frequency

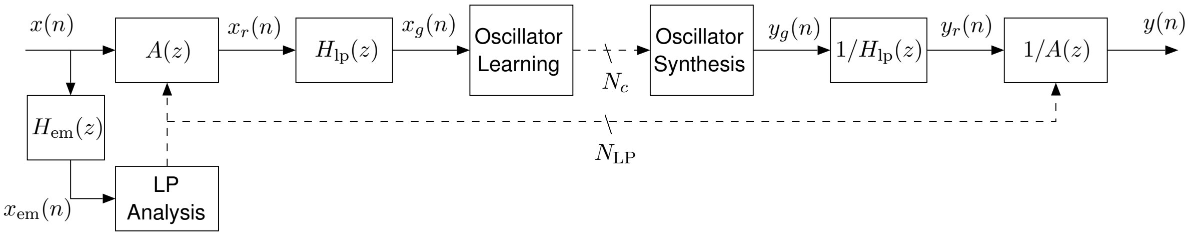


Figure 4.19: Combining LP inverse filtering and the oscillator model

$f_s = 16\,\mathrm{kHz}$). Subsequent low-pass filtering according to eq. 4.9, with $k_{\mathrm{lp}} = 0.95$ results in estimated 'glottal signals' $x_g(n)$ as depicted in fig. 4.20.

As noted before, these signals roughly resemble the form of the acoustic source signal in the Liljencrants-Fant model (cf. fig. 2.2 on page 10). In any case there are periodic negative peaks at the supposed instants of glottal closure. There is, however, no clear distinction between closed and open glottal phase, and for some of the signals 'ripples' are encountered. Ripples are mainly due to the interaction between vocal tract pressure and glottal movement, which is neglected in LP inverse filtering[9]. The shape of the 'glottal signal' also depends on speaker identity [Chi95, PQR99], fundamental frequency, emotional state of the speaker [KS95], etc.

Nevertheless, the estimated 'glottal signals' in general reveal a less complex structure in phase space (fig. 4.21) as compared to the full speech signals (fig. 4.9 on page 57). Commonly the signal trajectories are unfolded to open loops even in the plotted two-dimensional representations. Moreover, the 'curls' in the phase-space representations of the full speech signals for the vowels /e/ and /i/, due to formant resonances, are removed by inverse filtering. From this fact we may already expect that the 'glottal signals' can be modeled easier than the full speech signals.

As depicted in fig. 4.18, the number of false neighbors for the 'glottal signal' of vowels is close to zero for an embedding dimension $N \geq 3$, suggesting that we can again use $N = 4$ for the oscillator model. Determining the optimal embedding delay for the 'glottal signals' by means of MI between delayed signal samples is even less unambiguous than for the full speech signal (cf. sect. 4.1.2). Due to the recursive low-pass filtering MI between delayed samples as a function of delay is smoothed, and a first minimum can be identified even less clearly than for the full speech signals. There is, however, generally a broad minimum starting at a delay $L = 10$ samples for male speakers, and at about $L = 8$ for female speakers suggesting that using the same embedding delay as for the full speech signals ($M = 13$) may not be too wrong here, as well.

Using the same embedding and RBF network parameters as for the full speech signals (sect. 4.3.1) the first finding is that regularization is again mandatory for generating stable oscillator signals. As for the full speech signals the oscillator output signal displays large amplitude peaks if no regularization is applied.

For regularized RBF network learning, however, stable re-synthesis is possible for a significantly higher number of signals. For example, using a fixed regularization factor $\lambda = 10^{-5}$ for a vowel /e/ from a male speaker we obtain the time signals and phase space representations of the signals in the combined LP and oscillator model as depicted in fig. 4.22. This very

[9] For formant synthesis, improvements in quality have been achieved by using source signals including these ripples [PCL89].

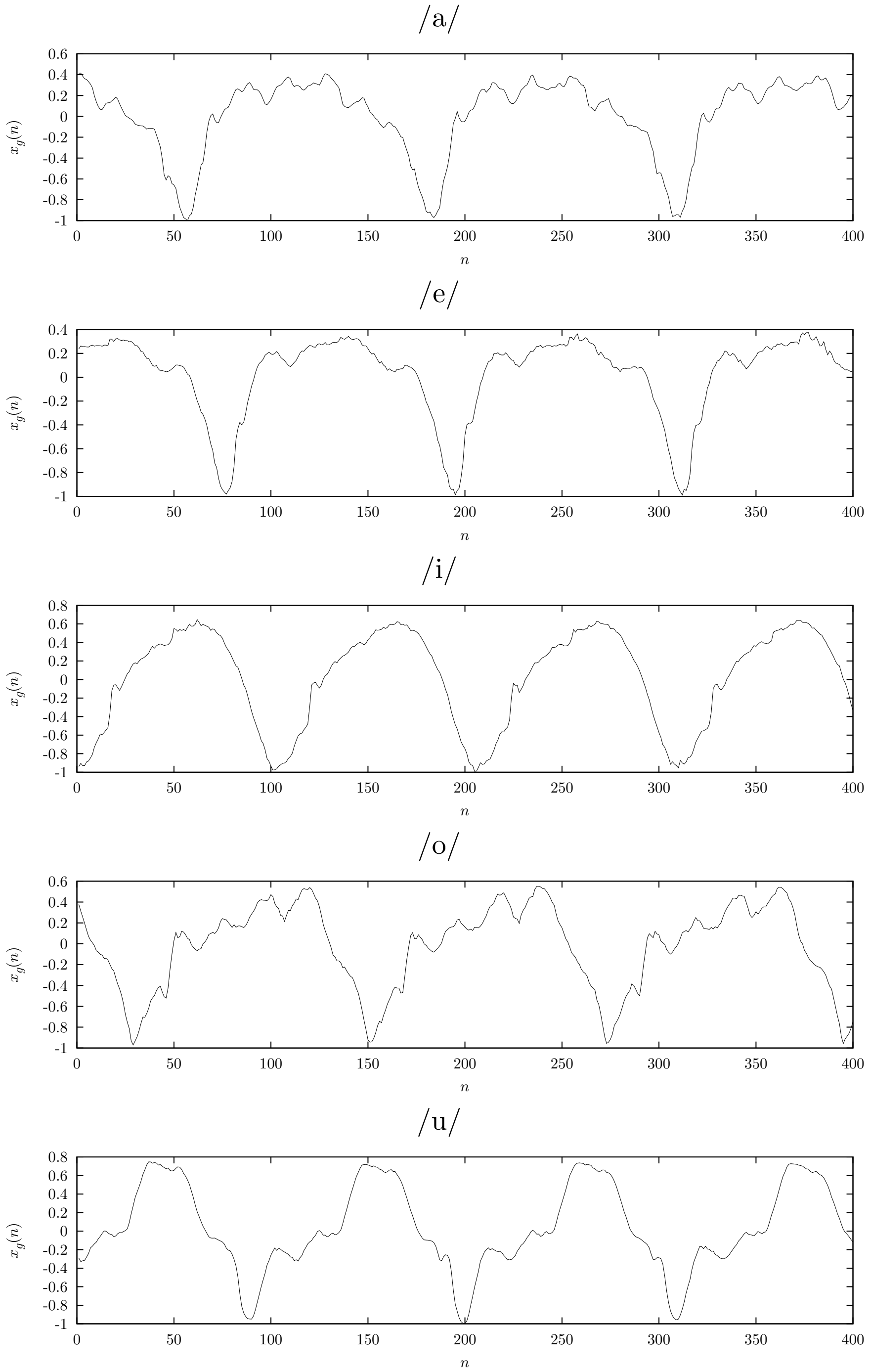


Figure 4.20: Estimated 'glottal signals' $x_g(n)$ of different vowels from one male speaker.

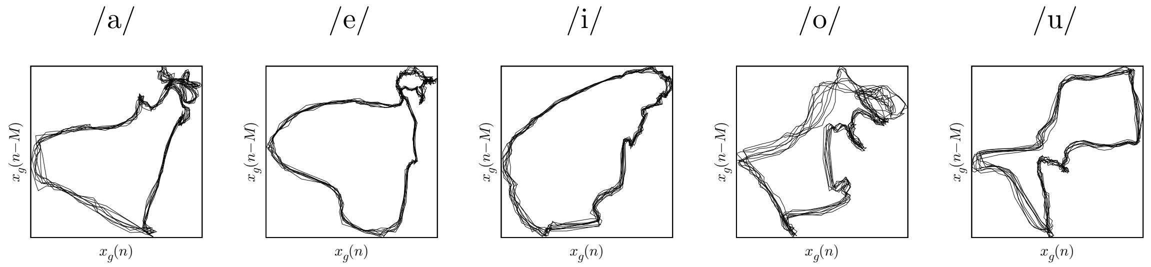


Figure 4.21: Two-dimensional projection of the phase space representation of ‘glottal signals’ (cf. fig. 4.20) of vowels from one male speaker.

signal could not be stably re-synthesized by a model of the same complexity in the full signal domain. Yet, in the glottal signal domain the oscillator can successfully re-generate $y_g(n)$, and subsequent high-pass filtering by $1/H_{\mathrm{lp}}(z)$ and LP synthesis filtering $1/A(z)$ yields a synthetic full speech signal $y(n)$ similar to the original, both in the time domain and in phase space.

From the reappearance of ‘curls’ in the synthetic full speech signal $y(n)$ in fig. 4.22 we can already suppose that the formant structure is reproduced. Comparing the spectra of the original signal and the synthetic full speech signal in fig. 4.23 we find that, in the frequency range up to 4 kHz (important for the distinction of vowels), virtually no difference can be found concerning the spectral peaks.

Comparing the synthetic signals from the combined LP and oscillator model and the oscillator model applied in the full speech signal domain for vowel /o/ (vowel /e/ could not be re-generated in the full signal domain) we find that both the time signal and the phase-space representation are indistinguishable from the original signal. In the spectral domain fig. 4.24, however, the combined LP and oscillator model exhibits a formant in the high-frequency range (at 6 kHz) present in the original signal spectrum, that is not captured in the model for the full speech signal, albeit a lower regularization factor is used in the oscillator model applied to the full speech signal than in the combined LP and oscillator model. In general, the combined LP and oscillator model yields a perceptually more natural reconstruction of the high-frequency range of wide-band speech signals.

Still, the synthetic speech signals sound unnatural. This can be attributed to the periodicity of the oscillator output signal: It is seen from the phase-space representations of the synthetic signals in fig. 4.22 that the trajectories follow a limit cycle – in contrary to the trajectories of the natural signal. The spectrum of the synthetic signal in fig. 4.23 is more like a line spectrum in the higher frequency range than the original signal spectrum, as well. The majority of oscillator generated ‘glottal signals’, and hence also the full speech output signals are strictly periodic, and thus sound unnatural.

Periodicity could possibly be reduced by reducing regularization, thus allowing for more detailed modeling of the nonlinear function. We will next look at automatic determination of the regularization factor.

4.4.5 Cross-validation vs. Bayesian learning

In this section we will compare the determination of the regularization factor of an RBF nonlinear function model by cross-validation and Bayesian learning, for the application of the oscillator model to ‘glottal signals’. Again we consider a possible number of 15 values for $\lambda \in [10^{-12}, 10^{2}]$ (one per decade) in the cross-validation procedure, and we use 300 randomly chosen training data points out of $P = 3000$ for the validation set. Bayesian training is terminated, if the variation of α and γ per iteration is less than 1 %.

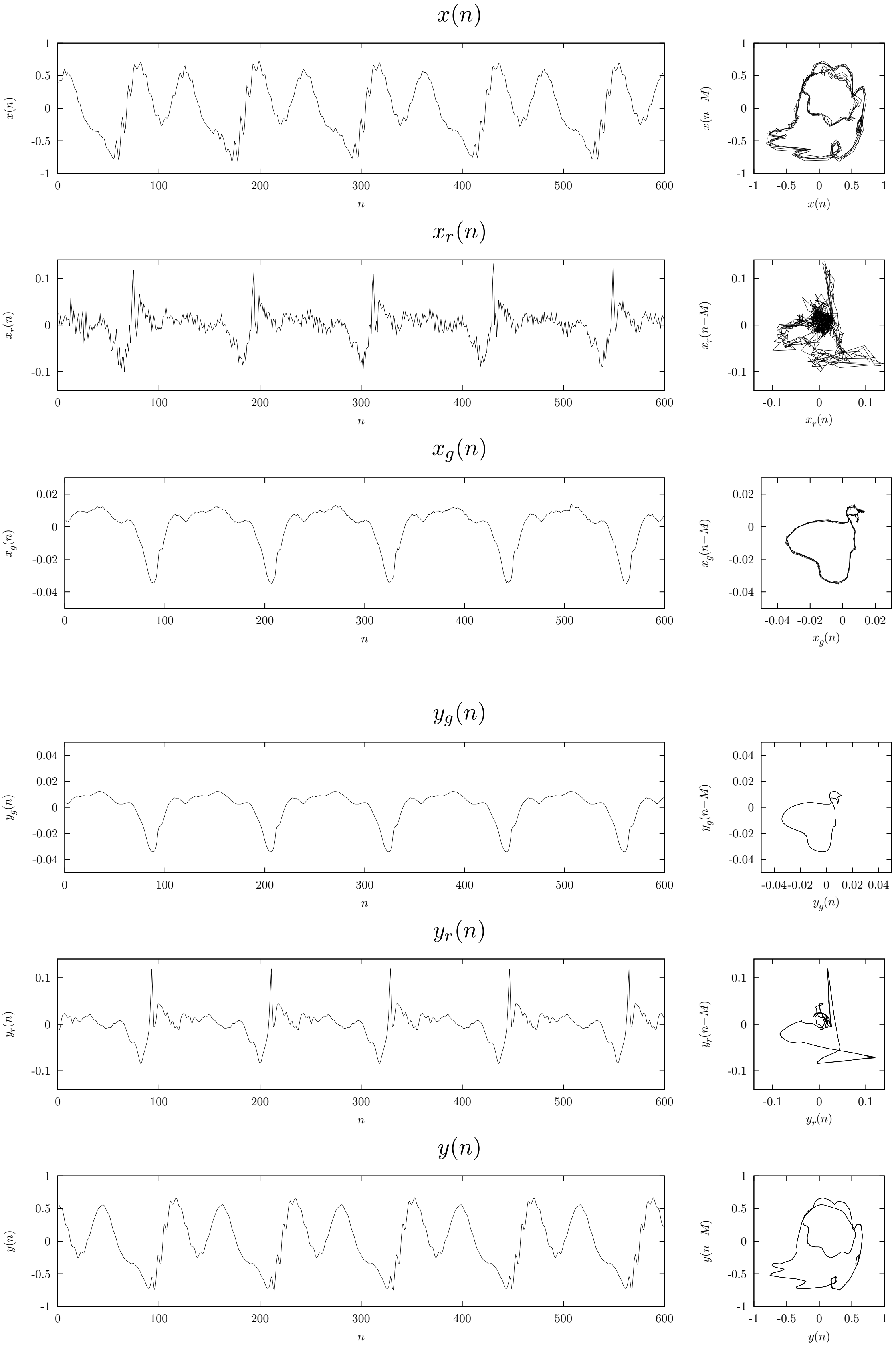


Figure 4.22: Time-signals and phase-space representations of the signals in the combined LP and oscillator model for male vowel /e/.

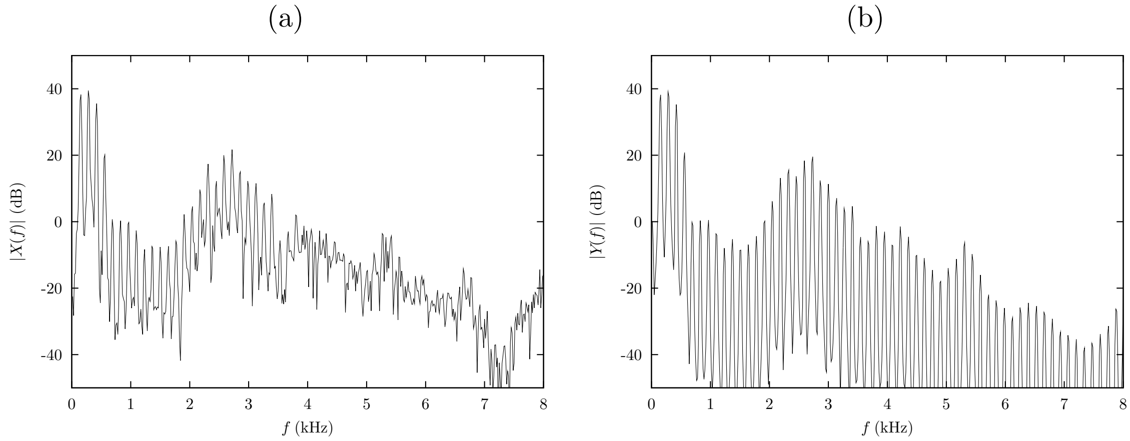


Figure 4.23: Spectra of (a) the original signal $x(n)$, and (b) the oscillator output signal $y(n)$ for synthesis of vowel /e/ with the combined LP and oscillator model (corresponding to the time signals $x(n)$ and $y(n)$ in fig. 4.22).

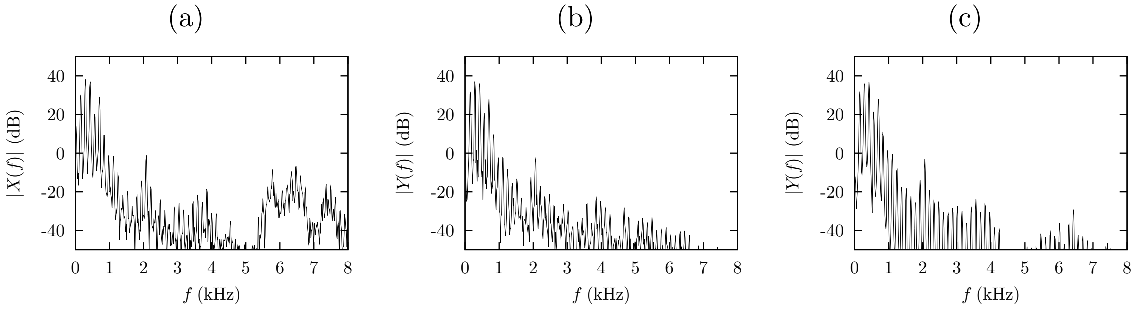


Figure 4.24: Spectra of (a) the original signal (vowel /o/), (b) of the oscillator output signal for full-speech synthesis using minimum regularization to still achieve stable synthesis, $\lambda = 10^{-9}$ (cf. fig. 4.7), and (c) of the output signal from the combined LP and oscillator model using a regularization factor $\lambda = 10^{-5}$.

Thus, we have a fixed complexity of the training using cross-validation comprising 15 matrix inversions, and we find a comparable number of 22.4 iterations on average in the Bayesian training.

Like for the full speech signal (cf. sect. 4.3.2 on page 52) the regularization factor from Bayesian learning is generally higher than the one found by cross-validation. The geometric mean of the quotient is $\mathrm{gmean}(\frac{\lambda_{\mathrm{bay}}}{\lambda_{\mathrm{xv}}}) = 2.14 \times 10^4$, meaning that on average λ_{bay} is four orders of magnitude higher than λ_{xv}, again. Correlation between λ_{xv} and λ_{bay} is lower than for the full speech signals: $c(\ln \lambda_{\mathrm{xv}}, \ln \lambda_{\mathrm{bay}}) = 0.513$.

A number of 28 male vowel signals from the database (of a total of 88 signals) could be successfully re-synthesized using the regularization factor found by cross-validation, and a number of 49 using the regularization factor from Bayesian learning. In contrary to full signal modeling the total number of successfully synthesized signals is reduced by increasing regularization above the value found by Bayesian learning for male vowels. However, also a reduction of the regularization factor below the value found by Bayesian learning reduces the number of stably re-synthesized signals. For female speakers a similar result of 29 successfully re-synthesized signals with regularization according to cross-validation, and 40 with Bayesian regularization (of a total of 72 signals) is found. As opposed to signals from male speakers, for signals from female speakers the number of successfully re-synthesized signals can be further increased when increasing regularization above the Bayesian choice also when modeling the ‘glottal signals’. In table 4.3 the number of successfully re-synthesized signals from male and female speakers are listed for all vowels in the database.

Table 4.3: Number of signals successfully re-synthesized by the combined LP and oscillator model for vowel signals using cross-validation and Bayesian learning, and by increasing and decreasing the regularization factor found by Bayesian learning. For each of 11 male and 9 female speakers one signal of each vowel was used for training, i. e., a total of 88 signals for male and 72 signals for female speakers.

Male speakers

Vowel	/a/	/e/	/i/	/o/	/u/	/ø/	/ɛ/	/ʏ/	total
Cross-validation	2	2	4	6	7	2	2	3	28
Bayesian	2	6	4	8	7	4	8	10	49
$\lambda = 100\lambda_{\text{bay}}$	5	6	4	8	7	5	3	8	46
$\lambda = 0.01\lambda_{\text{bay}}$	2	4	6	5	6	4	5	9	41

Female speakers

Vowel	/a/	/e/	/i/	/o/	/u/	/ø/	/ɛ/	/ʏ/	total
Cross-validation	5	4	2	3	3	3	6	3	29
Bayesian	5	5	2	8	5	6	6	3	40
$\lambda = 100\lambda_{\text{bay}}$	8	9	6	7	5	7	7	4	53
$\lambda = 0.01\lambda_{\text{bay}}$	5	7	2	4	5	6	7	2	38

Looking at two examples for the cross-validation training and validation error as a function of the regularization factor in fig. 4.25 we see that the validation error displays a very flat minimum. For the vowel /e/ the minimum of the validation error marked by a cross can still be identified. For the other vowel /o/, however, the validation error function is within a range of 0.1 dB over 6 decades of the regularization parameter. Thus the actual choice of λ_{xv} within this range may highly depend on the (random) choice of the validation set. This is exemplified in fig. 4.26 where the regularization factor found by cross-validation as a function of the number of training data P for different size of the validation set is plotted. λ_{xv} displays large variations as a function of P and depending on the size of the validation set.

Hence, we find that determining the regularization factor by cross-validation (using one validation set) is not robust with respect to the choice of the validation set and number of training data. This is due to the flat minimum of the validation error as a function of regularization factor we encounter for many vowel signals. The very same problem occurs for the modeling of signals from a chaotic system (the Lorenz system) with a low amount of additive noise [Ran03]. This problem is more critical for the ‘glottal signals’ than for the full speech signals: Since we are dealing with low-pass signals (recall that the ‘glottal signal’ has a nominal spectral fall-off of −12 db per octave) the noise-like components dominating the higher frequencies of a full speech signal are damped. Hence, the validation error for learning the ‘glottal signals’ is flat towards low values of the regularization factor λ, whereas for the full speech signals the higher ‘noise’ level results in an increase of validation error at lower λ values.

Bayesian training, on the other hand, assigns a higher regularization factor than cross-

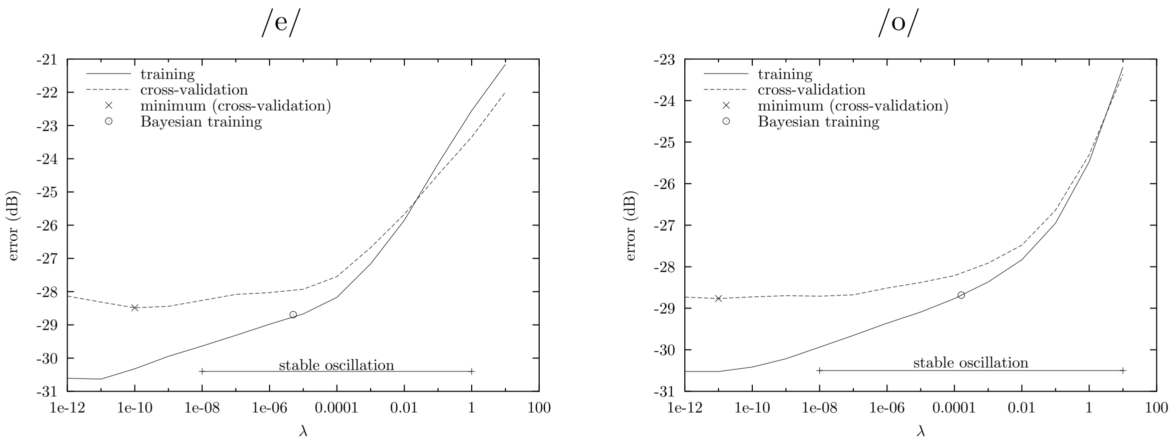


Figure 4.25: Training and cross-validation error as a function of regularization factor λ for the 'glottal signals' of vowels /e/ and /o/. The choice of λ_{xv} at the minimum of the cross-validation error is indicated by a cross. The regularization factor found by Bayesian learning λ_{bay} is indicated by the circle, at an ordinate position corresponding to the Bayesian training error.

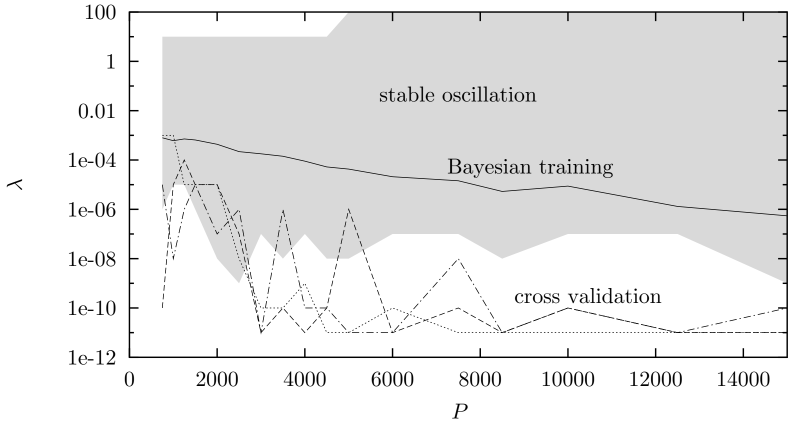


Figure 4.26: Regularization factor λ found by Bayesian learning (solid line) and by cross-validation with one validation set comprising 10% (dashed line), 20% (dotted line), and 30% (dash-dotted line) of the training data for validation as a function of the number of training samples P for the vowel /o/. The range of the regularization factor where stable re-synthesis is possible is indicated by the gray shaded region.

validation, going with a higher prediction error (at least on the training set), but achieves a robust result also in the case of flat cross validation error functions. The choice of λ_{bay}, as read from fig. 4.25, is always at the 'right' end of a flat validation error function, at higher λ values. Robustness of the Bayesian approach is also reflected in the smoothly decreasing value of the regularization factor as a function of the number of training data in fig. 4.26.

4.4.6 Noise variance estimation by Bayesian RBF learning

In sect. 3.1.3 the ability of the Bayesian training algorithm of RBF networks to accurately estimate the noise variance for a one-dimensional training signal has been shown (table 3.1 on page 34). We now ask if the Bayesian algorithm is also able to give reasonable estimates for the power of the noise-like component of speech signals when used for the training of the oscillator model.

As a test for the performance of Bayesian learning for speech signals we use an artificial 'glottal signal' generated with the Liljencrants-Fant model (cf. sect. 2.2 on page 8) and add stationary noise. An LF source signal like in fig. 2.2 on page 10 with a fundamental period of 160 samples is used and white Gaussian noise is added. The signal is embedded in a four-dimensional phase space using embedding lag $M = 13$, RBF centers on a hyper-lattice with $K = 5$ grid lines per dimension of input space, outer grid-lines at $D = 1.2$, $d_{\mathrm{BF}} = d_{\mathrm{diag}}$, and we use a number of $P = 3000$ samples for training.

The values estimated for the noise power by the Bayesian training process are given in table 4.4 for additive white Gaussian noise and for additive low-pass filtered Gaussian noise. For the white noise case the estimated noise power σ_n^2 well matches the noise power σ^2 in the training signal over several orders of magnitude, with a general over-estimation of noise power. Even for the more critical low-pass filtered noise case (the low-pass filtering introduces a low-dimensional structure into the noise signal that can be modeled by the predictor) the estimated noise power follows the actual value well, although the noise power is under-estimated now, in general.

Table 4.4: Parameters estimated by Bayesian learning of a noisy artificial glottal signal generated with the Liljencrants-Fant model. The glottal signal is corrupted by additive white Gaussian noise or by low-pass filtered Gaussian noise (one-pole recursive low-pass, pole at $z = 0.75$) with variance σ^2.

Signal parameters		White Gaussian noise		Low-pass noise	
SNR	σ^2	σ_n^2	$\lambda = \alpha\,\sigma_n^2$	σ_n^2	$\lambda = \alpha\,\sigma_n^2$
40	7.6×10^{-6}	2.20×10^{-5}	4.70×10^{-8}	1.43×10^{-5}	1.77×10^{-8}
30	7.6×10^{-5}	1.49×10^{-4}	1.13×10^{-4}	7.13×10^{-5}	6.15×10^{-5}
20	7.6×10^{-4}	1.22×10^{-3}	8.61×10^{-3}	4.57×10^{-4}	1.06×10^{-2}
15	2.4×10^{-3}	3.62×10^{-3}	4.03×10^{-2}	1.30×10^{-3}	4.85×10^{-2}
10	7.6×10^{-3}	1.10×10^{-2}	3.51×10^{-1}	3.81×10^{-3}	1.84×10^{-1}
5	2.4×10^{-2}	3.34×10^{-2}	1.69	1.15×10^{-2}	5.37×10^{-1}
0	7.6×10^{-2}	1.01×10^{-1}	5.83	3.57×10^{-2}	1.37

Considering that even sustained voiced speech signals are not strictly periodic we further test the Bayesian learning algorithm for LF signals with a randomly varying fundamental period (jitter) and amplitude (shimmer) of individual fundamental cycles. In fig. 4.27 the regularization factor and the estimated noise level (relative to the noise-free training signal) found by Bayesian learning are plotted as a function of the SNR of the training signal for different combinations of jitter and shimmer amplitude up to 5% (i. e., a standard variation of 5% in fundamental period relative to the nominal fundamental period, and standard variation of 5% in maximum amplitude E_e of the LF signal relative to the nominal $E_e = 1$). A natural sustained vowel signal uttered by a healthy speaker would have jitter and shimmer values in the range below 5%, cf. also sect. 5.3.1.

Regardless of the amount of jitter and shimmer, the Bayesian learning algorithm consistently gives a good estimate of the signal SNR and accordingly assigns an appropriate regularization factor. The noise power is slightly over-estimated, but the estimation is quite robust with respect to jitter and shimmer. A very similar result has been found for the modeling of

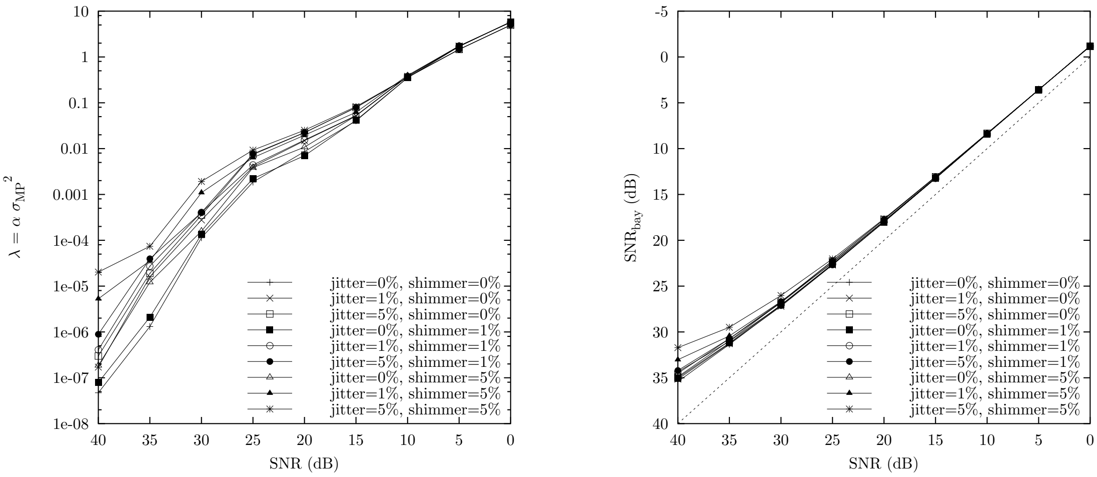


Figure 4.27: Regularization factor λ and SNR value estimated by Bayesian network training for a noisy artificial glottal signal as a function of training data SNR for different amount of jitter and shimmer.

noisy time series from the Lorenz system [Ran03], where the estimate for the noise power is shown to be robust with respect to the number of RBF network centers and an increase of the embedding dimension, too. A comparable result is also obtained for LF signals corrupted by additive low-pass filtered noise, with the estimated $\mathrm{SNR}_{\mathrm{bay}}$ about 5 dB lower then for the white noise case in fig. 4.27, i. e., the noise power is now slightly under-estimated.

In fig. 4.28 noisy training signals with 1% jitter and 1% shimmer and the oscillator generated signals, along with their two-dimensional phase-space representations are depicted. The LF source signal waveform is acceptably re-generated by the oscillator for SNR down to 10 dB. For lower SNR the oscillator output signal attains a constant value. The robust identification of an almost periodic waveform in moderate noise, but also the constant output signal for too high a noise level is very convenient for modeling mixed excitation and purely unvoiced speech signals, see Chapter 5.

A similarly precise estimation of the noise level as for stationary LF signals is also found for noisy LF training signals with linearly increasing amplitude from 0.2 to 1.0 over $P = 3000$ training samples. Results found for the regularization factor λ and for $\mathrm{SNR}_{\mathrm{bay}}$ are comparable to the numbers in fig. 4.27. Again, the noise level is slightly over-estimated. Re-synthesis of the linearly amplitude modulated LF signal, however, fails even for high SNR. Instead of a signal with increasing amplitude only a periodic signal (with intermediate amplitude) is generated in some cases. This problem could not be relieved by using multiple examples of amplitude modulated LF signals for the training. This points out the necessity to use *stationary speech signals* for the training of the oscillator model.

4.5 Non-stationary modeling

So far, all the speech signals used for training and re-synthesized by the oscillator model were (considered) stationary. For real speech synthesis tasks, it is of course necessary to account for the non-stationarity of the speech signal. In this section we will present attempts to generate signals with varying fundamental frequency and spectral content.

The experiments described in this section shall, to some extent, justify the choice of fixed parameters of the oscillator model, like embedding delay M and dimension N, and the nonlinear function model (RBF centers) for all speech sounds. Only a fixed choice of these model

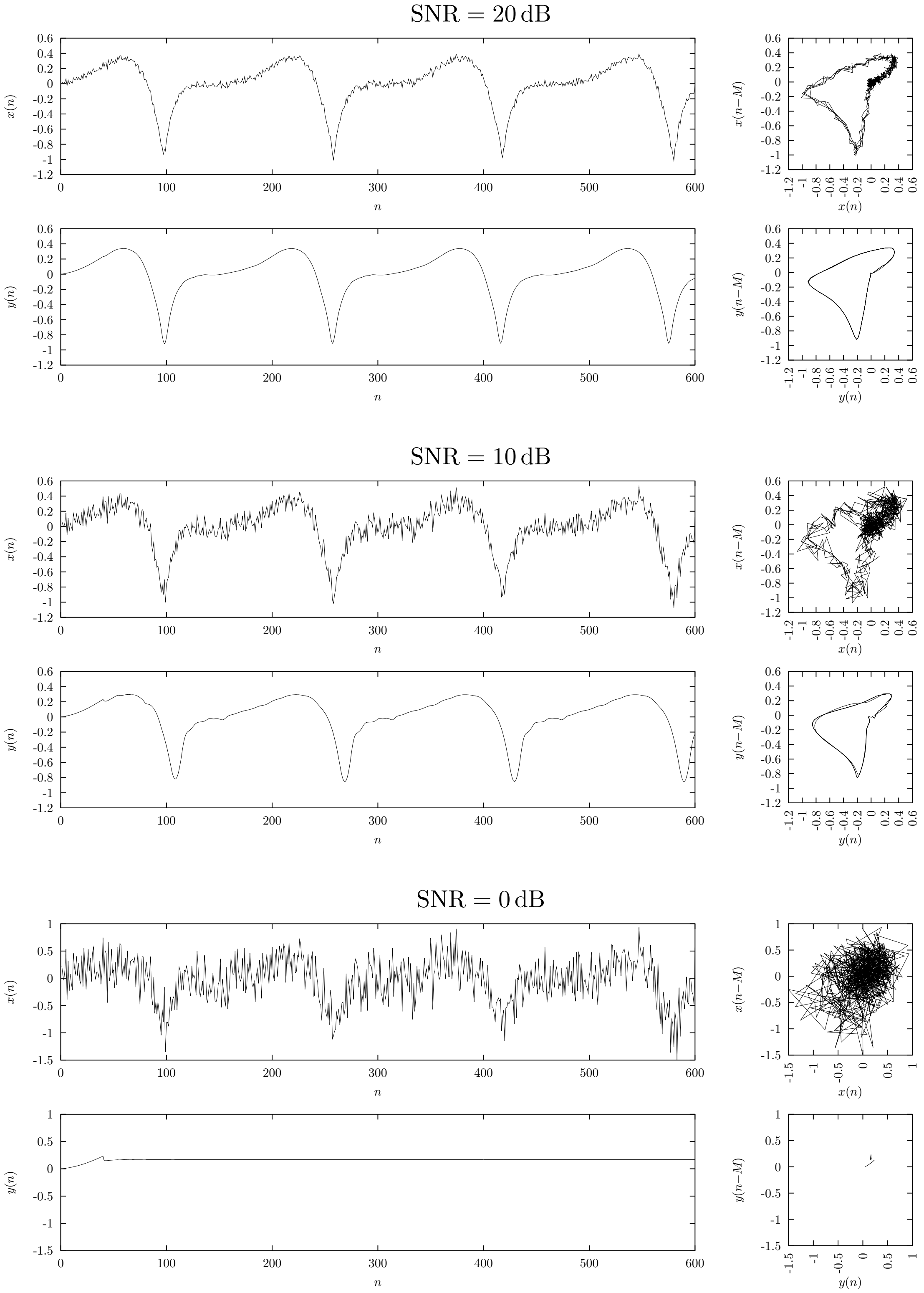


Figure 4.28: Training signals $x(n)$ and re-synthesized signals $y(n)$ and the two-dimensional projection of their phase-space representations for LF glottal signals with 1% jitter and 1% shimmer and different amount of additive white noise.

parameters makes it possible to generate transitions between models learned from different sounds, since for that aim we have to »*follow nature, which has just* one *glottis, and just* one *mouth, emitting all sounds, which only for this reason combine*«:

> Itzt fieng ich an einzuſehen, daß ſich die einzelnen Buchſtaben zwar erfinden, aber auf die Art, wie ich es angriff, nimmermehr in Sylben zuſammenbinden ließen, und daß ich ſchlechterdings der Natur folgen müßte, die nur eine Stimmritze, und nur einen Mund hat, zu dem alle Laute herausgehen, und eben nur darum ſich miteinander verbinden.

Wolfgang van Kempelen [vK70, p. 407].

4.5.1 Interpolation of fundamental frequency

The possibility of interpolation between models trained on speech signals with differing fundamental frequency has been shown in [Man99, sect. 7.4] for the full speech signal by linear interpolation of the RBF network weights for fixed centers and basis functions. Stable oscillator output signals with fundamental frequency F_0 in the range of the F_0-values of two training signals could be generated by linear interpolation of the RBF network weights according to

$$\boldsymbol{w}_{\text{int}} = (1 - k_{\text{int}})\boldsymbol{w}_1 + k_{\text{int}}\boldsymbol{w}_2 \quad , \tag{4.10}$$

with $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ being the RBF weights determined for the two signals, and $k_{\text{int}} \in [0, 1]$ the interpolation factor.

Our investigations show that using weights from models with a regularization factor determined by cross-validation or Bayesian training did not yield a stable oscillator behavior for interpolated weight values in general. However, if regularization is increased – i. e., the variation of the weight values over phase space is reduced – stable synthesis by interpolated models is possible. For instance, for models trained on two realizations of the vowel /a/ with $F_0 = 135\,\text{Hz}$ and $F_0 = 180\,\text{Hz}$, respectively, and using a regularization factor $\lambda = 10^{-1}$ for RBF weight learning, the synthetic glottal signals in fig. 4.29 were generated. Model parameters were $N = 4$, $M = 13$, $k_{\text{em}} = 0.75$, $k_{\text{lp}} = 0.95$, $K = 5$, $P = 3000$, $d_{\text{BF}} = 1$, $d_{\text{prun}} = 1$ (203 and 184 of 625 centers were pruned during training for model one and two, respectively, and the according weights in $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ were set to zero value).

From fig. 4.29 it can be seen that according to the interpolation factor k_{int} a smooth transition of the fundamental period waveform and the signal representation in phase space between the signals generated using the weights for the first ($k_{\text{int}} = 0$, top bar) and the second ($k_{\text{int}} = 1$, bottom bar) training signal is achieved. However, the fundamental frequency of the output signal from the interpolated models switches to F_0 of the second training signal already for $k_{\text{int}} = 0.25$. Thus, a reliable control of F_0 by linear interpolation of weights does not seem possible. Moreover, weight interpolation leads to a smoothing of the glottis closure impulses (negative peaks), resulting in a steeper spectral fall-off of the generated signals. This can be clearly seen in the spectra of the generated full speech signals in fig. 4.30. Similarly to the weights, the LP coefficients were interpolated linearly in the log-area-ratio domain (see [Ran99]). Although LP synthesis establishes the spectral envelope in the low-frequency domain – and the interpolated signals are perceptually clearly recognizable as vowel /a/ – the signals generated using interpolated weight values lack high-frequency components present in the synthetic signals using the original weights from the training signals.

It has to be noted, too, that for training signals with a larger difference in fundamental frequency or fundamental waveform shape linear interpolation of the weights often results in

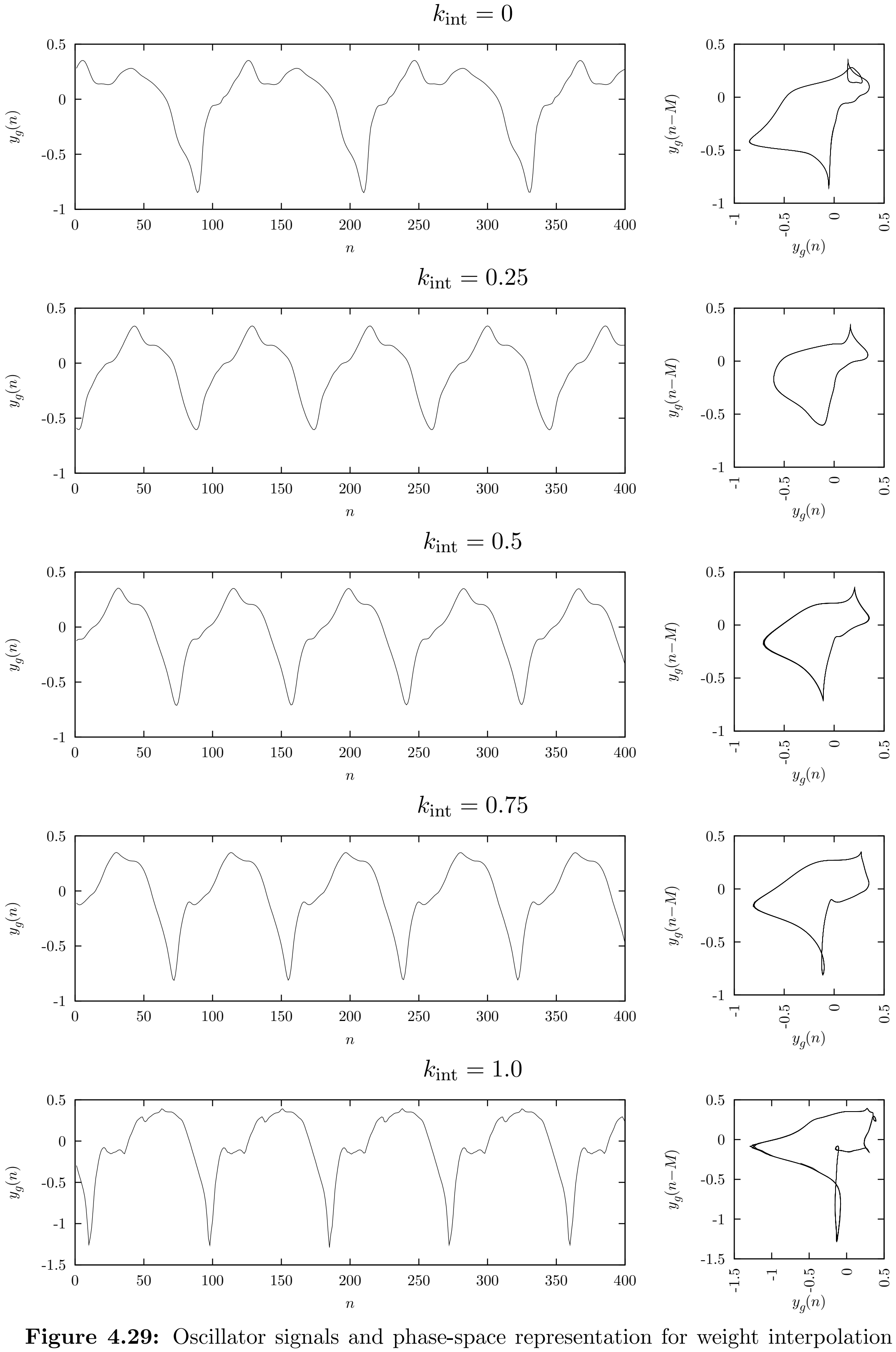
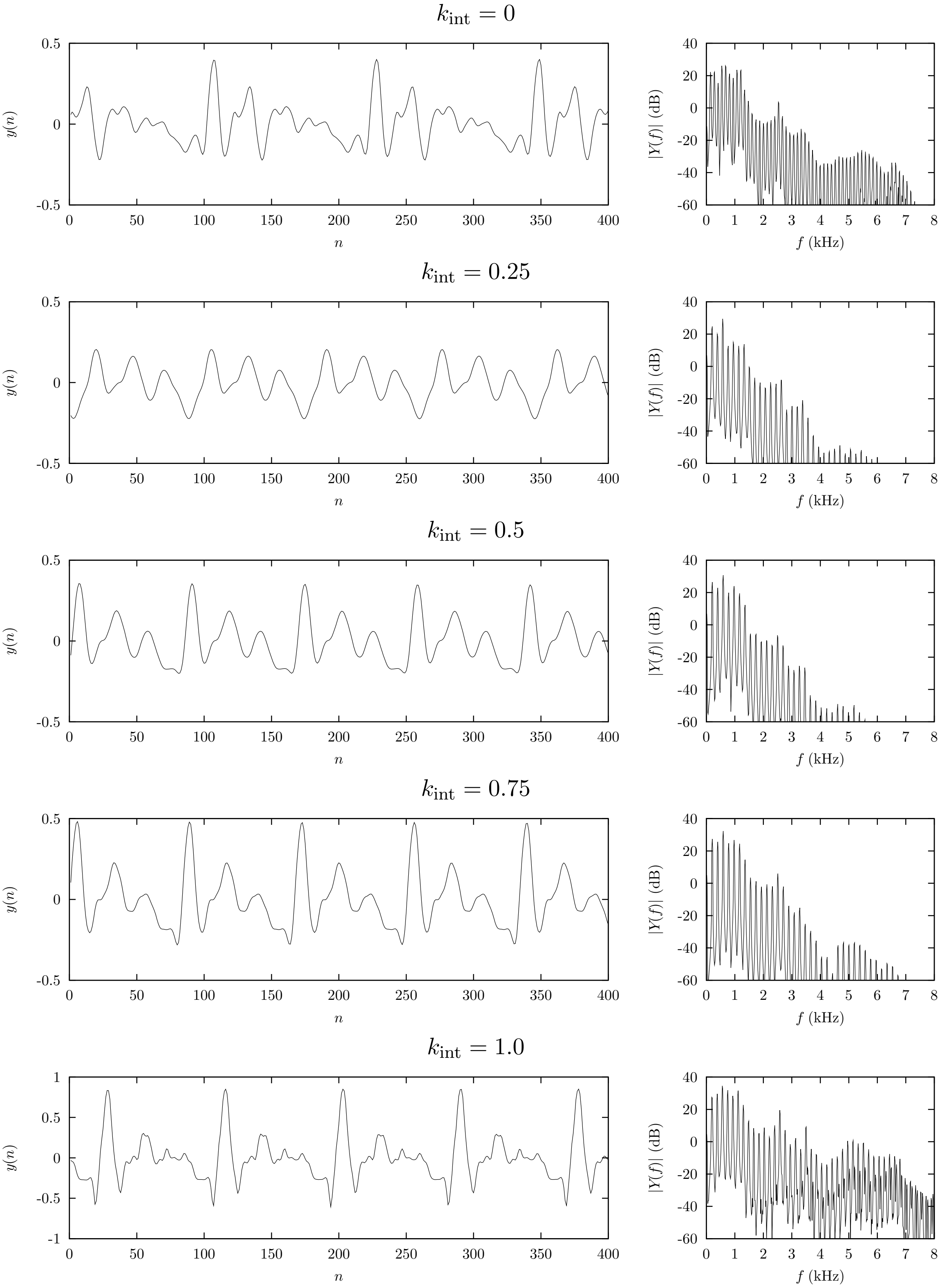


Figure 4.29: Oscillator signals and phase-space representation for weight interpolation in the glottal signal domain between two models for vowel /a/ with different F_0.



$k_{\text{int}} = 0$

$k_{\text{int}} = 0.25$

$k_{\text{int}} = 0.5$

$k_{\text{int}} = 0.75$

$k_{\text{int}} = 1.0$

Figure 4.30: Full speech signals and spectra for interpolation between two signals with different F_0 for vowel /a/.

unstable oscillator behavior. Weight interpolation thus should be used with caution, and e. g., for signals from the same speaker with not too large differences in F_0, amplitude, and waveform shape or phase space structure only.

4.5.2 Morphing of vowel sounds

Considering that the signals found by LP inverse filtering and successive low-pass filtering of vowel signals display a more similar structure in phase space (fig. 4.21 on page 73) as opposed to the full speech signals for one specific speaker (fig. 4.9 on page 57), we also can try to interpolate models for different vowel sounds. The idea of this "morphing" in the phase-space domain has been presented in [BM96] for an elaborate nearest neighbor model (cf. sect. 3.2.1). We test morphing by interpolation of RBF network weights.

Two models were trained on vowel sounds from one speaker. We use a regularization factor $\lambda = 10^{-2}$ for both models, which is again higher than the ones found by cross-validation ($\lambda_{\text{xv}} \approx 10^{-6}$) or Bayesian training ($\lambda_{\text{bay}} \approx 10^{-4}$). When training the oscillator model on the full speech signals for the vowels /o/ and /i/, linear interpolation of the weights (eq. 4.10) using a time-variant interpolation factor k_{int} varying from $k_{\text{int}} = 0$ to $k_{\text{int}} = 1$ over a time-range of 6000 samples yields the synthetic signal depicted in fig. 4.31. It comes with no surprise that due to the significantly different phase-space structure of the full speech signals of the two vowels the interpolated model fails to generate an adequate transition between the vowels.

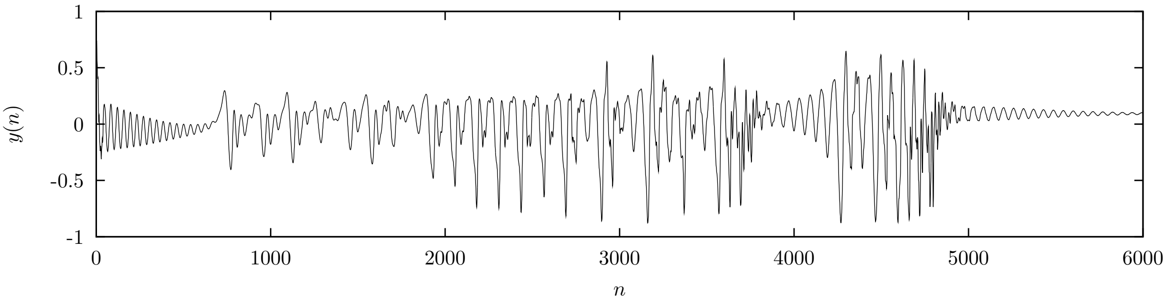


Figure 4.31: Interpolation between models for the full speech signal of the vowels /o/ and /i/.

If, however, the network weights from the combined LP and oscillator model are interpolated – i. e., the transition is generated in the glottal signal domain – a signal as depicted in fig. 4.32 is generated. Since the original 'glottal signals' are similar it is possible to generate a stable transition. For the re-generation of the full speech signal also the LP synthesis filter has to be interpolated. Again, we use linear interpolation of the filter coefficients in the log-area-ratio domain. And, since we are dealing with a non-stationary filter now, we utilize not the direct form filter implementation, but a normalized lattice filter structure [MG76], which displays robust behavior when switching coefficients [Ran00]. LP filter coefficients were interpolated every 16 samples (corresponding to 1 ms).

The thus re-generated full speech signal is depicted in fig. 4.33 along with its spectrogram and F_0 trajectory. As can be seen the interpolation of the combined LP and oscillator model yields a stable signal with nearly constant signal amplitude, smooth transitions of formants and almost constant F_0 trajectory. Also perceptually the signal can be clearly recognized as a 'diphthong', without any notable distortion.

Again, some caution is necessary here: We find many other examples where the generation of a stable transition between different vowels fails. A number of possible reasons can be

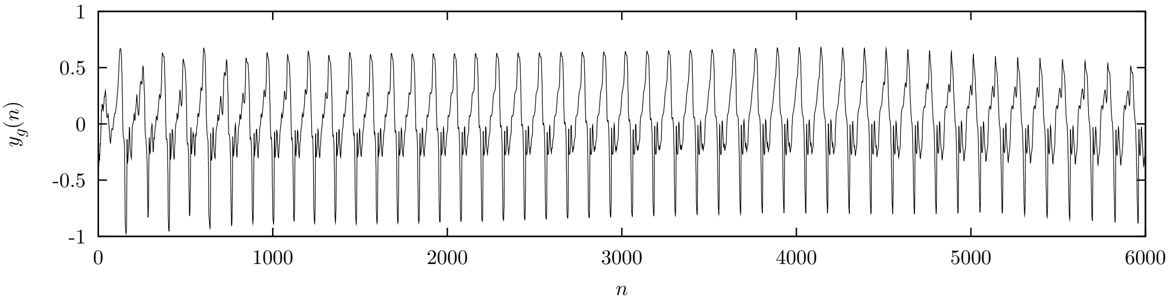


Figure 4.32: Interpolation between models for the glottal signal of the vowels /o/ and /i/.

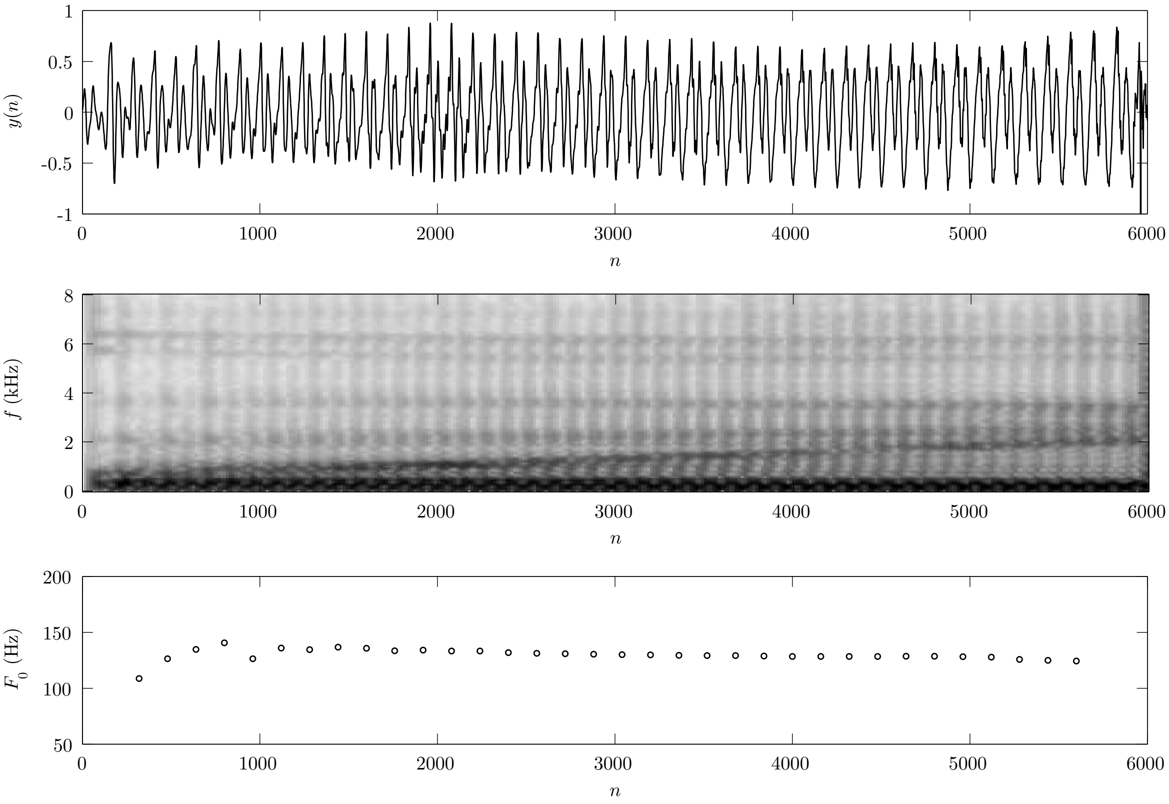


Figure 4.33: Full speech signal, spectrogram, and F_0 trajectory from interpolation between models of the vowels /o/ and /i/ using the combined LP and oscillator model.

identified: First, the training signals are normalized for oscillator training, and thus amplitude mismatches in phase space may occur for different vowels, leading to an incorrect relation between the signals trajectories. However, normalization is necessary since the LP analysis filters have different gain for different vowels. Second, the 'glottal signals' found by inverse filtering still contain a lot of features not attributable to the glottal pressure signal only. E. g., information about the speaker, speaking style, and prosodic parameters have an influence on the 'glottal signal' (see, e. g., [SV01]). Also, the 'glottal signal' found by inverse filtering still contains the all-pass part of the full speech signal which cannot be removed by minimum-phase LP analysis filtering (cf. sect. 4.4.2). At last, in the combined LP and oscillator model the mutual influence between vocal tract and glottal movement is neglected, since there is no feedback path from the LP synthesis filter to the oscillator.

4.6 Conclusion

In this chapter we investigated the performance of the oscillator model for stationary vowel signals when modeling the full speech signal and when modeling the 'glottal signal' estimated by linear prediction and subsequent low-pass filtering. We compared different realizations of the nonlinear function in the oscillator model, and particularly two ways of determining the regularization factor for Gaussian RBF networks: (a) cross-validation and (b) Bayesian learning. Finally we explored the feasibility of interpolation between models trained on stationary speech signals to generate signals with intermediate fundamental frequency or transitions between different sounds, if an RBF network is used as nonlinear function realization.

By the analysis of false neighbors we find that modeling the dynamics of sustained vowel signals is possible in a low-dimensional phase space. An embedding dimension of $N = 4$ should be sufficient. Mutual information between delayed samples as a function of delay reveals different optimal embedding delays for individual vowel signals, however, using a fixed compromise embedding delay for all signals is possible. A suitable compromise value for the embedding delay is $M = 13$ (for a signal sampling rate of 16 kHz), which resulted in the largest number of successfully re-synthesized signals from male speakers in our database. For signals from female speakers the same embedding delay is used, too, although the optimal embedding delay might be somewhat smaller. The search for a better nonlinear predictor using a non-equidistant embedding only results in a minor increase in prediction gain, and the embeddings with the highest prediction gain often implicate low embedding delays – which prevents the unfolding of the signal trajectory in phase space, thus impairing oscillator stability.

Using an RBF network as the realization of the nonlinear function in the oscillator model, stable synthesis can only be achieved by the application of regularization during network training. The additional exclusion of all network centers with a certain minimum distance from the training signal trajectory from the training process reduces large amplitude peaks in the oscillator output signal, and by varying the regularization factor the amount of high-frequency components can be altered to a small extent. Here, regularizing matrix inversion during RBF training and regularized GRBF training are comparable in terms of prediction error, with the GRBF yielding a worse conditioning on the matrix that has to be inverted.

Determining the regularization factor by cross-validation yields a generally lower amount of regularization – and a lower number of successfully re-synthesized vowel signals – as compared to the application of the Bayesian training algorithm, whereas both methods have approximately the same computational complexity.

A brief comparison of the other nonlinear function models introduced in Chapter 3 reveals that the RVM considerably reduces the number of RBF network centers and achieves almost the same performance in terms of prediction gain and number of stably re-synthesized vowel signals as the Bayesian trained RBF network. Using MLPs as nonlinear function models we are also able to achieve stable re-synthesis, even with a network of much lower complexity than the RBF networks used (in terms of the number of adjustable network parameters). However, in our experiments we find that stability of the MLP based oscillator highly depends on the random initialization of the network weights. The locally constant or nearest neighbor model always achieves stable re-synthesis and naturally incorporates the noise-like component of a speech signal that can not be re-generated by the other models. The MARS model commonly yields unstable oscillator behavior with the output tending to infinity, and stable re-synthesis could only be achieved in a six-dimensional embedding space for specific signals. Moreover, all these models lack the direct relation between their adjustable parameters for different training signals inherent to RBF networks with fixed basis functions and fixed center positions.

A common characteristic of stably re-synthesized full speech signals is a lack of high-frequency components. Furthermore it was found that signals displaying a complex trajectory structure in phase space – like, e. g., the vowels /a/, /e/ and /i/ – can less often be stably re-synthesized than, e. g., the vowels /o/ and /u/, which display a less complex trajectory. Both facts motivate the combination of the oscillator model with inverse filtering by linear prediction, thus shifting the nonlinear modeling task to the 'glottal signal' and attaining spectral shaping by linear prediction filtering. Linear prediction and subsequent low-pass filtering of the full speech signals yields 'glottal signals' that show an 'open loop' trajectory structure.

As compared to the oscillator applied to model full speech signals, a significantly higher number of vowel signals could be stably re-synthesized using the combined LP and oscillator model with an RBF based nonlinear function realization, and a more faithful reconstruction of the signal spectra, also in the higher frequency range of wide-band speech signals is observed.

Again, the Bayesian algorithm for RBF network training and determination of the regularization factor outperforms cross-validation in terms of the number of stably re-synthesized vowel signals. The Bayesian algorithm is found particularly useful for training signals with a low amount of noise (or noise-like components, as voiced speech signals) resulting in a flat cross-validation error function.

As for the simple regression task (sect. 3.1.3) the Bayesian algorithm used for function identification in the oscillator model gives a robust and accurate estimate for the variance of a noise signal added to the LF speech source signal. Despite perturbations like jitter and shimmer, the variance of an additive white or low-pass filtered Gaussian noise signal is identified well. Re-synthesis by the oscillator model yields signals close to the original LF source signal for SNR values down to 10 dB. For lower SNR the identification of the original signal from a limited number of training samples is not possible, and the oscillator model generates a constant output signal. Also for LF signals with linear amplitude modulation re-synthesis is not possible, pointing to the need for stationary speech signals as training data for the oscillator model.

Concerning the notion of interpolation between RBF network weights for fundamental frequency control considered in [Man99], we verify this possibility also in the 'glottal signal' domain, however, the robust control of fundamental frequency still seems a big challenge and signals generated from models with their weights interpolated again lack high-frequency components due to a smoothing of the glottis closure impulses by interpolation. However, by the very interpolation approach smooth and stable transitions between two vowels could be obtained using linear interpolation between the two models, which is a step towards general purpose synthesis with the oscillator model by connecting models for different speech sounds.

Concluding, it can be said that stable re-synthesis of stationary vowel signals by the oscillator model based on a training of the nonlinear function in the oscillator without manual supervision should be preferably based on an RBF network using the Bayesian algorithm for determining the regularization factor, and should incorporate inverse filtering of the speech signal. Linear prediction as a straightforward approach to inverse filtering, used in many speech transmission and synthesis algorithms, is a viable complement to nonlinear signal modeling, significantly increasing the number of signals that can be successfully modeled by the oscillator and enhancing the spectral reproduction. Nevertheless, perceptual inspection of re-generated signals still reveals a missing component of natural speech signals. The means to re-synthesize this missing signal part will be investigated in the following chapter.

Chapter 5

Oscillator-plus-noise model

For signal prediction or system modeling, an unpredictable component in the original signal is usually considered an additive noise signal and/or measurement noise. This noise is seen as a *nuisance signal* and makes the signal prediction or system identification difficult, and is generally not to be captured by a model. Applying the oscillator model to speech signals we implicitly distinguish between a signal component that is a low-dimensional oscillatory signal component (produced by the vocal folds vibration) and a noise-like signal component that stems from high-dimensional processes (e. g., turbulent air flow). If we model the signal production system by a low-dimensional oscillator and a smooth nonlinear function, only the low-dimensional signal component will be captured[1].

Speech signals, however – even voiced signals from vowels – generally comprise a noise-like component due to high-dimensional turbulent air flow that is not predictable by a low-dimensional system. And, if we aim at modeling the entire speech signal, the signal component related to noise-like sources has to be considered as a relevant part of the *signal* itself rather than as *nuisance*.

Based on the observation that the noise-like signal components dominate the speech spectrum at higher frequencies – for example in the early harmonic-plus-noise model [SLM95] only noise like components are considered above a 'maximum voiced frequency' – we can also explain the yet unsatisfying spectral reproduction of vowels by the oscillator model. High frequency components due to noise-like excitation are not re-synthesized because they are just not captured by the low-dimensional oscillator model. And we cannot hope to be able to model those high-dimensional signal components using a higher number of parameters for our nonlinear function model or a (reasonably) higher embedding space dimension!

Looking at other speech synthesis models, the sources for oscillatory and noise-like signals are in most cases distinct systems. In van Kempelen's speaking machine, the oscillatory source has been realized by an ivory reed (similar to the reed in single-reed woodwind instruments), whereas the noise source has been realized as a constriction resulting in turbulent air flow [vK70, pp. 410–426]. Also in formant speech synthesis the (periodic) oscillatory source and the noise source are strictly distinct, e. g., a pulse train and a (white) random signal in the simple model of fig. 2.1 on page 9. State-of-the-art sinusoidal models like the harmonic-plus-noise model explicitly distinguish between modeling the oscillatory signal component by a sum of harmonically related sinusoids and the noise-like component by an amplitude modulated auto-regressive process[2] [Sty01, Bai02b].

In this chapter we develop an extension of the oscillator model for speech signals that includes a reproduction system for the noise-like signal component. Relating to the harmonic-

[1] The realization of the nonlinear function by the locally constant model described in sect. 3.2.1 on page 36 is an example of a non-smooth nonlinear function model that entails noise modeling.

[2] I. e., by the amplitude modulated output of a time-varying white noise excited LP synthesis filter.

plus-noise model we baptize this system *oscillator-plus-noise model*.

We will first see how the noise-like component of a speech signal can be assessed, either directly from the signal, or by examining the modeling error of speech synthesis and coding algorithms, and have a look at the necessities that have to be met for natural re-synthesis of the noise-like component of a speech signal.

Assuming that the oscillator model is able to correctly identify the low-dimensional oscillatory component of a speech signal, we will consider the modeling error for the prediction of the original signal as related to the noise-like excitation. We will show that vowel signals can be well re-generated – including noise-like excitation due to turbulent air flow at the vocal folds – by augmenting the combined LP and oscillator model from sect. 4.4.3 by a nonlinear predictor, used to re-generate a modulated additive noise signal component.

We then will consider the more difficult case of voiced fricatives, extending the model with a second LP path for the spectral shaping of the noise-like signal component independent from the spectral characteristic of the oscillatory component, and show that with this extension general mixed-excitation speech sounds can be re-synthesized satisfactorily, and that also unvoiced (purely noise-like) sounds can be reproduced by the same system. Finally, we will show that the oscillator-plus-noise model is superior to the basic oscillator model in terms of reproduction of short-term variations in the signal and trajectory properties in embedding phase space, suggesting higher naturalness of the generated synthetic speech signals.

5.1 Noise-like component in speech signals

Decomposing a speech signal into the oscillatory and noise-like component is a quite difficult task. Not only is the determination of the intrinsic signal-to-noise ratio (SNR) generally limited in accuracy [Zar02], but natural speech signals pose a number of other problems on analysis. Two among them are:

- The oscillatory component is not strictly periodic – it displays short-term variations in waveform shape, fundamental frequency (jitter), and amplitude (shimmer), even for intentionally sustained voiced phonemes.

- The noise-like component is not stationary white noise, but spectrally shaped and modulated in amplitude by the oscillatory signal component.

Nevertheless, a number of methods have been proposed for assessing the relation between oscillatory and noise-like signal components (e. g., [dK94, MGS95, Sty96, JS01, TBW$^+$03]), and we will introduce some of them in the following.

Besides objective measures, the noise-like component can also be related to the modeling error of speech coding and synthesis algorithms for the oscillatory component, as it is commonly pursued in harmonic-plus-noise synthesis algorithms. Generally, some basic requirements for natural reproduction of the noise-like component have to be considered.

5.1.1 Objective measures

One measure for the *harmonics-to-noise ratio (HNR)* proposed in [dK94] is based on an estimation of the harmonic component in the cepstral domain. The peaks in the cepstrum at lags of the fundamental period and its multiples are considered to be evoked by the harmonic signal component. Cepstral components around these peaks are classified as evoked by the harmonic signal components, and the corresponding components in the spectral domain are subtracted from the original spectrum. The so computed spectrum of the noise-like signal component is aligned appropriately below the original spectrum and the HNR is computed as

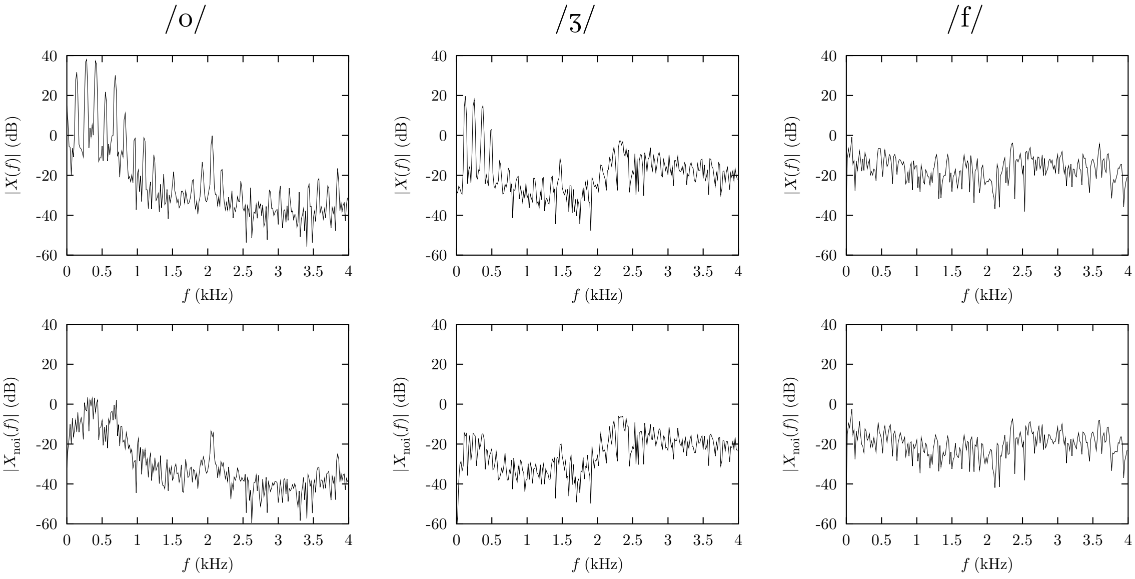


Figure 5.1: DFT magnitude spectra of the original signal (top row) and the signal component classified as noise-like (bottom row) by the HNR estimation procedure [dK94] for a vowel /o/, a voiced fricative /ʒ/, and an unvoiced fricative /f/. The corresponding values found for the HNR measure are 23.1 dB, 7.8 dB, and 2.8 dB, respectively.

the total spectral energy of the original signal in relation to the energy of the noise-like signal component's spectrum.

Examples for the amplitude spectra of the original signals and the signal component classified as noise-like for a vowel, a voiced fricative, and an unvoiced fricative are depicted in fig. 5.1.

Another measure is the *glottal-to-noise excitation (GNE) ratio* [MGS95]. The GNE ratio measure is based on the correlation between the Hilbert envelopes of the linear prediction residual signal in different non-overlapping frequency bands. For a signal evoked by glottal oscillation the glottis closure impulse triggers an impulse of the Hilbert envelope in all frequency bands. Hence, a strong correlation between the envelopes in all frequency bands is observed. In contrast, a signal with noise excitation will not result in correlated Hilbert envelopes in distinct frequency bands. Thus, the correlation between the Hilbert envelopes is high if the speech signal stems primarily from glottal excitation, whereas for a noise excited signal the correlation between the Hilbert envelopes in the non-overlapping frequency bands is lower. Therefore, the value of the GNE ratio provides a measure for glottis evoked vs. noise evoked signal components. The GNE ratio is to a high degree immune to variations in fundamental frequency (jitter) and amplitude (shimmer) of individual fundamental periods [MGS95].

The GNE ratio estimation is performed by applying conventional linear prediction analysis and inverse filtering to the speech signal[3]. A fast Fourier transform (FFT) is applied and the Hilbert envelopes are calculated by performing the inverse FFT on a number of non-overlapping frequency bands using only FFT points corresponding to positive frequencies. For each combination of different frequency bands the maxima of cross correlation between the envelopes are computed (taking into regard also some small time lags) and the largest maximum of the correlation values represents the GNE ratio.

A quite simple distinction between voiced, mixed excitation, and unvoiced speech signals can be obtained by a measure for *spectral flatness* of the short-term DFT spectrum. Spectral

[3] Without pre-emphasis, and without the low-pass filtering described in sect. 4.4.

flatness is the ratio of the geometric and the arithmetic mean of the spectral energy distribution [MG76]. As such, its value is limited to a range between zero and one (or $-\infty$ and 0 dB, respectively), and equal to one (0 dB) only for a perfectly flat spectrum. Since a periodic signal has a line spectrum whereas a noise signal has a rather flat energy spectrum, a periodic or almost periodic voiced speech signal in general displays a more variable spectrum than a noise-like unvoiced speech signal. A higher spectral flatness measure will thus be found for speech with higher amount of noise-like excitation.

In table 5.1 values computed for the above measures are stated for some examples of voiced, mixed excitation, and unvoiced phonemes spoken by male speakers. We can see that a differentiation in the measures between the different excitation categories is apparent.

Table 5.1: HNR, GNE, and spectral flatness (SF) measures for a number of sustained phonemes with voiced, mixed, and unvoiced excitation.

Excitation	Voiced			Mixed			Unvoiced	
Phoneme	/aː/	/ɛ/	/oː/	/v/	/ʒ/	/z/	/s/	/f/
HNR (dB)	17.8	22.2	21.9	9.8	8.1	5.4	2.1	2.6
GNE	0.92	0.96	0.96	0.75	0.92	0.91	0.74	0.78
SF (dB)	−24.1	−22.6	−22.1	−18.6	−15.8	−11.2	−9.6	−5.0

The stochastic nature of the noise-like signal component also results in a quicker fall-off of the mutual information (MI) between delayed signal samples as a function of delay L for mixed excitation and unvoiced speech signals compared to voiced speech signals. In fig. 5.2 the average MI between signal samples as a function of delay is depicted for three speech signals from male speakers. For the vowel signal the mutual information displays a first minimum (cf. fig. 4.3) around a delay of 10 samples (0.6 ms) and higher values for increasing delay, whereas for the mixed-excitation fricative a vague first minimum occurs at a delay of 4 samples (0.25 ms) and the function further decays for increasing delay. For the unvoiced fricative the average MI quickly decays to zero. Besides the possibility to distinguish between excitation categories based on the shape of MI as a function of delay L, these figures indicate that mixed excitation and purely unvoiced sounds carry an unpredictable (noise-like) component that cannot be modeled by an oscillator.

The above measures seem to give a quite reliable distinction between the different modes of excitation for a speech signal. It has to be noted, however, that these measures can also vary significantly with other attributes of speech production, e. g., with accentuation or emotional state of the speaker [ARK+99, ARK+00, ARK+03]. Moreover, only the HNR gives a *quantitative* measure of the amount of noise-like excitation present in the speech signal. Another way to assess the unvoiced signal component is to examine the modeling error of speech synthesis and coding algorithms for the oscillatory component of voiced speech signals.

5.1.2 Error signal in models for voiced speech

An implicit distinction between the oscillatory (or to be precise, the harmonic) and the noise-like component of a speech signal is found in sinusoidal models for speech synthesis (cf. sect. 2.4.3 on page 15). Particularly, in sinusoidal modeling of harmonically related signal components only, i. e., in harmonic-plus-noise models (e. g., [Bai02b]), the residual or error signal can be considered as the noise-like component of the speech signal [Sty96]. In table 5.2

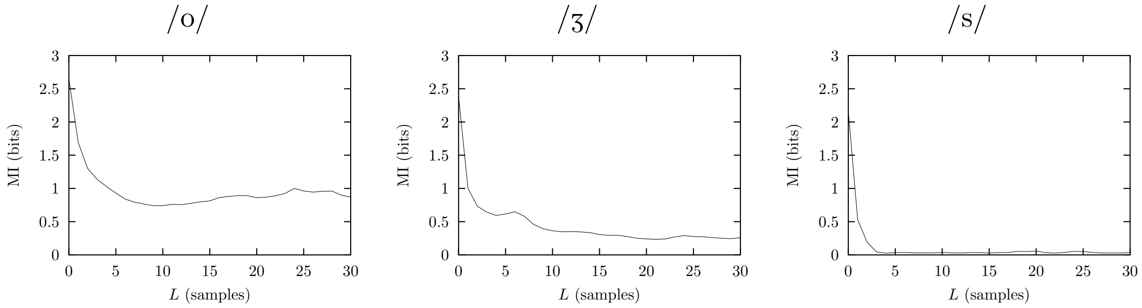


Figure 5.2: Mutual information (MI) between signal samples as a function of delay L for a vowel /o/, a voiced fricative /ʒ/, and an unvoiced fricative /s/ (male speaker). Mutual information is calculated by the function `mutual` in the nonlinear time-series analysis package TISEAN [HKS99].

the corresponding ratio between the energy of the speech signal component that is modeled by a sum of harmonically related sinusoids and the residual energy is given. Stated are mean values for some vowels and voiced fricatives. We can observe that the values correlate well with the values of the HNR measure listed in table 5.1.

Table 5.2: HNR found for some vowels and voiced fricatives in the analysis-by-synthesis process of a pitch-synchronous harmonic-plus-noise model (after [Bai02b, table 3.1]).

Excitation	Voiced				Mixed		
Phoneme	/a/	/i/	/u/	/y/	/v/	/ʒ/	/z/
HNR (dB)	24.04	26.22	24.13	21.52	11.96	4.22	3.26

Waveform interpolation for low-rate speech coding is another example for model based distinction between oscillatory and noise-like signal components. Here, a decomposition of the speech signal into a slowly varying periodic (oscillatory) component and a rapidly varying (noise-like) component is performed [MK93, KH94, MB95, KH95a, KH95b]. Coding gain is achieved by transmitting the slowly varying component only once for several fundamental periods – i. e., by using one *prototype waveform* for several fundamental periods – and due to the noise-like nature of the rapidly varying component, which allows transmission of only its magnitude spectrum, disregarding the phase, since the human auditory system does not regard for phase information of unvoiced speech signals[4]. Relating the energy of the slowly varying component of a speech signal to the energy of the rapidly varying component yields another HNR measure.

5.1.3 Requirements for natural re-synthesis

For natural re-synthesis of the noise-like speech signal component some requirements have to be met. Obviously, the noise-like signal has to be spectrally shaped. In formant synthesis (sect. 2.2) the white noise source signal is fed through the vocal tract filter, resulting in formants at filter resonance frequencies (it is possible to use a lower number of formants for unvoiced

[4] It is, however, also found important to use pitch-synchronous modulation of the noise-like component in waveform interpolation models [MK93, SK98].

sounds than for voiced sounds). Shaping of the noise-like signal component by an LP vocal tract filter is also commonly pursued in harmonic-plus-noise synthesis models [Sty01, Bai02b], with the LP filter coefficients deduced from LP analysis of the error signal from harmonic modeling.

Less obviously, the noise-like signal component of mixed excitation speech (including noise-like components of 'purely voiced' phonemes like vowels) has to be modulated in amplitude according to the phase of the oscillatory signal component. A not modulated (stationary) additive noise signal is not 'integrated' in the speech signal, meaning it is perceived as stemming from another signal source than the oscillatory signal component. This 'streaming' effect can be mitigated by using high-pass noise bursts or cyclo-stationary noise synchronized in phase with the periodic excitation in model based synthesis [Hol81, Her91, SK98, JS00a, JS00b, LS01]. Again, also in harmonic-plus-noise models, a pitch-synchronous modulation of the noise-like signal component using a parametric envelope is applied [SLM95, Bai02b] (cf. sect. 2.4.3).

5.2 Derivation of the oscillator-plus-noise model

Based on the observation that a low-dimensional oscillator model still fails to reproduce the high-frequency range of wide-band speech signals faithfully, and under the assumption that this is due to the missing modeling of high-dimensional noise-like signal components, we will now introduce the *oscillator-plus-noise model* for speech signals. The primary consideration for the derivation of this model is that the missing noise-like component in the oscillator output signal is related to the error signal of the respective predictor, and that an additive noise component similar to the prediction error signal should be generated and added to the oscillator output signal.

5.2.1 Properties of the prediction error

The prediction error of the nonlinear signal predictor used in the oscillator model for vowels is, as stated several times in Chapter 4, in the range of 20 to 30 dB below the mean signal power (for models yielding stable synthesis). The values stated for the prediction error have been computed for the training data, however, the error for predicting unseen data, e. g., from the validation data set in cross-validation, is commonly only slightly higher.

Also for the prediction of the 'glottal signal' of vowels the prediction error is of the same order of magnitude. However, concerning the spectral properties an important difference is found between the prediction error for the full speech signal and for the 'glottal signal': Whereas the spectrum of the full speech signal prediction error varies substantially, the prediction error on the 'glottal signal' commonly displays a flat (white) spectrum for many vowels, as depicted in fig. 5.3. The spectral flatness measure [MG76] for the examples in fig. 5.3 yields -10 dB for the full speech signal case and -4 dB for the 'glottal signal' case (a perfectly flat spectrum would have 0 dB flatness). In both cases the harmonically spaced spectral lines suggest a periodic behavior of the prediction error signal (not necessarily periodicity of the waveform).

In the auto-correlation function (fig. 5.4), whiteness relates to a quick decay of the absolute value of the correlation between delayed signal samples for increasing lag L. Whereas for the prediction error of the full speech signal, the absolute value of the auto correlation function decreases only slowly, for the prediction error of the 'glottal signal', the correlation between adjacent signal samples ($L = 1$) is already quite small, $|r_e(1)|/r_e(0) \simeq 0.1$. Periodicity of the waveform can be read from the peak at the fundamental period ($N_0 \simeq 120$), however, this peak is not large: $r_e(N_0)/r_e(0) \simeq 0.3$ (whereas the auto-correlation of the 'glottal signal' itself at the fundamental period is almost one: $r_x(N_0)/r_x(0) \simeq 0.99$).

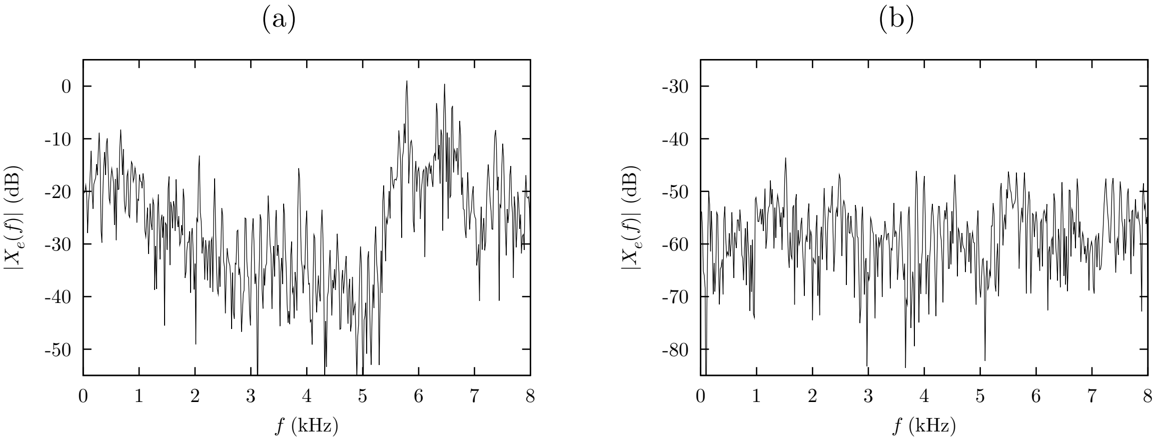


Figure 5.3: DFT magnitude spectra of the prediction error signal for (a) prediction of the full speech signal, and (b) prediction of the 'glottal signal' for vowel /o/, male speaker.

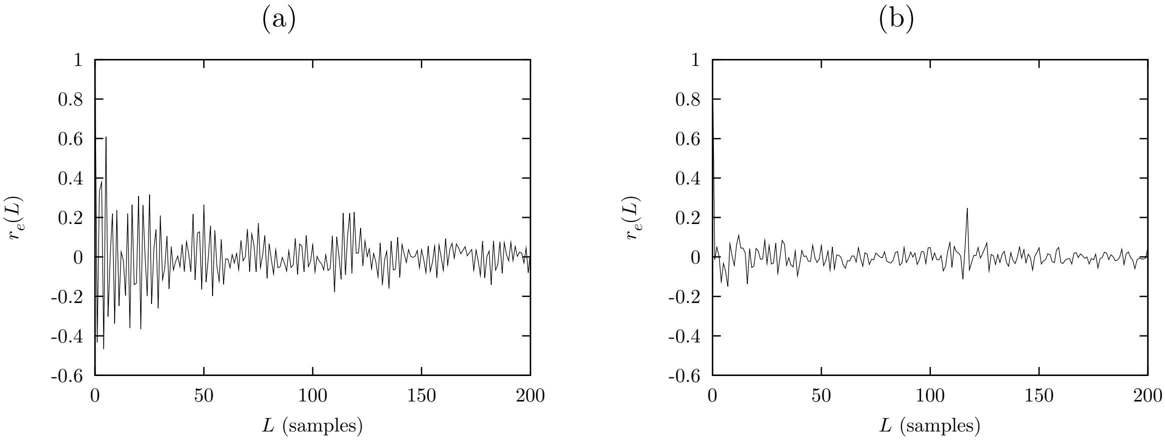


Figure 5.4: Auto-correlation of the prediction error signal for (a) prediction of the full speech signal, and (b) prediction of the 'glottal signal' for vowel /o/, male speaker.

Concerning the amplitude distribution, the prediction error signal almost perfectly fits a Gaussian distribution, as depicted in fig. 5.5.

The error signal of the nonlinear predictor applied to the 'glottal signal' of a vowel thus is almost white and nearly Gaussian distributed. We should, however, also take a look at the temporal evolution of the error signal, since a stationary random signal does not integrate perceptually with the oscillatory speech signal component, as noted in sect. 5.1.3. In fig. 5.6, along with the target 'glottal signal' $x_g(n)$ and the predicted signal $\hat{x}_g(n)$, the temporal evolution of the prediction error signal as well as an estimate of the error signal amplitude $\hat{a}(n)$ is displayed. $\hat{a}(n)$ is computed as the absolute value of the prediction error signal smoothed by a fifteen point moving average (MA) filter[5]. A modulation of the prediction error amplitude over time is evident, with maxima in amplitude at the minima of $x_g(n)$ (supposedly the instants of glottal closure), and, in this case, consistent maxima at the local minima before the maximum of $x_g(n)$ (possibly the instants of glottal opening). This observation, in addition to the low peak at the fundamental period in the auto-correlation of the waveform itself, indicates that the periodic behavior of the prediction error signal of voiced speech signals is mainly due to

[5] Computation of the MA smoothed Hilbert envelope of $e(n)$ yields a comparable amplitude estimation.

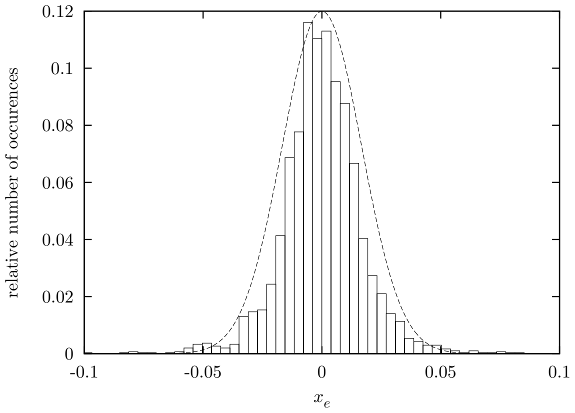


Figure 5.5: Amplitude distribution of the prediction error signal for prediction of the 'glottal signal' for vowel /o/, male speaker. The dashed line is a Gaussian function with the same variance as the prediction error signal.

its pitch-synchronous amplitude modulation.

An amplitude modulation of the prediction error signal with maxima at the glottis closure instants matches a speech production model, if we assume that the sources of the noise-like component in vowel signals is turbulent air flow at the vocal folds. These turbulent air flow will be more pronounced if the vocal folds are close to each other, as right before glottal closure or after glottal opening. Thus, besides a small systematic error (peak of the error auto-correlation at the fundamental period), the prediction error signal may well correspond to the noise-like excitation.

Summarizing, for vowel speech signals, the prediction error of the *nonlinear predictor* in the combined LP and oscillator model can be considered a modulated white Gaussian noise signal. Assuming that a similar signal component is still missing in the *oscillator model* output, we now have a method at hand to re-generate a noise-like component in addition to the oscillator output signal by adding an amplitude modulated white Gaussian noise signal. Modulation of the noise signal amplitude should be synchronized in phase with the oscillatory signal component. Although we have no means to immediately control the phase of the autonomous oscillator – and the oscillator output signal may display notable phase offset compared to the training signal after only a few periods – we can deduce the phase from the state of the oscillator, i. e., from the *location on the trajectory in phase space* over time.

5.2.2 Synthesis of vowels with noise

All the above considerations lead to the conclusion that synthetic vowel signals including the noise-like component can be re-generated by the LP and oscillator model if an appropriately amplitude modulated white Gaussian noise signal is added to the oscillator output signal. In the full speech signal this will lead to a compensation of missing high-frequency components, since the white noise is transformed into a high-pass signal by the $1/H_{\mathrm{lp}}(z)$ synthesis filter.

To ensure the phase-synchronous amplitude modulation of the noise component we propose the *oscillator-plus-noise model* depicted in fig. 5.7. The amplitude of the noise component is predicted from the current state of the oscillator $\boldsymbol{y}_{\mathrm{osc}}(n)$ by a second nonlinear function $f_n(\cdot)$. $f_n(\cdot)$ is realized by an RBF network of the same structure as the network for signal prediction[6]

[6] The nonlinear function for noise amplitude prediction might as well be realized simpler than the nonlinear function for signal prediction, i. e., using a nonlinear function model with less parameters than for signal prediction. Using RBF networks it is, however, of advantage to use the very same structure since the nonlinear

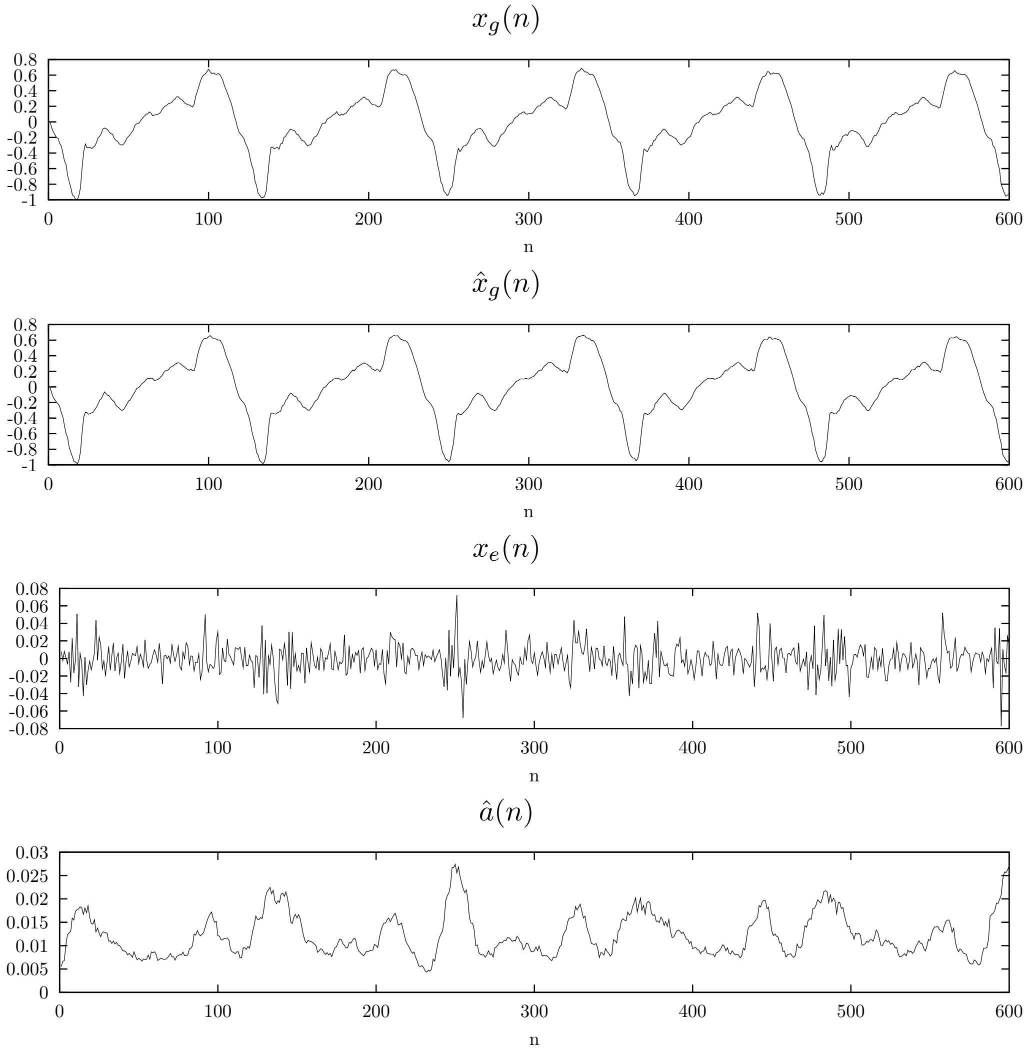


Figure 5.6: ‘Glottal signal’ $x_g(n)$, predicted ‘glottal signal’ $\hat{x}_g(n)$, prediction error $x_e(n) = x_g(n) - \hat{x}_g(n)$, and its estimated amplitude $\hat{a}(n)$ for vowel /o/, male speaker.

to predict the amplitude of the prediction error $\hat{a}(n)$ from the predictor (training) state vector $\boldsymbol{x}_g(n)$:

$$\hat{a}(n) = f_n(\boldsymbol{x}_g(n)) \ .$$

In the re-synthesis stage the amplitude of a (zero mean) white Gaussian noise signal $e(n)$ with variance $\sigma^2_{\text{noi}} = 1$, $p(e) = \mathcal{N}(0, 1)$, is modulated according to the oscillator state by

$$\tilde{a}(n) = f_n(\boldsymbol{y}_{\text{osc}}(n)) \ .$$

The modulated noise signal $y_{\text{noi}}(n) = \tilde{a}(n)e(n)$ is added to the oscillatory signal component $y_{\text{osc}}(n)$ to yield the synthetic ‘glottal signal’ $y_g(n)$. The oscillator-plus-noise synthesis model in fig. 5.7 is applied as a replacement for the ‘Oscillator Synthesis’ part in the combined LP and oscillator model (sect. 4.4, fig. 4.19 on page 71).

For sustained vowel signals the Bayesian learning of the RBF weights for $f_n(\cdot)$, from an estimated prediction error amplitude $\hat{a}(n)$ computed using five-point MA smoothing, reveals

part of the network function (i. e., the output of the basis functions) does not have to be re-computed as it is already available from the RBF network used for signal prediction.

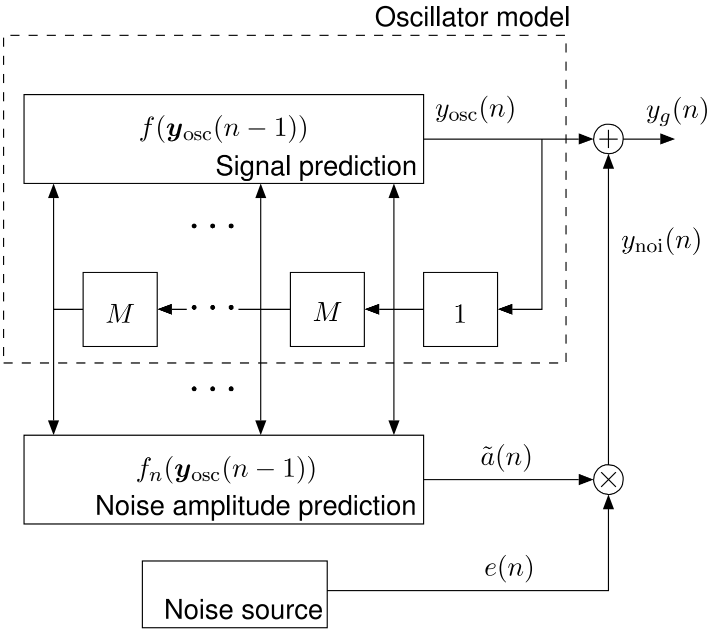


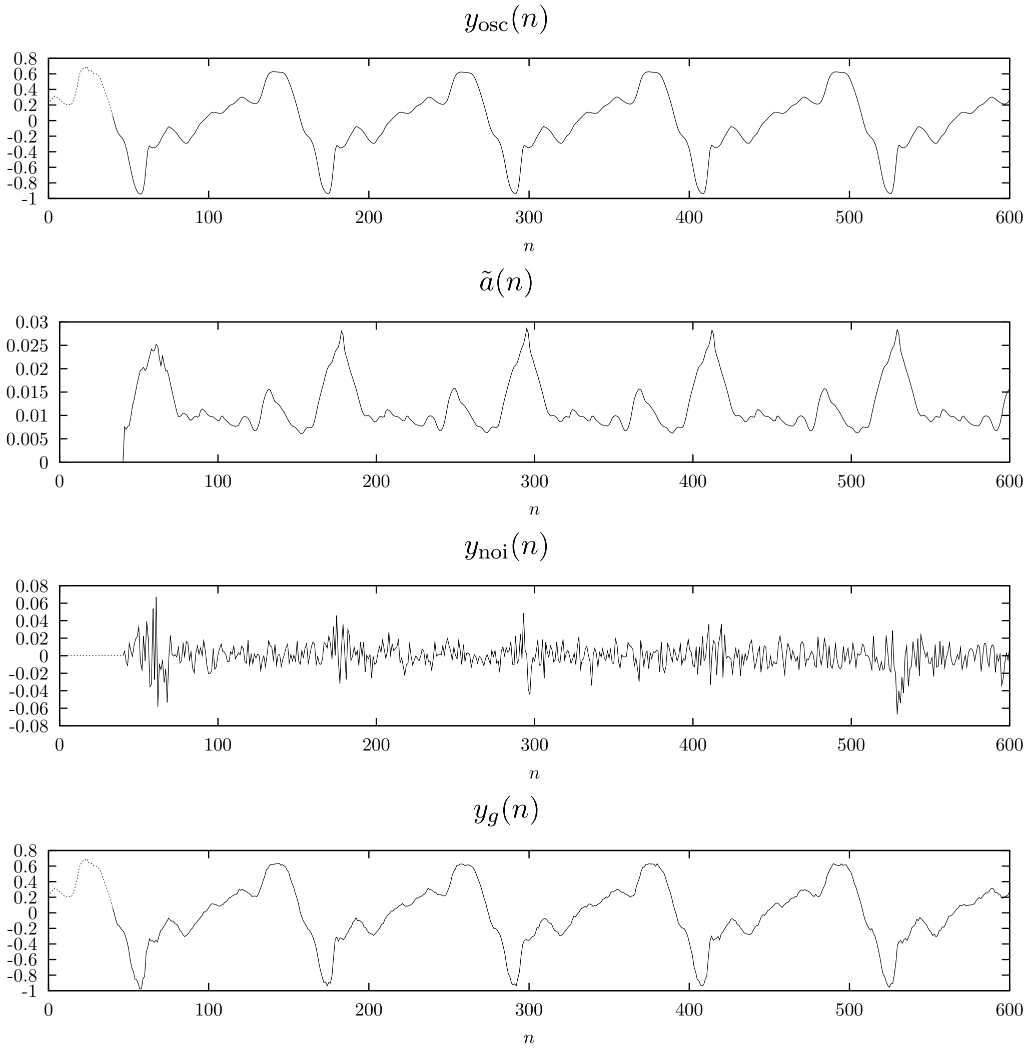
Figure 5.7: Oscillator-plus-noise model for synthesis of vowel sounds in the 'glottal signal' domain

relatively small variation of the resulting regularization factor λ_{bay}, with a geometric mean of $\text{gmean}(\lambda_{\text{bay}}) = 0.102$, and a standard deviation 11.3. Thus, for the training of the noise prediction model a fixed regularization factor of $\lambda = 0.1$ is used further on.

Signals generated by the oscillator-plus-noise model corresponding to the signals analyzed in sect. 5.2.1 (fig. 5.6 on the preceding page) are depicted in fig. 5.8. The oscillator is initialized with samples from the training signal and the noise amplitude with zeros, making up the first $(N-1)M+1 = 40$ samples in this case. In the following the amplitude $\tilde{a}(n)$ for the noise-like signal component displays an almost periodic evolution over time, in synchronization with the oscillatory signal component $y_{\text{osc}}(n)$. As for the estimated amplitude of the prediction error $\hat{a}(n)$, the largest maxima in $\tilde{a}(n)$ occur at the time instants of the minima of $y_{\text{osc}}(n)$, and secondary maxima are found at the local minima before the maxima of $y_{\text{osc}}(n)$.

A comparison between the original full speech signal, the 'glottal signal' derived by LP inverse filtering and low-pass filtering, and the according re-generated signals without and with the noise-like signal component is given in fig. 5.9. The essential difference between the perceptual quality of the synthetic speech signal generated by the oscillator-plus-noise model compared to the signal generated by the oscillator model without a noise signal added is certainly not evident from the time signals, due to the high dynamic range of both the full speech signal and the 'glottal signal'. The DFT spectra allow for a closer inspection: Whereas the line spectrum of the re-synthesized oscillatory component $y_{\text{osc}}(n)$ without the noise-like component added is retained in the full speech signal $y(n)$, the spectrum of the 'glottal signal' $y_g(n)$ with modulated white noise added displays a 'noise floor' that is transformed by the high-pass filter $1/H_{\text{lp}}(z)$ and the LP synthesis filter $1/A(z)$ to fit the spectrum of the original speech signal. Thus, the oscillator-plus-noise model yields a considerably better spectral reconstruction of the original signal than the combined LP and oscillator model alone (and much better than the oscillator model applied to the full speech signal).

A better modeling of the original speech signal can also be inferred from the phase-space representations. Here, the noise-like deviations from a periodic trajectory in the original full speech signal and in the according 'glottal signal', fig. 5.10 (a), that are not re-produced by the


$y_{\mathrm{osc}}(n)$

$\tilde{a}(n)$

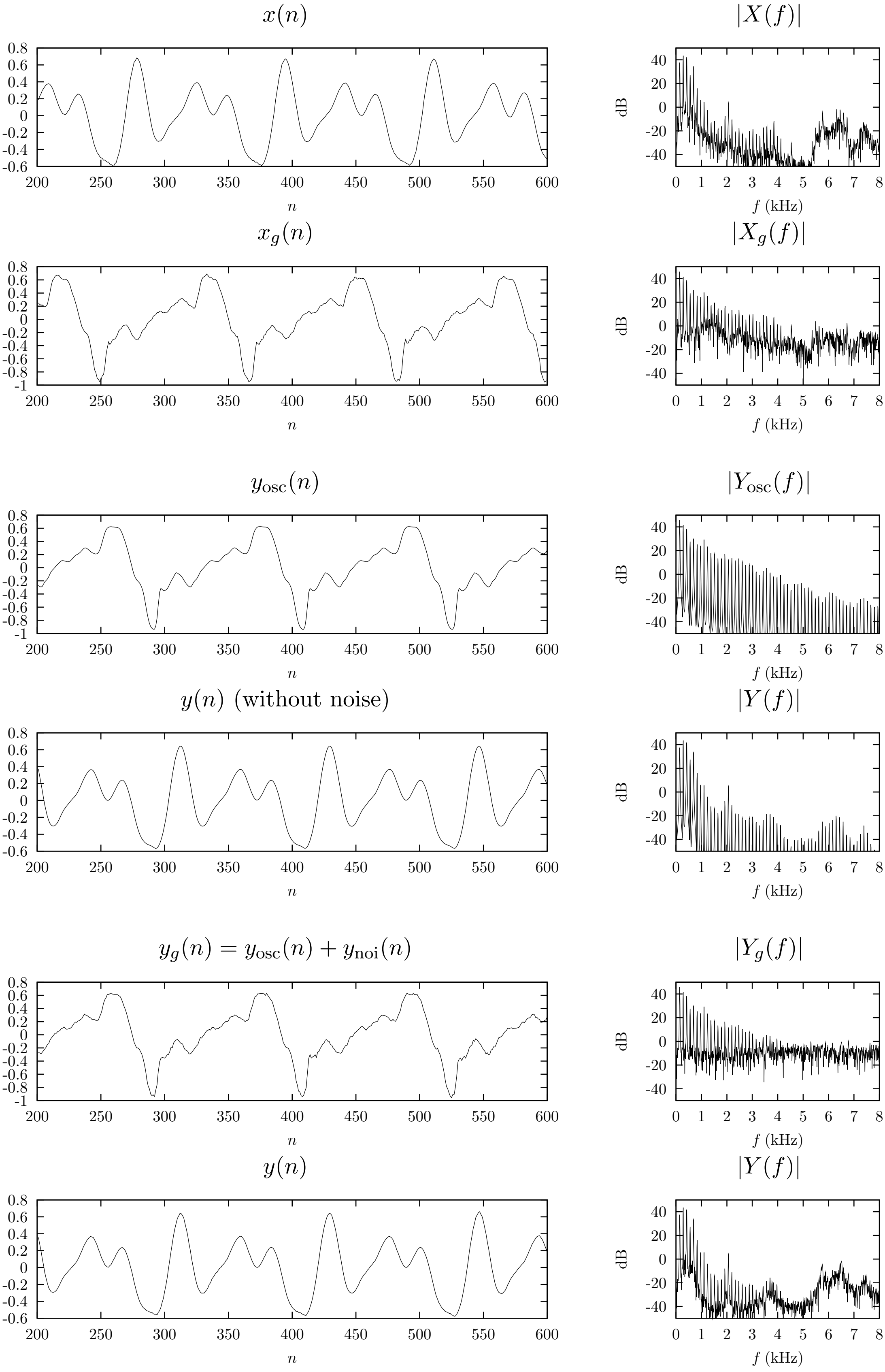
$y_{\mathrm{noi}}(n)$

$y_g(n)$

Figure 5.8: Oscillator generated signal $y_{\mathrm{osc}}(n)$, noise amplitude $\tilde{a}(n)$ predicted from the oscillator state vector, amplitude modulated noise-like component $y_{\mathrm{noi}}(n)$, and re-generated 'glottal signal' $y_g(n) = y_{\mathrm{osc}}(n) + y_{\mathrm{noi}}(n)$ for male vowel /o/. The oscillator is initialized with with signal values from the training signal and the noise amplitude with zeros (dotted lines).

oscillator model alone – the trajectories of the synthetic signals follow a limit cycle, fig. 5.10 (b) – are rendered in the output of the oscillator-plus-noise model, fig. 5.10 (c). According to the noise amplitude prediction the scale of the deviations varies over phase space. Little deviations are generated in regions where the signal is well predictable, as in the top-left quadrant of the phase space plots for the 'glottal signal', whereas larger deviations arise in regions where the original signal carries unpredictable noise-like components, as in the bottom half of the phase-space plots for the 'glottal signal'. Note, that the modulated noise component introduced in the 'glottal signal' is spread over time – almost equally effecting all parts of the trajectory of the full speech signal – by the IIR LP synthesis filtering.

The time evolution of the predicted noise amplitude $\tilde{a}(n)$ in the example depicted in fig. 5.8 closely resembles the parametric envelope for the noise-like signal component used in a harmonic-plus-noise model [SLM95]. However, in the oscillator-plus-noise model the evolution of the noise amplitude over the fundamental period is identified and re-generated individually for each vowel signal. Thus, the specific form of the envelope for the fundamental cycle may



$x(n)$ $|X(f)|$

$x_g(n)$ $|X_g(f)|$

$y_{\text{osc}}(n)$ $|Y_{\text{osc}}(f)|$

$y(n)$ (without noise) $|Y(f)|$

$y_g(n) = y_{\text{osc}}(n) + y_{\text{noi}}(n)$ $|Y_g(f)|$

$y(n)$ $|Y(f)|$

Figure 5.9: Time domain signals and DFT magnitude spectra for the original speech signal $x(n)$, the 'glottal signal' $x_g(n)$ found by LP inverse filtering and low-pass filtering, the oscillator generated signal $y_{\text{osc}}(n)$ and the re-generated full speech signal $y(n)$ without noise, and the oscillator-plus-noise model generated 'glottal signal' $y_g(n)$ and the resulting full speech signal $y(n)$ for male vowel /o/.

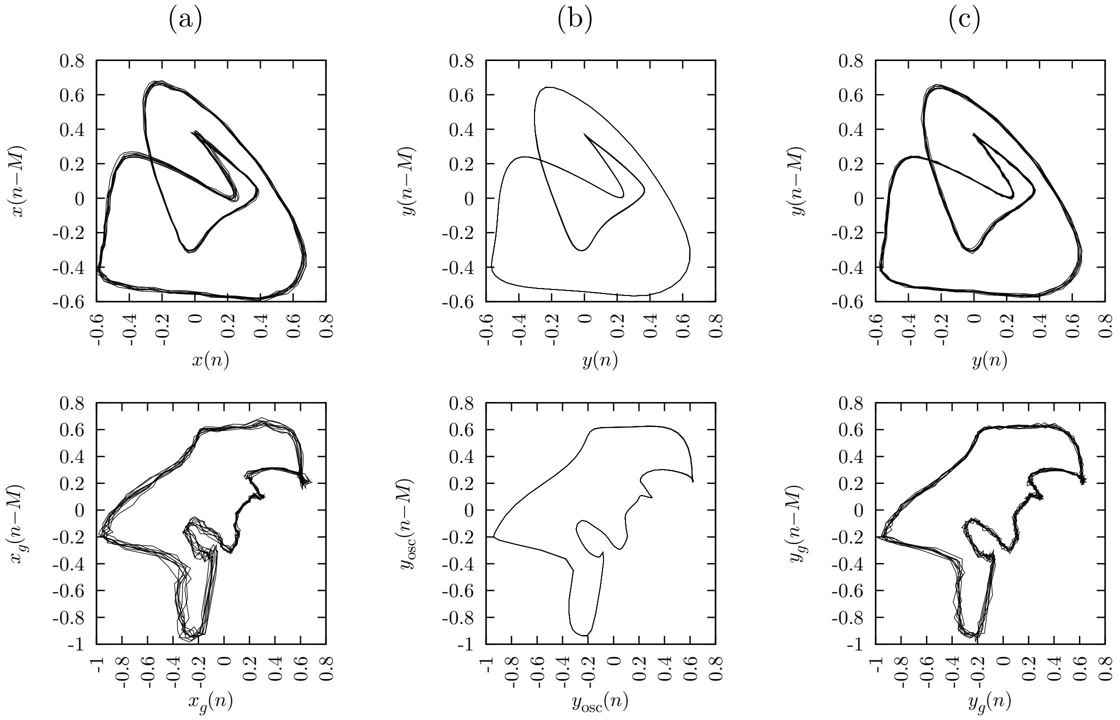


Figure 5.10: Two-dimensional projections of the phase-space representations of (a) original full speech signal (top) and 'glottal signal' found by inverse filtering and used as training signal for the oscillator (bottom), (b) 'glottal signal' re-generated by the oscillator model without noise (bottom) and resulting synthetic full speech signal (top), (c) 'glottal signal' generated by the oscillator-plus-noise model (bottom) and resulting synthetic full speech signal (top) for vowel /o/ from male speaker.

vary from one vowel to another. For example, whereas for some of the example vowel signals in App. C, figs. C.1-C.6, essentially one triangular maximum at the (supposed) glottis closure instant is observed, a second maximum (supposed at the instant of glottis opening) as in the example presented in figs. 5.6 and 5.8, with varying position and amplitude is found for other vowels.

Regarding the relation to harmonic-plus-noise models it has to be mentioned that the noise amplitude modeling used in the oscillator-plus-noise model could well be incorporated for the signal specific noise modulation in harmonic-plus-noise models by training the nonlinear function to predict the amplitude of the error signal of harmonic modeling from a delay embedding of the training signal, and predicting the noise modulation amplitude from an embedding of the generated harmonic signal component. As the harmonic modeling of most harmonic-plus-noise synthesis algorithms is applied in the full speech signal domain, there may, however, evolve problems for some signals corresponding to the difficulty to achieve stable behavior of the oscillator model in the full speech signal domain: Since the phase-space representation of some full speech signals is more complex than that of the 'glottal signal' (cf. signals for /a/, /e/, /i/ in figs. 4.9 and 4.21), the correspondence between signal phase in the fundamental period and position in phase space may not be one-to-one.

5.2.3 Analysis of mixed excitation speech signals

The mean energy of a noise-like component in vowel signals is relatively low. As mentioned above, the energy of the prediction error of a nonlinear predictor is in the range of 20 to 30 dB or more below the signal energy. The HNR estimated from the signal or from a harmonic-plus-noise synthesis model is in the same range (tables 5.1 and 5.2).

For mixed excitation speech signals, like voiced fricatives, HNR measures yield numbers in the range of 10 down to 3 dB. SNR estimates using the oscillator model with the Bayesian learning algorithm (cf. sect. 4.4.6) for modeling the full speech signal give similar results [Ran03], some numbers are given in table 5.3. Thus, the task to correctly identify the oscillatory component of mixed excitation speech signals can be considered substantially more difficult than for voiced signals, like vowels. However, the successful identification of LF source signals in additive noise with an SNR of 10 dB in sect. 4.4.6 suggests that also the oscillatory component of mixed excitation speech signals with a not too strong noise-like signal component can possibly be identified by an oscillator using Bayesian training.

Table 5.3: Estimated $\mathrm{SNR}_{\mathrm{bay}}$ for some vowels, as well as voiced and unvoiced fricatives. $\mathrm{SNR}_{\mathrm{bay}}$ is computed as mean speech signal power related to estimated noise power (i. e., as the posterior SNR).

Excitation	Voiced			Mixed			Unvoiced	
Phoneme	/aː/	/ɛ/	/oː/	/v/	/ʒ/	/z/	/s/	/f/
$\mathrm{SNR}_{\mathrm{bay}}$(dB)	25.4	29.8	36.1	19.1	12.7	4.9	5.5	0.39
$\lambda = \alpha\sigma_n^2$	2.2×10^{-4}	3.3×10^{-1}	1.5×10^{-3}	9.9×10^{-2}	2.4×10^{-1}	2.9	2.0	88

In this section we test the oscillator model on mixed excitation speech signals, like voiced fricatives, and we motivate an extension of the oscillator-plus-noise model for spectral shaping of the synthetic noise-like signal component for a better re-generation of such signals. First, we will take a look at one of the signals we are concerned with and the according ‘glottal signal’ derived by the inverse filtering procedure from sect. 4.4.3 in fig. 5.11. Here, a strong noise-like component, modulated in amplitude according to the phase of the oscillatory component, is clearly visible already in the full speech signal[7]. However, in contrast to the vowel signal (cf. fig 5.6), for the voiced fricatives the amplitude maxima of the noise-like component do in general not coincide with the glottis closure instants, (supposedly lying) at the minima of both the full speech signal and the ‘glottal signal’ in fig. 5.11 [JS00a]. Maximal noise excitation still may be related to the principal voiced excitation at glottal closure. However, the main noise excitation does not take place at the glottis, but at obstacles behind a restriction along the vocal tract where turbulent air flow is generated [JS00b, NA00]. Thus, the time for the acoustic impulse generated by the glottal closure to travel along the vocal tract to the restriction, and convection of turbulent air flow between the restriction and following obstacles results in a delay between the GCIs and the maxima of noise-like excitation. For the voiced fricative /z/, a delay of 4.0 ms is identified in [JS00b], corresponding to 64 samples at 16 kHz sampling rate, which seems a reasonable value also for the signal in fig. 5.11.

The inverse filtering procedure (sect. 4.4.3), in particular the low-pass filtering, attenuates the noise-like component. Thus, the task of nonlinear function estimation for the modeling of the oscillatory signal component of mixed excitation speech signals by the oscillator model is comparable to the task of modeling vowel signals.

As found for several example signals (cf. also App. C), identification and re-synthesis of the oscillatory signal component of voiced fricatives with the oscillator model is possible, but again for many voiced fricatives only in the glottal signal domain (cf. [Ran03]). The analysis

[7] Recall that we had to look at the nonlinear prediction error signal to identify the modulation for vowel signals.

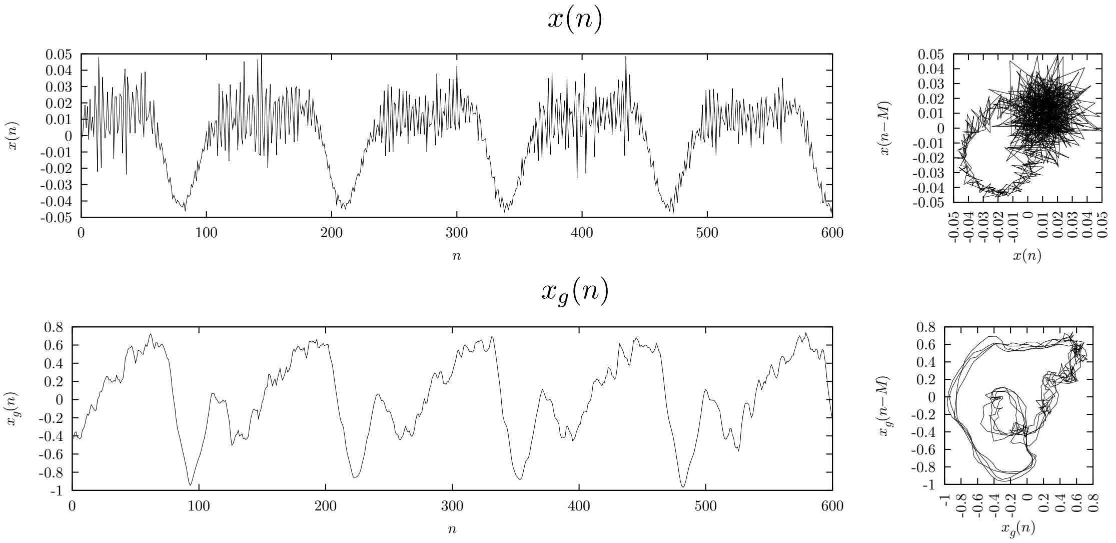


Figure 5.11: Mixed excitation speech signal $x(n)$ from the voiced fricative /z/, and the according 'glottal signal' $x_g(n)$ found by inverse filtering, along with their two-dimensional phase-space representations.

and re-synthesis of the voiced fricative /z/ from fig. 5.11 in the 'glottal signal' domain with the oscillator-plus-noise model as described in sect. 5.2.2 is depicted in fig. 5.12. Note, that for this signals the estimated (and the re-generated) noise amplitude envelope have a distinctively different form than for the vowel signal in fig. 5.8 on page 99. Here, the maxima of the noise amplitude are lying in between the minima of the 'glottal signal'. Using the noise amplitude prediction function, this behavior is well reproduced in the synthetic noise signal. Re-synthesis of voiced fricatives, however, often results in a perceptually poor output signal. In particular the synthetic signals seem to have different spectral properties than the original signals.

The conjectured reason is that for voiced fricatives our model may not suffice to correctly reproduce the noise-like signal component, as the assumption that the noise-like component is generated as a modulated white noise signal at the same physical position as the oscillatory component (i. e., at the glottis) and filtered by the same vocal tract transfer function is not appropriate. As noted above, for voiced fricatives the noise-like excitation is due to turbulent air flow at restrictions along the vocal tract. Thus, the filtering due to sound propagation in the vocal tract can pose a different spectral influence on the noise-like source signal than on the source signal of the oscillatory component, due to the different source positions.

The analysis of the DFT spectrum and auto-correlation function of the prediction error for the 'glottal signal' of the voiced fricative /ʒ/, in the same way as performed for vowel signals in sect. 5.2.1, figs. 5.3 and 5.4, reveals a less white DFT spectrum and a slower decay of the autocorrelation function for the voiced fricative than for vowels, as depicted in fig. 5.13. Hence, in this case the re-generation of the noise-like speech component by a modulated white noise signal added to the oscillator generated 'glottal signal' is not satisfactory.

5.2.4 Oscillator-plus-noise model with a second LP path

To enhance the reproduction of voiced fricatives – or mixed excitation speech signals in general – we propose to extend the oscillator-plus-noise model by a second LP analysis/synthesis path, in addition to the one for the oscillatory signal component, with the objective to whiten the prediction error signal during analysis [Kub95] and to achieve the correct spectral shaping of

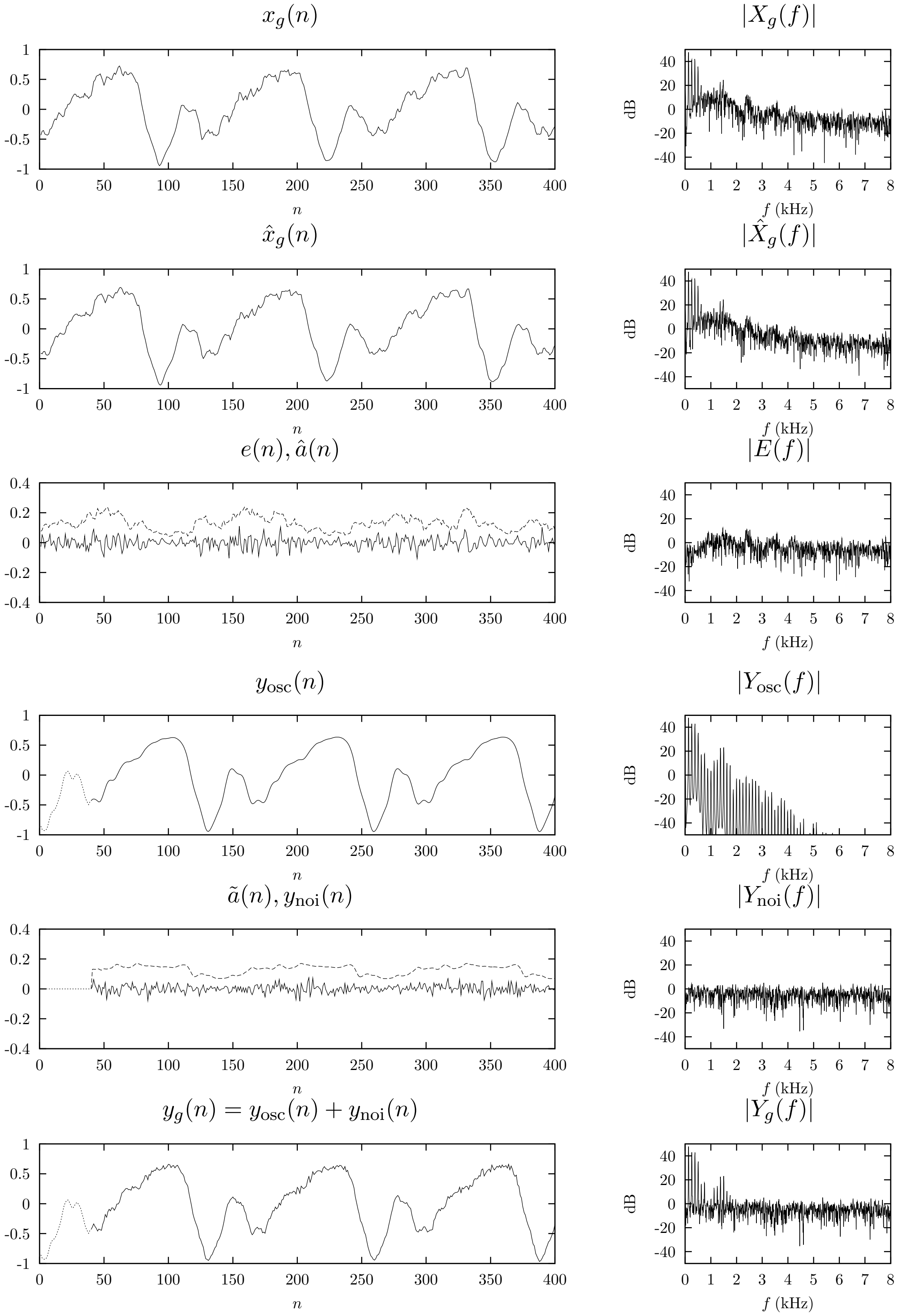


Figure 5.12: Time domain signals and DFT magnitude spectra of the original 'glottal signal' $x_g(n)$ found by LP inverse filtering and low-pass filtering, the predicted signal $\hat{x}(n)$, and the prediction error signal $e(n)$ along with its amplitude estimate $\hat{a}(n)$ (dashed, exaggerated) for voiced fricative /z/. For synthesis the oscillator generated signal $y_{\text{osc}}(n)$, the predicted noise amplitude $\tilde{a}(n)$ and the accordingly modulated noise signal $y_{\text{noi}}(n)$, as well as the oscillator-plus-noise model generated 'glottal signal' $y_g(n)$ are depicted. The respective original and synthetic full speech signals can be found in fig. 5.17.

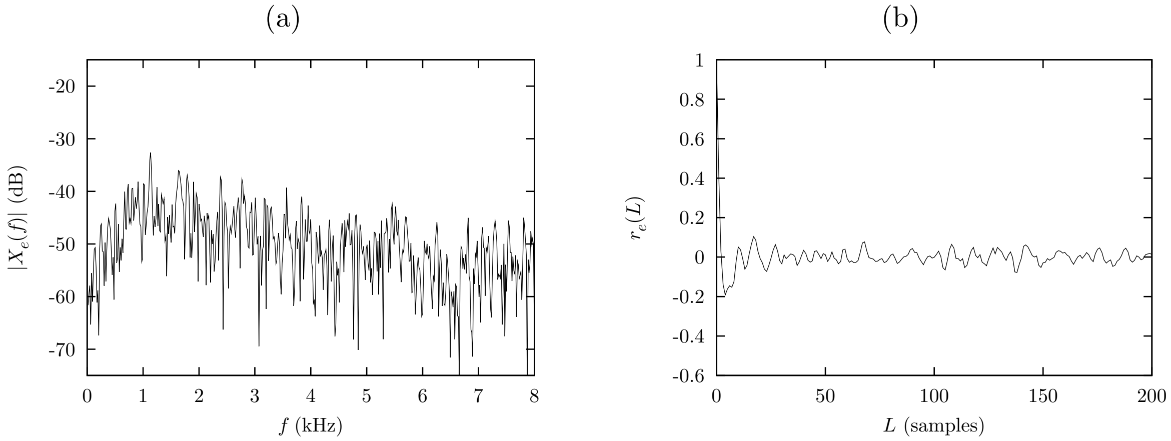


Figure 5.13: DFT magnitude spectrum (a) and auto-correlation function (b) of the prediction error on the 'glottal signal' for a voiced fricative /z/, male speaker.

the noise-like signal component in the synthesis stage.

A schematic of the proposed extension of the oscillator-plus-noise model is depicted in fig. 5.14. The main difference to the previous model is the introduction of a second LP path for spectral analysis and *individual spectral shaping* of the noise-like signal component during synthesis. The noise-like component in the full speech signal domain $\hat{x}_{\text{noi}}(n)$ is derived as the difference of the original speech signal $x(n)$ and the predicted 'glottal signal' $\hat{x}_g(n)$ filtered by the synthesis filters $1/H_{\text{lp}}(z)$ and $1/A(z)$. LP analysis of the signal $\hat{x}_{\text{noi}}(n)$ (without pre-emphasis) determines the coefficients for a second LP inverse filter $A_{\text{noi}}(z)$. Thus, the noise-like component $\hat{x}_{\text{noi}}(n)$ is whitened. From the according residual signal $x_{r\_\text{noi}}(n)$ the amplitude trajectory for the noise-like component is extracted by rectification and ten point moving average filtering. The estimated amplitude trajectory $\hat{a}_{\text{noi}}(n)$ of the whitened signal $\hat{x}_{r\_\text{noi}}(n)$ is used to train the noise amplitude prediction model. In the synthesis stage a white noise signal is modulated in amplitude by $\tilde{a}_{\text{noi}}(n)$, which is predicted from the oscillator state variable $\boldsymbol{y}_g(n)$ as before. The modulated white noise signal, however, is now filtered by the separate LP synthesis filter $1/A_{\text{noi}}(z)$. The resulting noise component in the full speech signal domain $y_{\text{noi}}(n)$ is added to the oscillator generated signal $y_g(n)$ filtered by the synthesis filters $1/H_{\text{lp}}(z)$ and $1/A(z)$, to form the synthetic speech signal $y(n) = y_{\text{osc}}(n) + y_{\text{noi}}(n)$. In this way the spectral properties of the oscillatory component and the noise-like component are shaped independently, allowing for a faithful spectral reconstruction of both speech signal components.

In fig. 5.15 the different spectral shaping of the oscillatory and the noise component is illustrated by the frequency responses of the synthesis filters for the oscillatory and the noise-like signal component of the voiced fricative /ʒ/. Although the transfer functions of the two synthesis filters display an overall similar shape, the synthesis filter for the noise-like signal component emphasizes the frequency range between 1.5 and 2.5 kHz by about 6 dB as compared to the filters in the oscillatory signal path.

To illustrate the effect of using a second LP path to arrive at a whitened noise-like component the DFT spectra of the according nonlinear prediction error signals $e(n) = x_g(n) - \hat{x}_g(n)$ and of the residual signal of the second LP analysis filter $\hat{x}_{\text{r\_noi}}(n)$ are depicted in fig. 5.16. Note, that if the non-white prediction error signal $e(n)$ is modeled by a white noise source and filtered by synthesis filters $1/H_{\text{lp}}(z) \cdot 1/A(z)$ with a frequency response as in fig. 5.15 (as in the oscillator-plus-noise model without a second LP path), this results in a severe lack of noise-like signal components in the frequency range between 1.5 and 2.5 kHz, because these components are (a) not emphasized in the white noise source *and* (b) not amplified by the synthesis filters.

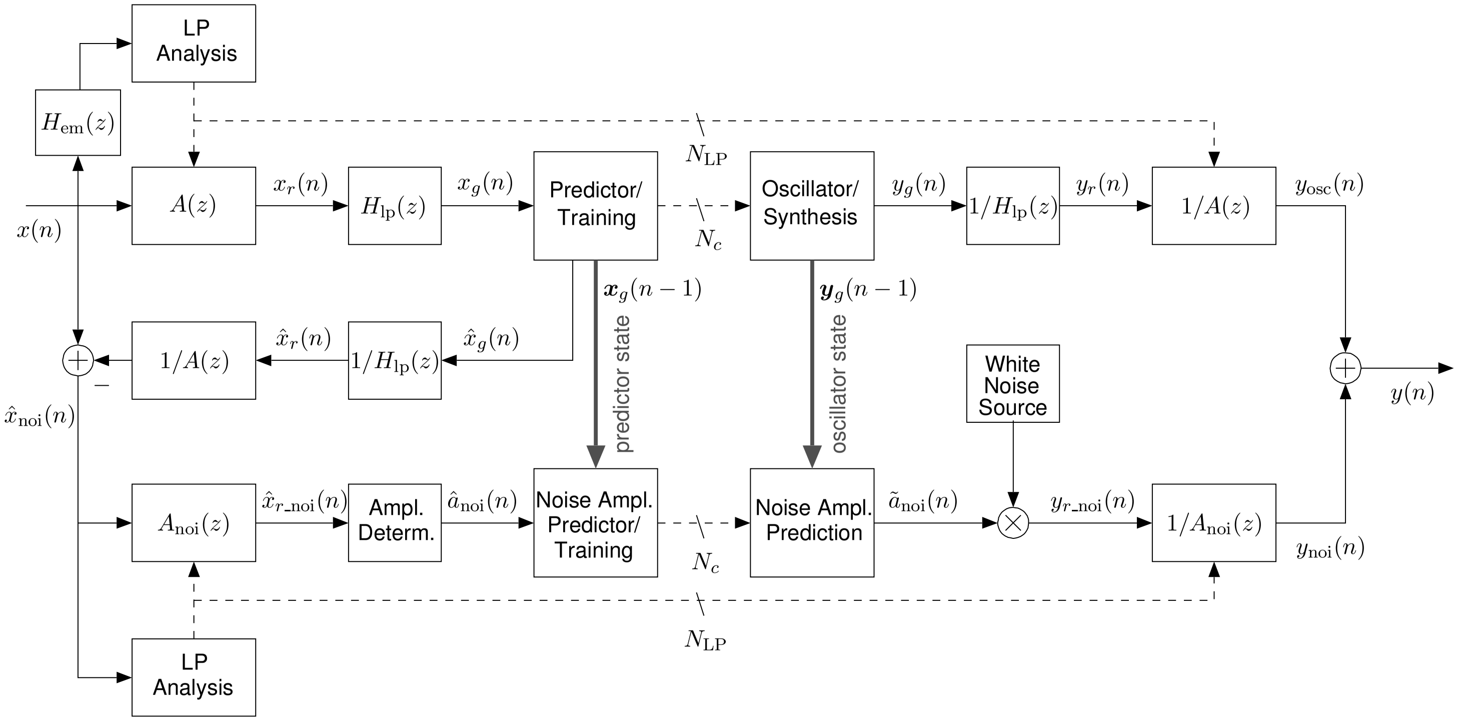


Figure 5.14: Oscillator-plus-noise model augmented by a second LP path for the spectral shaping of the noise-like signal component independent of the oscillatory component.

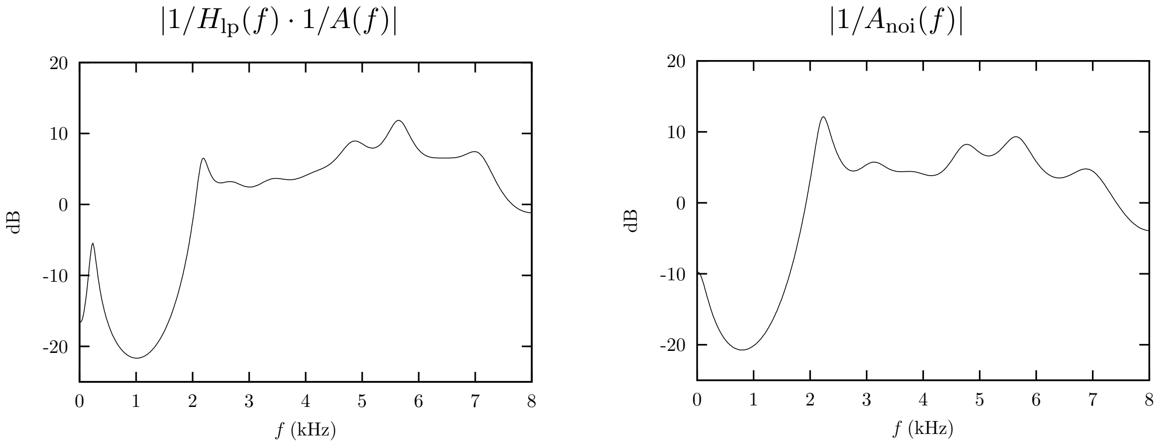


Figure 5.15: Transfer functions of the synthesis filters $1/H_{\mathrm{lp}}(z) \cdot 1/A(z)$ for the oscillatory signal component, and $1/A_{\mathrm{noi}}(z)$ for the noise-like signal component in the extended oscillator-plus-noise model for voiced fricative /ʒ/.

In the oscillator-plus-noise model extended by a second LP path, on the other hand, the spectrum of the residual signal of the second LP inverse filter $\hat{x}_{\mathrm{r\_noi}}(n)$ is close to white, and the white noise signal used for synthesis is filtered by $1/A_{\mathrm{noi}}(z)$, which emphasizes the noise-like components around 2 kHz, resulting in a perceptually more natural spectral shaping of the noise-like signal component.

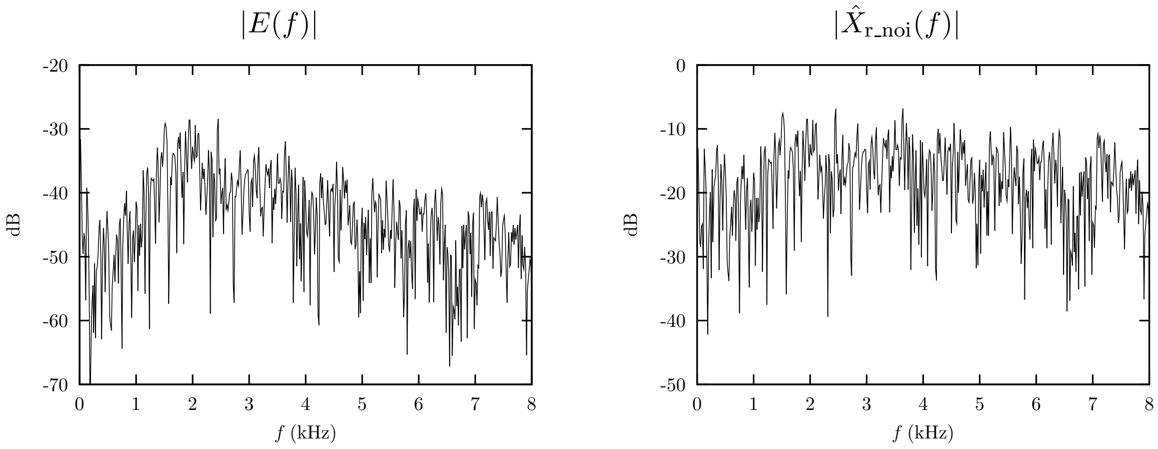
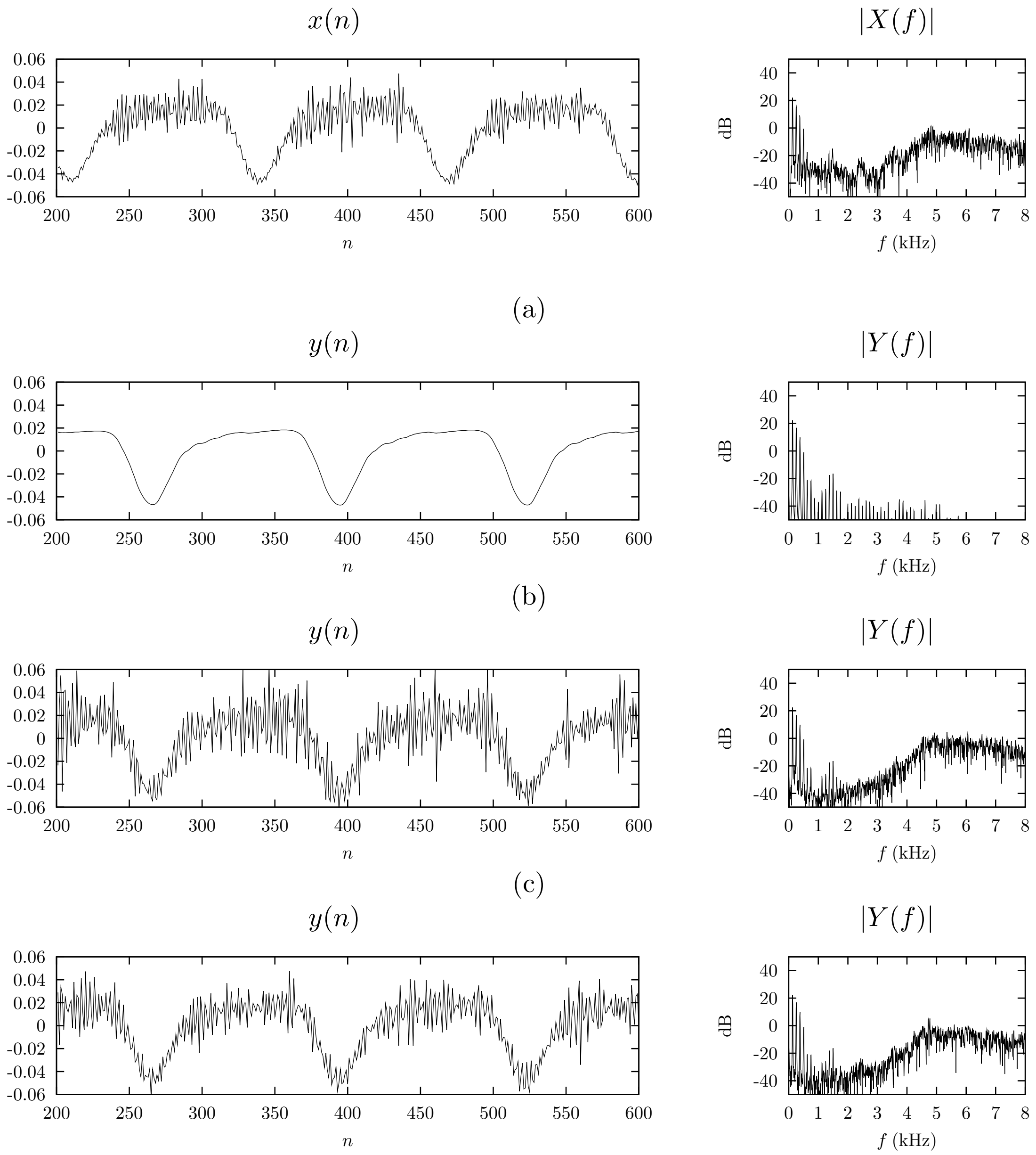


Figure 5.16: DFT magnitude spectra of the prediction error signal $e(n) = x_g(n) - \hat{x}_g(n)$ and of the residual signal $\hat{x}_{\mathrm{r\_noi}}(n)$ from the second LP inverse filter in the extended oscillator-plus-noise model for voiced fricative /ʒ/.

In fig. 5.17 the time-domain signals and DFT spectra of the original signal of the voiced fricative /z/ and the regenerated signals from (a) the oscillator model with one LP path without noise added, (b) the oscillator-plus-noise model with one LP path, and (c) the oscillator-plus-noise model augmented with the second LP path for the noise-like component are depicted. From (a) it is clearly seen that such mixed excitation signals are not regenerated adequately with the oscillator model alone. Comparing the DFT spectra in (b) and (c) we see a small, but perceptually important increase of magnitude of the noise-like signal component around 2 kHz due to the individual LP filtering. And from the time-domain signals in (b) and (c) we may also deduce that the envelope modeling for the noise-like component is improved if a



(a)

(b)

(c)

Figure 5.17: Full speech signals and DFT spectra of the voiced fricative /z/. Original speech signal $x(n)$, and synthetic signals $y(n)$ of (a) the oscillator model with one LP path without noise, (b) the oscillator-plus-noise model with one LP path, and (c) the oscillator-plus-noise model with two LP paths.

dedicated LP filter is used.

More examples for the application of the extended oscillator-plus-noise model to a number of speech signals, with their time-domain signals, their two-dimensional phase-space representation, and DFT spectra are given in App. C. Note the clearly different noise envelope estimated and re-generated for the voiced fricatives (figs. C.7-C.9), with a minimum at the supposed glottis closure instant, in comparison to the noise envelope for the vowels (figs. C.1-C.6), with a maximum at the supposed glottis closure instant.

The figures in App. C also comprise examples for the analysis and re-synthesis of unvoiced fricatives in figs. C.10-C.11. As seen for the modeling of noisy LF source signals (sect. 4.4.6), the oscillator using Bayesian trained RBF models assigns a high regularization factor for signals with a low SNR, resulting in a constant output signal. For most unvoiced fricatives the very same behavior – low SNR and accordingly high regularization factor, cf. table 5.3 – is encountered, and the oscillator generates a constant output signal close to zero. This results in the re-generation of unvoiced sounds by the oscillator-plus-noise model (with or without the second LP path) in the way of a noise-excited LP model (with a somewhat higher computational effort, though). A premise is of course the prediction of an appropriate amplitude for the excitation noise signal by the noise amplitude prediction function from an oscillator state corresponding to the constant oscillatory output signal, i. e., an appropriate flat function $f_n(\cdot)$ around zero. For all unvoiced fricatives which were re-synthesized in our experiments an appropriate amplitude prediction has been observed.

From the synthesis examples of unvoiced fricatives it can be seen that the low frequencies (below about 1 kHz) in the noise component $y_{\mathrm{noi}}(n)$ are attenuated, in comparison to the original speech signal. From a comparison of the noise floor in the spectra of the original signal and the oscillator-plus-noise generated signal this also seems to be the case for voiced fricatives. This may be due to the low-pass applied in the inverse filtering process, making the low frequency noise-like signal components in $x_g(n)$ predictable. Hence, these low frequency components are missing in the estimated noise-like signal component, and in the re-synthesized signal. A corresponding observation is the lower estimate for the noise level for LF signals in sect. 4.4.6, if low-pass filtered noise is added instead of white noise.

5.3 Assessment of naturalness

Although the utmost assessment of naturalness of synthetic speech signals is up to human judgement, i. e., to perceptual listening tests, we like to provide a first objective rating of the oscillator-plus-noise model regarding naturalness of the generated signals by comparing measures for jitter and shimmer, and for trajectory divergence behavior of oscillator generated signals with the according measures for natural speech signals.

5.3.1 Jitter and shimmer

Measures for jitter – short-term variations of the fundamental period – and shimmer – short-term variations of the amplitude of individual fundamental cycles – are often considered as related to naturalness of speech signals (see, e. g., [AI96]). Too high values for both measures, for example, indicate 'pathological voices' (e.g., due to vocal fold cancer). Too low values mean that the speech signal is close to periodic, which is virtually only the case for artificially generated speech signals, and which seems to contribute significantly to the easy identification of such 'computer voices' by humans.

It is thus desirable to achieve the right amount of short-term variations in a synthetic speech signal. In many synthesis algorithms jitter and shimmer are artificially introduced by the fundamental frequency control algorithm. The oscillator model – being a nonlinear system – is in principle capable of generating even chaotic signals, hence it should be possible to model and re-generate these short-term variations. In both [NPC99] and [MM01] it is reported that the jitter values for the oscillator generated signals match the according values in natural speech signals.

Measures for jitter and shimmer

Our computation of jitter and shimmer is as follows: Pitchmarks are computed in the 'glottal signal' domain using the function `pitchmark` from the Edinburgh Speech Tools (included in the Festival speech synthesis system, [BC05]). Pitchmark positions in time $m(i)$, with $i = 1, \ldots, N_m$ being the fundamental cycle index and N_m the number of identified pitchmarks, are at the minima of the 'glottal signals' – corresponding to glottis closure instants – at integer values of discrete time n. For the speech signals from a male speaker sampled at $f_s = 16\,\mathrm{kHz}$ with a fundamental period of $N_0 \simeq 100$ samples, discretization of jitter would be in 1 %-steps, with 1 % being a typical amount of jitter in natural sustained speech signals.

To increase the temporal resolution, all signals are up-sampled by a factor M,

$$\begin{aligned} x'(n') &= x(n'/M) \qquad &&\text{for} \quad n' = kM \ , \\ x'(n') &= 0 &&\text{for} \quad n' \neq kM \ , \quad k \in \mathbb{N} \ , \end{aligned}$$

and interpolated by a linear phase filter (according to [SdG91], for this task, interpolation and linear phase FIR filtering is reported to outperform, e. g., polynomial interpolation or FFT filtering). Pitchmarks $m'(i) = Mm(i)$ for the interpolated signal are shifted to the minima of $x'(n')$ within the range of $\pm M$ samples of discrete time n' (i. e., within the range of the last and the following sample in original discrete time n), thus refining the temporal resolution. For our purpose an interpolation factor of $M = 7$ has been found sufficient.

The local fundamental period $N_0(i)$ is computed as

$$N_0(i) = \frac{m'(i+1) - m'(i)}{M} \ , \qquad i = 1 \dots N_m - 1 \ . \tag{5.1}$$

Note, that $N_0(i)$ may take non-integer values.

The jitter time series $u_j(i)$ is computed as

$$u_j(i) = \frac{N_0(i+1) - N_0(i)}{\bar{N_0}} \ , \qquad i = 1 \dots N_m - 2 \ , \tag{5.2}$$

with $\bar{N_0}$ being the mean fundamental period

$$\bar{N_0} = \frac{1}{N_m - 1} \sum_{k=1}^{N_m - 1} N_0(k) \ . \tag{5.3}$$

The shimmer time series $u_s(i)$ is computed from the interpolated signal values at the pitchmark instants, normalized to the signal peak-to-peak amplitude, according to

$$u_s(i) = \frac{x'(m'(i+1)) - x'(m'(i))}{\max(x'(n')) - \min(x'(n'))} \ , \qquad i = 1 \dots N_m - 1 \ . \tag{5.4}$$

Nominal measures σ_j for jitter and σ_s for shimmer are computed as the standard variation of the respective time series

$$\sigma_j = \mathrm{std}(u_j(i)) \ , \qquad \sigma_s = \mathrm{std}(u_s(i)) \ . \tag{5.5}$$

Results for the oscillator generated signals

Using regularized RBF networks we only seldom find synthetic signals with a value for jitter or shimmer that significantly differs from zero. This may be due to the rather high amount of regularization applied by the Bayesian training algorithm, but also for a smaller regularization factor as determined by cross-validation the values for jitter and shimmer do not increase. Moreover, also manually varying the regularization factor for GRBFs to a minimum for still stable synthesis does not result in natural jitter and shimmer values in general.

There are, however, some cases where the 'correct' amount of jitter or shimmer can be reproduced by the oscillator model alone. Careful tuning of the regularization parameter for an oscillator using GRBFs (to $\lambda = 2.3 \times 10^{-2}$) results in a nominal jitter measure of $\sigma_j = 0.87\,\%$

for the oscillator generated signal of the vowel /o/, compared to $\sigma_j = 1.06\,\%$ found for the natural signal. For a higher regularization factor – but also for lower regularization factor – the jitter measure approaches zero (e. g., $\sigma_j = 0.08\,\%$ for a regularization factor according to Bayesian training).

The proper jitter measure for the specific case, however, goes with a far too large nominal shimmer measure of $\sigma_s = 8.04\,\%$, compared to $\sigma_s = 1.07\,\%$ found for the natural signal. Moreover, looking at the jitter and shimmer time series in detail, reveals that in this case the durarion and the amplitude of individual fundamental cycles basically switches between two discrete values. This behavior – bifurcation, and generation of sub-harmonics – is a typical feature of deterministic nonlinear oscillators (see, e. g., [CHNV82, Moo92]), but is normally not found in speech signals from healthy speakers, but rather indicates pathological voices [VMJ96, dPG00, CBM03], the expression of emotions like fear [KS95], or depression [OSS+04].

As depicted in fig. 5.18, the natural signal (a) displays pitchmarks varying apparently randomly[8]. The according histograms for the jitter and shimmer time series indicate a mono-modal Gaussian-like probability density function. The oscillator generated signal (b) is almost strictly periodic, and the jitter and shimmer distributions are concentrated around zero. The signal from our tuned oscillator (c) with the (almost) correct nominal jitter measure, however, displays a first sub-harmonic in the signal and a bi-model pdf for jitter and shimmer. The signal is perceived as equally non-natural just like the strictly periodic signal (b).

In fig. 5.18 (d) the signal generated by the oscillator-plus-noise model and its jitter and shimmer distribution is depicted. Here the almost strict periodicity of the oscillator signal is broken by the added noise signal. As for the natural signal, mono-modal distributions for jitter and shimmer are found. The same is true for the signal generated by the extended oscillator-plus-noise model, depicted in fig. 5.18 (e). Since the noise-like signal component is added to the oscillatory component only in the full speech signal domain, here, the jitter and shimmer analysis was done for the signal found by inverse filtering of the full speech output signal $y(n)$ using the filters for the oscillatory component $A(z)\,H_{\mathrm{lp}}(z)$ (i. e., without a further LP analysis).

The nominal measures for jitter and shimmer obtained for the signals in fig. 5.18 are given in table 5.4. Although the measures for the oscillator-plus-noise generated signals are somewhat too large, meaning that the noise amplitude may be over-estimated if the prediction error amplitude is used for its estimation – signals (d) and (e) are clearly perceived as more natural than signals (b) or (c).

Table 5.4: Nominal jitter and shimmer measures for the signals in fig. 5.18

Signal	(a)	(b)	(c)	(d)	(e)
Jitter: σ_j	1.06 %	0.08 %	0.87 %	1.55 %	1.62 %
Shimmer: σ_s	1.07 %	0.00 %	8.04 %	1.54 %	1.39 %

A statistical evaluation of the jitter and shimmer measures on our database of sustained speech signals, comprising nine vowels, three nasals, and one liquid, each recorded from nine female and eleven male speakers gives the numbers in table 5.5. The analysis was restricted to signals that could be stably re-synthesized by the oscillator model and for which reliable pitchmarks could be computed.

[8] As already mentioned, the positions of pitchmarks, as well as the jitter time series of natural speech signals can be partially predicted, and modeled by an AR process [SG97, EK96].

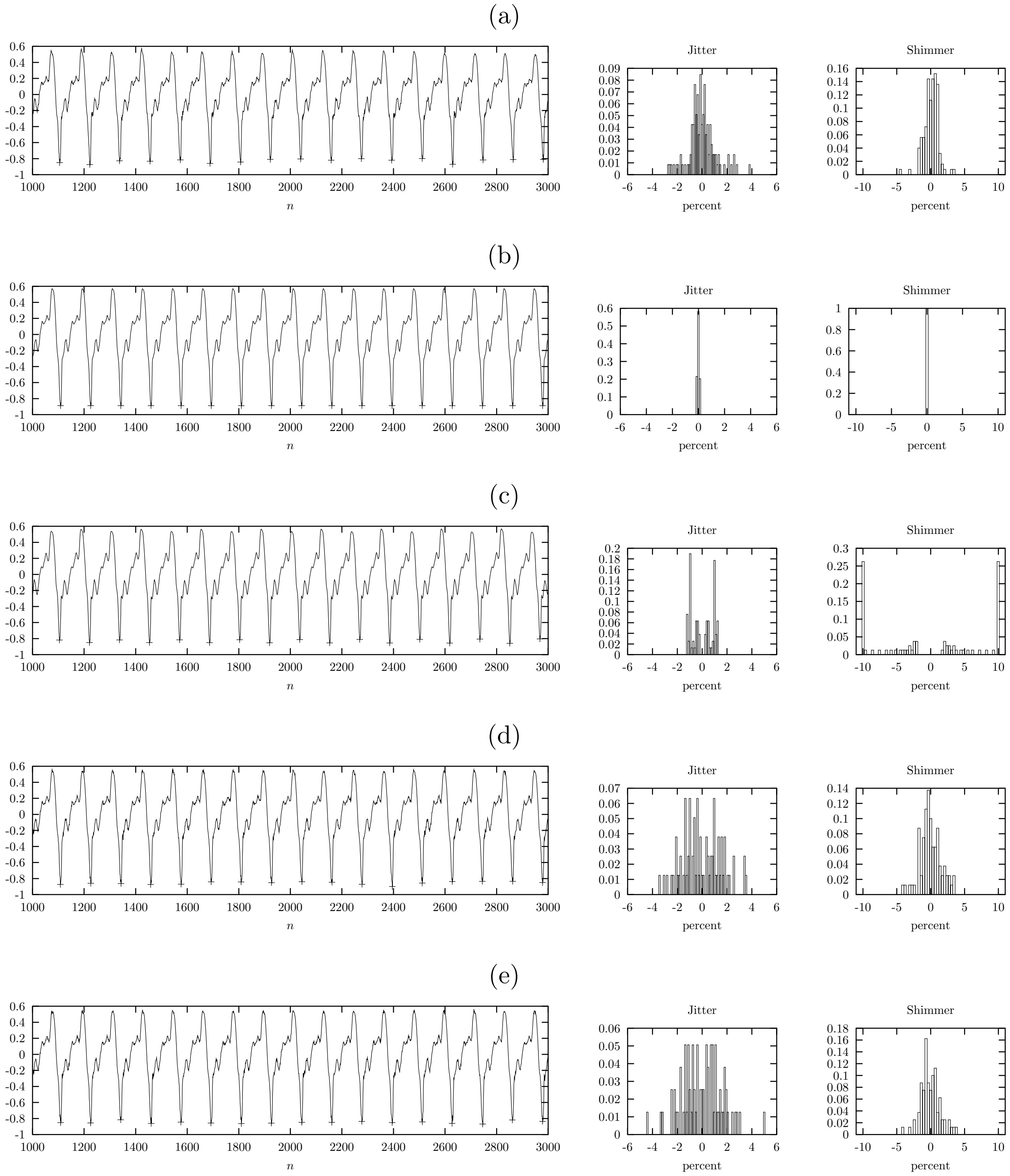


Figure 5.18: Time-domain signals with pitchmarks (plus signs) and according histograms for the jitter and shimmer time series $u_j(i)$ and $u_s(i)$ of (a) the natural 'glottal signal' $x_g(n)$, (b) the oscillator generated signal $y_o(n)$ for an oscillator with Bayesian regularization (regularization according to cross-validation yields a similar behavior), (c) the oscillator generated signal $y_o(n)$ for GRBFs with $\lambda = 2.3 \times 10^{-2}$, (d) the oscillator-plus-noise generated signal $y_g(n)$, (e) the extended oscillator-plus-noise generated signal $y(n)$, inversely filtered by $A(z)\,H_{\mathrm{lp}}(z)$. The according nominal jitter and shimmer measures are listed in table 5.4.

Table 5.5: Statistical results for the nominal jitter and shimmer measures for signals from a database of sustained vowel/nasal/liquid sounds for (a) the original signals, (b) the synthetic signals generated by the oscillator without noise, (c) the oscillator-plus-noise model generated signals, and (d) the extended oscillator-plus-noise model generated signals.

	(a)	(b)	(c)	(d)
mean(σ_j)	1.01 %	0.28 %	1.42 %	1.42 %
std(σ_j)	0.81 %	0.75 %	1.06 %	1.03 %
mean(σ_s)	1.45 %	0.58 %	2.14 %	2.13 %
std(σ_s)	0.74 %	2.21 %	1.86 %	1.55 %
$c(\sigma_j, \sigma_j^{\text{orig}})$	1	0.025	0.55	0.50
$\text{gmean}(\frac{\sigma_j}{\sigma_j^{\text{orig}}})$	1	0.098	1.47	1.49
$c(\sigma_s, \sigma_s^{\text{orig}})$	1	0.053	0.12	0.23
$\text{gmean}(\frac{\sigma_s}{\sigma_s^{\text{orig}}})$	1	0.029	1.36	1.39

The numbers in table 5.5 reveal that the expected values $E(\cdot)$ for both the jitter and the shimmer measure from the signals generated by the oscillator model alone are about one third of the measure from the natural signal, while the oscillator-plus-noise model and the extended model yield measures about 50 % higher than in the original. However, also the variance std$(\cdot)$ of the jitter and shimmer measures is high, indicating that the overall mean values are not very meaningful.

Some more information can be found from the correlation $c(\sigma_x, \sigma_x^{\text{orig}})$ between the measures for the original signal and for the synthetic signals. Here, the signals from the oscillator model without noise show a correlation of only 0.025 for the jitter measure, while the oscillator-plus-noise generated signals have correlation of 0.55 and the signals from the extended oscillator-plus-noise model 0.50. Closer inspection reveals that, as already mentioned above, the signals generated by the oscillator alone are often almost periodic, i. e., have a jitter and shimmer measure close to zero, and only few signals display variations of individual fundamental cycles (like sub-harmonics) that, however, lead to jitter and shimmer measures much higher than for the original signal.

For the signals generated by the (extended) oscillator-plus-noise mode, on the other hand, the jitter measures correlate quite well with the ones for the original signals. This characteristic can also be read from the geometric mean of the quotient between the jitter measure of the synthetic and that of the original signals, $\text{gmean}(\frac{\sigma_s}{\sigma_s^{\text{orig}}}) = \exp\left(\text{mean}(\ln \frac{\sigma_j}{\sigma_j^{\text{orig}}})\right)$. The geometric mean of 0.098 for signals generated without noise says that on average these signals have a jitter measure one order of magnitude lower than the original signals. For the (extended) oscillator-plus-noise model generated signals the numbers show that the jitter measure is of the correct order of magnitude, although somewhat too high. One arrives at similar findings analyzing the numbers for the shimmer measure.

5.3.2 Trajectory divergence

The divergence of trajectories in phase space is a characteristic feature of nonlinear systems. For signals generated by nonlinear systems two trajectories passing through nearby points in phase space may diverge even on short time intervals. For a discrete time system this property is exemplified in fig. 5.19.

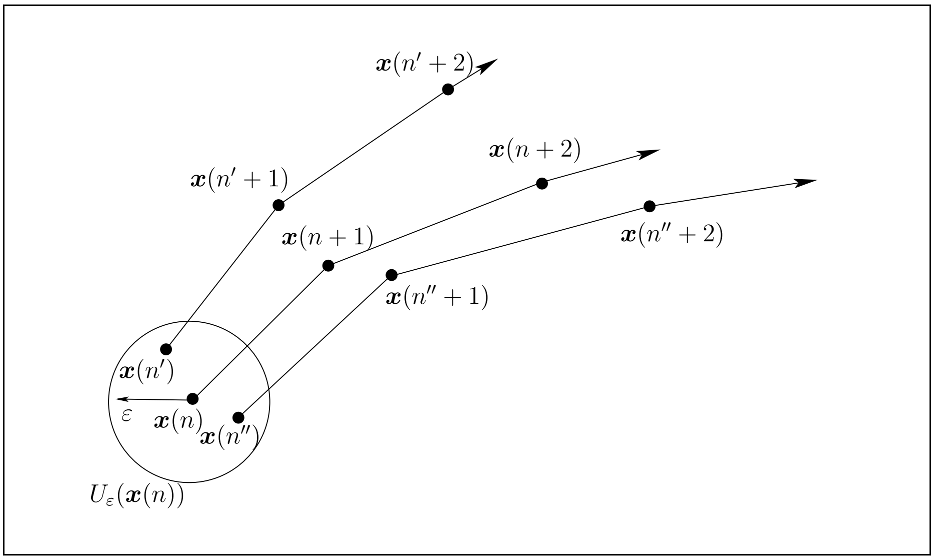


Figure 5.19: Divergence of signal trajectories in phase space

The divergence of trajectories in phase space can be illustrated by a set of functions $S(l; N, \varepsilon)$ of relative time lag l, parameterized by the embedding dimension N and a distance ε defining the size of the neighborhood region $U_\varepsilon(\boldsymbol{x}(n))$. A point on the trajectory is considered to be 'near' to $\boldsymbol{x}(n)$ if it is inside the region $U_\varepsilon(\boldsymbol{x}(n))$. The functions characterizing divergence are computed as

$$S(l; N, \varepsilon) = E\left(\log_{10}\left(\|\boldsymbol{x}(n'+l) - \boldsymbol{x}(n+l)\|\right)\right) \quad , \qquad \forall \boldsymbol{x}(n') \in U_\varepsilon(\boldsymbol{x}(n)) \ . \tag{5.6}$$

The expected value $E(\cdot)$ in this equation is approximated by averaging over all occurrences of points $\boldsymbol{x}(n') \in U_\varepsilon(\boldsymbol{x}(n))$.

An example for $S(l; N, \varepsilon)$ for a natural speech signal is depicted in fig. 5.20 (a). For a chaotic signal, the largest Lyapunov exponent can be deduced from the functions $S(l; N, \varepsilon)$ if a clear linearly increasing slope (i. e., an exponential divergence of the trajectories) starting at a small lag l up to some reasonable lag before saturation is observed for all values of neighborhood size ε, and for all embedding dimensions N above a certain minimum embedding dimension [HKS99]. The resulting positive value for the largest Lyapunov exponent would indicate chaotic behavior, which has been claimed for speech signals [BM94], but which is also often questioned.

Since for the sustained vowel signals used here a sufficiently clear linear slope in the functions $S(l; N, \varepsilon)$ could not be observed, we limit ourselves to comparing the trajectory divergence behavior of natural and oscillator generated signals based on the similarity of the functions $S(l; N, \varepsilon)$ in the example plots in fig. 5.20.

From these plots we see that the oscillator generated signal with a regularization factor determined Bayesian learning, fig. 5.20 (b), or by cross-validation, fig. 5.20 (c), display only a minimal divergence of trajectories, characterized by the flat evolution of the functions $S(l; N, \varepsilon)$. The oscillator with the regularization parameter tuned for the correct jitter measure (cf. sect. 5.3.1) yields a somewhat more similar divergence behavior for the functions $S(l; N, \varepsilon)$ for lower embedding dimension $N = 3$ and $N = 4$ (top-most lines of each function bundle starting at the same ordinate value at $l = 0$ in fig. 5.20 (d)). However, for the higher embedding dimensions $N = 5$ and $N = 6$ the functions $S(l; N, \varepsilon)$ are again quite flat. This

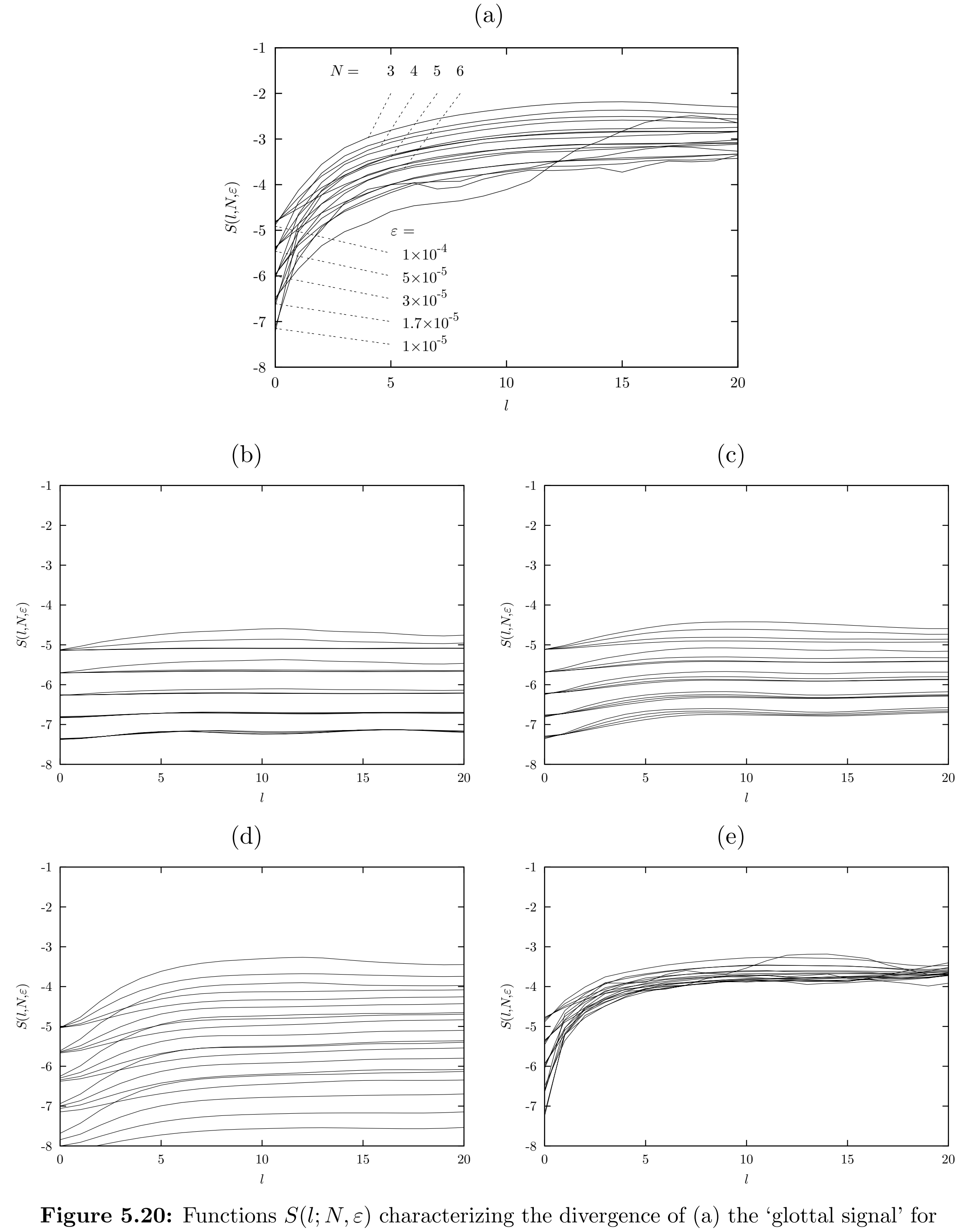


Figure 5.20: Functions $S(l; N, \varepsilon)$ characterizing the divergence of (a) the 'glottal signal' for a natural speech signal (vowel /o/), and of oscillator generated 'glottal signals' for this vowel, with (b) the synthetic signal generated by the oscillator without noise with $\lambda = 6 \times 10^{-5}$ according to Bayesian learning, (c) with $\lambda = 1 \times 10^{-9}$ according to cross-validation, (d) with regularization tuned for the correct jitter measure ($\lambda = 2.3 \times 10^{-2}$, cf. sect. 5.3.1), and (e) for the signal generated by the oscillator-plus-noise model with Bayesian regularization. The values for the embedding dimension N and for the size of the neighborhood ε indicated in subplot (a) were used for all other subplots, too.

is of course related to the observation in sect. 5.3.1 that the so tuned oscillator generates a signal with period doubling only and not a high-dimensional chaotic signal.

For the oscillator-plus-noise generated signal, on the other hand, the functions $S(l; N, \varepsilon)$ show a general increase independent of embedding dimension N in fig. 5.20 (e), although the signals do not diverge to the same degree as the original signal and the functions $S(l; N, \varepsilon)$ saturate at a lower value (at an ordinate value of about -3.6 as compared to -2.8 for the original signal).

It may seem that the divergence behavior of the natural signal could be best reproduced by a combination of the behavior of the manually tuned oscillator's divergence in fig. 5.20 (d) and that of the oscillator-plus-noise model in fig. 5.20 (e), i. e., a more natural signal could be obtained by a tuned oscillator with added noise. This would, however, leave us with the task to manually tune the regularization parameter of the oscillator model for each training signal.

These observations lead us to conjecture that, in terms of trajectory divergence of the generated signal, the oscillator-plus-noise model will in general provide a more natural behavior as compared to the oscillator model without added noise.

5.4 Conclusion

Investigations on the nature of speech show that a noise-like signal component is an important integral part of voiced speech signals. This is obvious for 'mixed excitation' speech signals, like voiced fricatives. But also for 'purely voiced' speech signals, like vowels, the noise-like signal component plays an essential role. This signal component is added to the oscillatory component in many state-of-the-art synthesis systems like, e. g., elaborate formant or harmonic-plus-noise synthesis algorithms, or in speech coders based on waveform interpolation. Considering that the oscillator model is a low-dimensional signal model we have to provide additional means to include the high-dimensional noise-like component in the generated synthetic speech signal.

From speech analysis and former research (as well as from simple experiments with stationary additive noise signals) it becomes clear that a pitch-synchronous amplitude modulation of the noise-like signal component is requisite for high quality synthesis. Furthermore, analysis of the speech generation process and our experiments show that for the faithful re-generation of general mixed excitation speech signals an individual spectral shaping of the noise-like signal component is necessary.

To satisfy these requirements we introduce the oscillator-plus-noise model, providing the means of pitch-synchronous amplitude modulation of the noise signal component based on the state of the oscillator, and the extended oscillator-plus-noise model for individual spectral shaping of the noise-like signal component by a distinct LP analysis and synthesis path.

For vowel signals the analysis of the prediction error signal of the nonlinear predictor used in the oscillator model resembles an amplitude modulated white Gaussian signal if the oscillator model is used in combination with LP inverse filtering. For many vowel signals the amplitude modulation of the prediction error signal matches well with speech production models that suggest noise bursts correlating with glottal closure and glottal opening, and the corresponding amplitude envelope used for noise modulation in harmonic-plus-noise synthesis. With the oscillator model – in contrary to many other speech synthesis methods – this amplitude modulation need not be modeled by a parametric envelope, since no explicit information about the signal phase is available. However, the signal phase is related to the position in phase space encoded by the oscillator state vector. A clear picture of the relation between signal phase and position in phase space is encountered for the 'open loop' trajectory structure of inverse filtered speech signals.

To achieve a pitch-synchronous modulation, the amplitude of the noise-like signal component is predicted from the state vector of the oscillator. To this end a second nonlinear function is introduced and trained to predict the amplitude of the error signal of the nonlinear predictor in the oscillator model from the predictor state vector. During synthesis the second nonlinear function generates an amplitude envelope for the additive noise signal based on the state vector of the oscillator. The additional effort of using another nonlinear function, introducing a number of additional parameters to the synthesis model, may seem untoward, but it allows for an individual modeling of the modulation envelope for each speech sound, which proves particularly relevant for mixed excitation speech signals.

In vowel signals re-generated by the oscillator-plus-noise model, the added noise signal results in an improved spectral reproduction of the original speech signal. The line spectrum often encountered for signals generated by the oscillator model is accompanied by a noise baseline in the lower frequencies and, due to the high-pass filtering in the synthesis process, the noise signal component dominates the higher frequency range in the full speech signal domain and yields a good reproduction of the spectral components missing until now.

In contrast to vowel signals, for mixed excitation speech signals – like voiced fricatives – the modeling of the prediction error signal by a white noise signal is not appropriate anymore. The larger spectral variation of the prediction error signal calls for an additional filtering of this signal to arrive at a noise-like signal component that can be substituted by a white noise signal. To this end we introduce a second LP path in the oscillator-plus-noise model for whitening of the noise-like speech component, and for the correct spectral shaping of the additive white noise signal during synthesis.

When modeling voiced fricatives a major benefit of the modeling of the noise amplitude by a state-dependent nonlinear predictor becomes evident: The estimated and re-generated envelope of the noise-like signal component for voiced fricatives displays a distinctively different shape than for vowels. With the second nonlinear predictor, however, the noise amplitude can be faithfully re-generated for each individual signal.

Using the same structure and training methods the oscillator-plus-noise model is also able to re-generate unvoiced fricatives.

The synthetic speech signals generated by the oscillator-plus-noise model resemble the original signals of vowels and voiced fricatives in oscillatory waveform shape, spectral characteristics, and in the form of signal trajectory in phase space. Furthermore, the objective measures for naturalness considered here, show that – with the noise-like signal component added – the synthetic signals also resemble the natural speech signals in terms of jitter and shimmer measure, as well as concerning trajectory divergence.

Chapter 6

Summary and conclusions

Based on previous investigations in the nonlinear oscillator model, on recent concepts of nonlinear function identification, and on findings from speech signal analysis and synthesis, we have presented robust methods for the application of the oscillator model to synthesis of stationary speech signals, including the re-production of a noise-like signal component.

6.1 Contributions

During the last decade the autonomous oscillator model and its application to speech signal modeling has been investigated in a number of publications. Most of these investigations targeted the modeling of the oscillatory signal component of vowels, with the aim of the correct re-production of features of the natural speech signal, like cycle-to-cycle variations. Also, most of the investigations directly applied the oscillator to modeling the full speech signal, since the nonlinear predictor in the oscillator model can incorporate linear prediction widely used in other speech processing tasks. However, good results of stably re-synthesized speech signals are often achieved with manual tuning of model parameters and for specific training signals, only.

In this thesis we have developed the means for modeling and stable re-synthesis of a large number of stationary speech signals using one and the same oscillator model structure, as well as an extension of the oscillator model allowing for the re-generation of the noise-like component of speech signals in addition to the oscillatory component.

The increase in the number of signals that can be stably re-synthesized is achieved, on the one hand, by the application of inverse filtering by linear prediction – which yields a simple trajectory structure of the oscillatory speech component in embedding phase space – and, on the other hand, by the robust estimation of regularization for the RBF networks used as nonlinear prediction function by a Bayesian training algorithm.

Inverse filtering transfers the signal modeling task to the ‘glottal signal’ domain, thus shifting the application of the oscillator from modeling a signal influenced by a widely varying resonance filter (the vocal tract) to the modeling of (an estimate for) the oscillatory signal generated by the vocal fold oscillation. Although such a modeling does not strictly fall in the class of ‘physical modeling’ the advantage of employing the oscillator model for modeling the ‘glottal signal’ can be read from the number of successfully re-synthesized vowel signals.

The comparison of nonlinear function models on a one-dimensional modeling task in Chapter 3 and for vowel signals in Chapter 4 reveals that a smooth function model like an RBF network is a favorable choice for modeling the oscillatory signal component using the oscillator model. Like in many other practical applications, the utilization of regularization for nonlinear function learning is mandatory. For this task the Bayesian estimation of the regularization factor for an RBF network has been shown to be a convenient tool for oscillator identification.

Considering that the oscillator model alone can only re-generate the low-dimensional oscillatory signal component of a speech signal we have investigated the means necessary for the additional adequate re-generation of the noise-like signal component of speech for the first time in this thesis. To this end the features of the noise-like component of voiced speech signals are analyzed and an extension of the oscillator model for the re-generation of the noise-like speech component, related to formant synthesis and harmonic-plus-noise models, is presented as the oscillator-plus-noise model.

The oscillator-plus-noise model provides the means for an appropriate pitch-synchronous amplitude modulation of the noise-like signal component based on the evolution of the state vector of the signal generation oscillator – i. e., the current position in phase space – and allows for the individual spectral shaping of oscillatory and noise-like speech signal components. Thus, stationary vowel signals, mixed excitation signals like voiced fricatives, and also unvoiced fricatives can be re-synthesized with high fidelity, resembling the natural speech signals in their spectral properties, in the form of signal trajectories in phase space, and perceptually[1]. A comparison of the synthetic signals with natural speech signals concerning short-term variations of fundamental period length and amplitude of individual fundamental cycles, as well as divergence of signal trajectories in phase space confirms these findings.

6.2 Discussion and potentials

Identification of the parameters of an oscillator model with an RBF network with Gaussian basis functions as nonlinear prediction function by means of Bayesian learning is a robust way to achieve successful re-synthesis of the oscillatory component of stationary speech sounds, particularly in combination with inverse filtering based on linear prediction. Our approach, aiming at a model with a pre-defined structure and as many parameters fixed as possible, shows that high-fidelity synthesis of speech with one oscillator model structure for all stationary speech sounds can be accomplished, with the signal dependent parameters being only the RBF network weights and the LP filter coefficients.

The oscillator-plus-noise model presented provides the means for a similar faithful re-generation of the noise-like signal component, allowing for the synthesis of natural vowel signals, signals for nasals and liquids, as well as voiced and unvoiced fricatives, i. e., all stationary speech sounds in general. Like the features of the oscillatory speech component, amplitude modulation envelope and spectral shape of the noise-like signal component are identified from the training signal. The necessity of correct amplitude modulation of a noise-like signal component is proved by perception experiments [Hol81, Her91, SK98] and the need to provide different modulation for different speech sounds is underlined by analysis results [JS00a, JS00b, LS01].

With our model variations in the speech signal related to the spoken phoneme, to fundamental frequency, and to speaker identity can be captured and re-generated, which is difficult with other model-based synthesis methods, like speech signal generation by the LF model [PQR99]. This also points at the capability to model, e. g., emotional speech sounds or different speaking styles.

[1] Audio examples can be retrieved from http://www.spsc.tugraz.at/erhard/publ/diss.html.

> Wenn ich alles Mißlungene ſo ausführlich, wie das Obige, hätte beſchreiben wollen, ſo hätte ich dieſes Werk leicht um einen Band vermehren können, welches aber wider meine, und gewiß auch des Leſers Abſicht geweſen wäre. Genug, wenn ich hier ſage, daß ich, alles zuſammen genommen, leicht ſo viel Maſchinwerk verworfen habe, als ſich mit einem ſtarken Pferde kaum fortbringen ließ.

Wolfgang van Kempelen [vK70, pp. 407f].

Disagreeing with Wolfgang van Kempelen, who states that ≫*it is against my and certainly as well against the reader's intention to also describe all failures in detail*≪, we want to point out some of the problems and drawbacks encountered during our work and the challenges pertaining for the use of the oscillator model for general purpose speech synthesis.

One of the most obvious drawbacks of the oscillator model as used here is the high complexity. The more versatile behavior of our signal generation model as compared to, e. g., parametric signal models as the LF model, is linked to a considerably large number of signal dependent parameters: For most of the examples in this thesis the oscillator model comprised an RBF network with several hundred weights (as compared to 5 parameters in the LF model), and in the oscillator-plus-noise realization used for our examples the number of parameters roughly doubles. It seems, however, not possible to achieve adequate speech synthesis with a smaller RBF network (i. e., when reducing embedding dimension and/or number of basis functions per dimension). Application of the RVM algorithm, however, can significantly reduce the RBF network complexity in the oscillator, and the complexity of the predictor for the noise amplitude may be simply reduced by choosing a smaller network (thus, however, trading the benefit that the basis functions have to be evaluated only once if the same RBF network is used for the oscillator and for noise prediction).

Despite the increase in the number of speech sounds that can be successfully re-synthesized, a major problem remains the identification of a stable oscillator from natural speech signals. Looking at the numbers of stably re-synthesized signals from a database of sustained vowel signals, in a considerable number of cases oscillator identification still fails. The primary reason for this is that many of the signals in the database are not as stationary as necessary (in particular variations in the signal amplitude, like a linear increase/decrease, result in an unstable oscillator, as found for linearly amplitude modulated LF signals, too). Hence, the training of the oscillator model and of the oscillator-plus-noise model, relies on sustained speech signal recordings, and the training of oscillator model parameters from continuous speech signals does not seem feasible with the methods considered here.

The very reason also still prevents the application of the oscillator model to the modeling of transient speech sounds, like plosives.

For the goal of general purpose speech synthesis with a nonlinear oscillator also some additional tasks are necessary. The most important of these tasks is to have a technique for the control and modification of fundamental frequency at hand which can be linked into the nonlinear model. For this task methods based on the interpolation of nonlinear function models have been proposed [Man99, LMM00]. As for other speech synthesis algorithms, the task of fundamental frequency modifications shall be easier to solve in the 'glottal signal' domain than on the full speech signal, as, for example, already pointed out by the better performance of LP-PSOLA as compared to PSOLA on the full speech signal. The example in sect. 4.5.1, however, shows that a robust control of fundamental frequency using RBF weight interpolation is still challenging. A method that aims at a decoupling of fundamental waveform shape and fundamental frequency has been proposed in [DA00], and applied to the modeling of glottal flow signals from inverse filtering: A second order resonator filter is coupled with nonlinear prediction, and the fundamental frequency is controlled by changing the resonance frequency. The application of this concept to the oscillator-plus-noise model is certainly worth to be investigated.

For inverse filtering, alternative methods to LP for modeling the vocal tract filter should be considered. Acoustic tube models, for example, are closely related to LP and would account for the incorporation of losses as well as sources along the vocal tract, or interaction between source and vocal tract filter. Acoustic tube modeling of the vocal tract filter is a step in the direction of physical modeling, in the manner of articulatory synthesis. A combination of an invertible physical model for the vocal tract and the oscillator model could yield a high quality 'quasi-articulatory' synthesis model.

The identification of oscillatory and noise-like signal components in the oscillator model learning method used here is not yet optimal. In the analysis of oscillator-plus-noise model generated speech signals both the jitter and the shimmer measure are somewhat too high. This might be due to the over-estimation of the amplitude of the noise-like signal component[2]. Since we consider the predictable part of a speech signal as the oscillatory component, the prediction error signal is the estimate for the noise-like signal component. The prediction error, however, also comprises modeling errors due to the application of a certain nonlinear function model. Thus, the estimate for the amplitude of the noise-like signal component based on the prediction error amplitude might be too high.

On the other hand the noise-like signal component lacks energy in the low frequency range (below 1 kHz). This is especially evident in the examples for re-synthesis of unvoiced fricatives given in App. C. Obviously, the low-pass filter in the inverse filtering process makes low frequency noise-like components predictable to some extent, hence they are attenuated in the prediction error signal, and thereupon also by the LP synthesis filter in the noise signal path. The predictability of low-frequency components seems to be accountable for the under-estimation of the power of low-pass filtered noise in artificial glottal signals reported in sect. 4.4.6 as well (cf. table 4.4).

For some mixed excitation signals the noise-like signal component dominates the spectral distribution, and hence has a large influence on the estimation of the LP filter for the oscillatory part from the full speech signal. Thus, the LP filter for the oscillatory signal component primarily models the spectral characteristics of the noise-like component, and may not result in an appropriate estimate for the 'glottal signal' associated with the oscillatory signal component. A remedy may be the iterative re-estimation of the inverse filter for the oscillatory component from a full speech signal with the current estimate for the noise-like component subtracted. This may also relieve the problem of predictability of the noise-like signal component in the 'glottal signal' of mixed excitation and unvoiced signals due to low-pass filtering.

[2] The somewhat too high amount of noise-like signal components in the synthetic signals is also confirmed by informal listening tests.

Appendix A

Abbreviations, acronyms, and mathematical symbols

A.1 Abbreviations and acronyms

ANN	artificial neural network
DC	direct current (signal)
DFT	discrete Fourier transform
FFT	fast Fourier transform
FIR	finite impulse response (filter)
GCI	glottis closure instant
GNE	glottal-to-noise excitation (ratio)
GRBF	Generalized Radial Basis Function (network)
HNR	harmonics-to-noise ratio
IIR	infinite impulse response (filter)
IPA	international phonetic alphabet
LP	linear prediction
LF	Liljencrants-Fant (model for voiced speech source)
MA	moving average (filter)
MARS	multivariate adaptive regression splines
MI	mutual information
MLP	multi-layer perceptron
MSE	mean-square error
RBF	radial basis function
RVM	relevance vector machine
pdf	probability density function
TTS	text-to-speech (system)

A.2 Mathematical symbols

F_0	fundamental frequency
T_0	fundamental period
n	discrete time
N	dimension
N_c	number of RBF centers
x	scalar (input) variable
$\hat{x}$	predicted value for x
$\{x(n)\}$	time series
$\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^{\mathsf{T}}$	vector (input) variable
$\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_P]$	matrix of input data for training
$\boldsymbol{t} = [t_1, t_2, \ldots, t_P]^{\mathsf{T}}$	vector of target (output) values
$\Vert \cdot \Vert$	Euclidean vector norm
$f(\cdot)$	a *nonlinear function*
$\hat{f}(\cdot)$	model for the nonlinear function
$\varphi(\cdot)$	radial basis function (RBF)
$\boldsymbol{\varphi}(\cdot) = [\varphi_1(\cdot), \varphi_2(\cdot), \ldots \varphi_{N_c}(\cdot)]^{\mathsf{T}}$	vector of RBF outputs
$\boldsymbol{\Phi} = \begin{bmatrix} \varphi_1(\boldsymbol{x}_1) & \cdots & \varphi_{N_c}(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{x}_P) & \cdots & \varphi_{N_c}(\boldsymbol{x}_P) \end{bmatrix}$	matrix of RBF outputs
$g(\cdot)$	nonlinear activation function (in MLP)
$\boldsymbol{w} = [w_1, w_2, \ldots w_{N_c}]^{\mathsf{T}}$	(true) weights of the RBF model
$\hat{\boldsymbol{w}} = [\hat{w}_1, \hat{w}_2, \ldots \hat{w}_{N_c}]^{\mathsf{T}}$	estimated weights of RBF model
λ	regularization factor for RBF training
$p(x)$	probability density function (pdf) of x
$p(x \mid y)$	conditional pdf of x given y
$\mathcal{N}(\mu, \sigma^2)$	Gaussian pdf with mean μ and variance σ^2
$E(x)$	expected value of x
$\mathcal{O}(N)$	computational complexity of order N
$H(z)$	z-domain system function of a filter
$A(z)$	system function of the LP inverse filter

Appendix B

Bayesian training of an RBF network

B.1 Derivation of the iterative algorithm

The Bayesian approach to model comparison and regularization for regression [Mac92a] assumes that the training output samples t_k are produced from the training input vectors $\boldsymbol{x}_k, k = 1 \dots P$ by an additive noise model

$$t_k = f(\boldsymbol{x}_k) + \epsilon_k \ , \quad p(\epsilon) = \mathcal{N}(0, \sigma_n^2) \ , \tag{B.1}$$

with additive zero-mean Gaussian noise samples ϵ_k with variance σ_n^2. σ_n^2 is the first *hyper-parameter* in the Bayesian model. For the application to RBF networks the function $f(\cdot)$ is modeled by the network function

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_c} w_i \, \varphi_i(\boldsymbol{x}) \ , \tag{B.2}$$

with a number of N_c weights w_i and basis functions $\varphi_i(\cdot)$. The weights are accumulated in a vector $\boldsymbol{w}$ later on. For the training algorithm the basis functions have to be fixed a priori, like, for example, the Gaussian basis functions used throughout this thesis: $\varphi_i(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\frac{\|\boldsymbol{x}-\boldsymbol{c}_i\|^2}{\sigma_g^2}\right)$, with a priori fixed centers $\boldsymbol{c}_i$ and variance σ_g^2.

The network weights $\boldsymbol{w}$ are considered random variables with a prior pdf $p(\boldsymbol{w}|\alpha)$, introducing one additional, scalar hyper-parameter α. Incorporating regularization by stating a preference for smoother network functions is done by choosing the prior for the weights as a Gaussian distribution with variance α^{-1}:

$$p(\boldsymbol{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{-\frac{N_c}{2}} \exp\left(-\frac{\alpha}{2}\|\boldsymbol{w}\|^2\right) \ , \tag{B.3}$$

thus preferring small values for the weights which result in a less variable network function.

The aim is to find the most probable values for the weights and the hyper-parameters given the training data, i. e., to maximize $p(\boldsymbol{w}, \alpha, \sigma_n^2|\boldsymbol{X}, \boldsymbol{t})$. Since this cannot be accomplished analytically, the task is divided into 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

$$p(\boldsymbol{w}, \alpha, \sigma_n^2|\boldsymbol{X}, \boldsymbol{t}) = p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{t}, \alpha, \sigma_n^2) \ p(\alpha, \sigma_n^2|\boldsymbol{X}, \boldsymbol{t}) \ . \tag{B.4}$$

For the first term on the right-hand side, the hyper-parameters are assumed to be known and the according posterior pdf for the weights $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t},\alpha,\sigma_n^2)$ is a multivariate Gaussian distribution[1],

$$p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t},\alpha,\sigma_n^2) = (2\pi)^{-\frac{N_c}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left(-\tfrac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{\mu})\right) \ , \tag{B.5}$$

with covariance and means, respectively:

$$\begin{aligned}\boldsymbol{\Sigma} &= (\frac{1}{\sigma_n^2}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} + \alpha\boldsymbol{I})^{-1} \ , \\ \boldsymbol{\mu} &= \frac{1}{\sigma_n^2}\boldsymbol{\Sigma}\,\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{t} \ .\end{aligned} \tag{B.6}$$

Here, $\boldsymbol{I}$ is the identity matrix of size N_c, and the $P \times N_c$ matrix $\boldsymbol{\Phi}$ is composed of the output of all basis functions $\varphi_i(\cdot)$ for all training input data vectors $\boldsymbol{x}_k, k = 1\ldots P$:

$$\boldsymbol{\Phi} = \begin{bmatrix} \varphi_1(\boldsymbol{x}_1) & \cdots & \varphi_{N_c}(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{x}_P) & \cdots & \varphi_{N_c}(\boldsymbol{x}_P) \end{bmatrix} \ . \tag{B.7}$$

The second part in the decomposition eq. B.4 is split according to

$$p(\alpha,\sigma_n^2|\boldsymbol{X},\boldsymbol{t}) \propto p(\boldsymbol{t}|\boldsymbol{X},\alpha,\sigma_n^2)\,p(\alpha)\,p(\sigma_n^2) \ . \tag{B.8}$$

Here, we neglect the normalization constant $p(\boldsymbol{t})$ 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 the choice of prior distributions for the hyper-parameters. Both α and σ_n^2 shall take only positive values. Moreover, they can be considered as scaling parameters. Thus, a uniform pdf on a logarithmic scale[2] is an appropriate choice. The prior pdf for α (and σ_n^2) thus should be

$$p(\alpha) = \begin{cases} \propto \frac{1}{\alpha} & \alpha > 0 \ , \\ 0 & \alpha \leq 0 \ . \end{cases} \tag{B.9}$$

This pdf cannot be normalized (because its integral diverges), however, for the application in the Bayesian algorithm this is not necessary, as we will see below. The uniform pdf on the

[1] Both eq. B.5 and eq. B.11 can be derived simultaneously from the identity $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t},\alpha,\sigma_n^2)\cdot p(\boldsymbol{t}|\boldsymbol{X},\alpha,\sigma_n^2) = p(\boldsymbol{t}|\boldsymbol{X},\boldsymbol{w},\alpha,\sigma_n^2)\cdot p(\boldsymbol{w}|\boldsymbol{X},\alpha,\sigma_n^2)$, with $p(\boldsymbol{w}|\boldsymbol{X},\alpha,\sigma_n^2) = p(\boldsymbol{w}|\alpha)$ from eq. B.3, and $p(\boldsymbol{t}|\boldsymbol{X},\boldsymbol{w},\alpha,\sigma_n^2) = p(\boldsymbol{t}|\boldsymbol{X},\boldsymbol{w},\sigma_n^2) = (2\pi\sigma_n^2)^{-P/2}\exp\left(-\frac{1}{2\sigma_n^2}\|\boldsymbol{t}-\boldsymbol{\Phi}\boldsymbol{w}\|^2\right)$, according to eqs. B.1 and B.2, using eq. B.7. The resulting product pdf can be separated in the pdf for the weights $\boldsymbol{w}$, eq. B.5, and in the pdf for the training output data $\boldsymbol{t}$, eq. B.11 which comprises all terms independent of $\boldsymbol{w}$.

[2] A more general choice for the pdf of the hyper-parameters is the Gamma distribution

$$p(\alpha) = \begin{cases} 0 & \text{for} \quad \alpha \leq 0 \ , \\ \dfrac{\lambda^n}{\Gamma(n)}\alpha^{n-1}e^{-\lambda\alpha} & \text{for} \quad \alpha > 0 \ , \quad \lambda, n > 0 \ , \end{cases}$$
$$\text{with} \quad \Gamma(n) = \int_0^\infty u^{n-1}e^{-u}du \ ,$$

which approaches a uniform distribution on a logarithmic scale in the case $\lambda, n \to 0$. In [Tip01] both choices are treated for the derivation of the relevance vector machine, here we stay with the special case of the uniform pdf on a logarithmic scale.

logarithmic scale has the advantage of making the learning process independent of the output training data scale [Tip01], see also App. B.2 below.

It is convenient to maximize the logarithm of the pdf in eq. B.8, and with respect to $\log\alpha$ and $\log\sigma_n^2$, which is treated assuming uniform distributions on a logarithmic scale or Gamma distributions as prior pdfs for the hyper-parameters in [Tip01] (for the relevance vector machine). We maximize

$$\log p(\boldsymbol{t}|\boldsymbol{X}, \log\alpha, \log\sigma_n^2) + \log p(\log\alpha) + \log p(\log\sigma_n^2) \ . \tag{B.10}$$

Due to $p(\log\alpha) = \alpha\, p(\alpha)$ the last two terms are constant for uniform prior distributions on a logarithmic scale.

$p(\boldsymbol{t}|\boldsymbol{X}, \alpha, \sigma_n^2)$ is a P-dimensional Gaussian distribution with zero means and a covariance matrix $\sigma_n^2 \boldsymbol{I} + \alpha^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^\mathsf{T}$:

$$p(\boldsymbol{t}|\boldsymbol{X}, \log\alpha, \log\sigma_n^2) = (2\pi)^{-\frac{P}{2}} |\sigma_n^2\boldsymbol{I} + \alpha^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^\mathsf{T}|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}\,\boldsymbol{t}^\mathsf{T}(\sigma_n^2\boldsymbol{I} + \alpha^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^\mathsf{T})^{-1}\boldsymbol{t}\right) \ . \tag{B.11}$$

Without the constant terms in eq. B.10 the log objective function for this case is

$$\mathcal{L} = -\frac{1}{2}\left(\log|\sigma_n^2\boldsymbol{I} + \alpha^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^\mathsf{T}| + \boldsymbol{t}^\mathsf{T}(\sigma_n^2\boldsymbol{I} + \alpha^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^\mathsf{T})^{-1}\boldsymbol{t}\right) \ . \tag{B.12}$$

Note that, in eq. B.11 and B.12, $\boldsymbol{I}$ is the $P \times P$ identity matrix.

From the derivative of the objective function with respect to the hyper-parameters the following equations at the maximum are found:

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\log\alpha} &= \frac{1}{2}\left(N_c - \alpha(\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\mu} + \mathrm{Trace}(\boldsymbol{\Sigma}))\right) = 0 \ , \\
\frac{\partial\mathcal{L}}{\partial\log\sigma_n^{-2}} &= \frac{1}{2}\left(P\,\sigma_n^2 - \|\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 - \mathrm{Trace}(\boldsymbol{\Sigma}\,\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi})\right) = 0 \ .
\end{aligned} \tag{B.13}$$

Appropriate update equations for the hyper-parameters derived from eq. B.13 [Mac92a] are

$$\begin{aligned}
\alpha^{\mathrm{new}} &= \frac{\gamma}{\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\mu}} \ , \\
\frac{1}{\sigma_n^2}^{\mathrm{new}} &= \frac{\|\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2}{P - \gamma} \ , \\
\gamma &= N_c - \alpha^{\mathrm{old}}\ \mathrm{Trace}(\boldsymbol{\Sigma}) \ .
\end{aligned} \tag{B.14}$$

The value of the intermediate parameter $\gamma \in [0, N_c]$ introduced here is a measure of the effective number of model parameters that are well determined [Mac92a].

When iterations of eq. B.6 and B.14 converge we find the Bayesian approximation for the function $f(\cdot)$ in eq. B.1 by taking the means $\boldsymbol{\mu}$ of the posterior distribution for the weights – i. e., the values maximizing eq. B.5, as the weights $\hat{\boldsymbol{w}} = \boldsymbol{\mu}$ for an RBF network with the same basis functions φ_i as used in the model (eq. B.2) and in the training process (eq. B.7):

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N_c} \hat{w}_i\,\varphi_i(\boldsymbol{x}) \ . \tag{B.15}$$

There are two approximations deviating from the exact Bayesian approach made in this procedure: Firstly, for the computation of the parameters for the weights posterior pdf (eq. B.5) in eq. B.6 it is assumed that the hyper-parameters are known, which corresponds to taking a delta-distribution for $p(\alpha, \sigma_n^2|\boldsymbol{X}, \boldsymbol{t})$. This is, however, a common assumption used, e. g., in the expectation-maximization (EM) algorithm [DLR77] (for an overview see [Moo96]). Secondly, the update of the hyper-parameters eq. B.14 is not an exact maximization of probability/likelihood, since the calculation of γ involves the old value for α, as well as $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ (which depend on α and σ_n^2) on the right-hand side of the equation. Nevertheless, this approximation turns out to be advantageous as compared to, e. g., assuming a Gaussian distribution for $p(\boldsymbol{w}, \alpha, \sigma_n^2 | \boldsymbol{X}, \boldsymbol{t})$ [Mac99]. Furthermore, it yields no problems with convergence. For RVM learning, even on the contrary, convergence is reported to be faster using an equivalent approximation as eq. B.14 than for using the exact maximization of the log objective function [Tip01, App.].

B.2 Scale-invariance for uniform prior pdfs on logarithmic scale

For the Bayesian learning of regularization factor and noise variance we stated a prior pdf (eq. B.9) for the hyper-parameter α (and σ_n^2) as

$$p(\alpha) = \begin{cases} \propto \frac{1}{\alpha} & \alpha > 0 \ , \\ 0 & \alpha \leq 0 \ . \end{cases} \tag{B.16}$$

This is a uniform distribution on a logarithmic scale. It cannot be normalized – however, this fact does not hinder its application in the Bayesian model. The uniform distribution on a logarithmic scale makes the Bayesian training process invariant to a linear scaling of training output data and of noise amplitude, as we will show below.

Recall that α is the inverse of the *variance* of the prior Gaussian pdf for the weights (eq. 3.18/eq. B.3), and σ_n^2 is the variance of the additive Gaussian noise (eq. 3.17/eq. B.1). To achieve scale invariance, amplitude scaling of the training output values should result in the corresponding scaling of the model weights, without changing model probability. Thus, the *standard deviation* $\sigma_w = \sqrt{1/\alpha}$ for the weights prior pdf should be able to take values in the range of, e. g., $[1, 2]$ equally likely as in $[10, 20]$ or $[0.1, 0.2]$.

Considering the positive root, the pdf of σ_w is obtained as

$$p(\sigma_w) \begin{cases} \propto \frac{1}{|-\frac{1}{2}\alpha^{-3/2}|} \frac{1}{\alpha} = 2\sqrt{\alpha} = \frac{2}{\sigma_w} \ , & \sigma_w > 0 \ , \\ 0 \ , & \sigma_w \leq 0 \ . \end{cases} \tag{B.17}$$

The probability for the weights standard deviation being in $[a, b]$, with $0 < a < b < \infty$ is

$$P(a < \sigma_w < b) = \int_a^b p(\sigma'_w) \, d\sigma'_w \propto \int_a^b \frac{2}{\sigma'_w} \, d\sigma'_w = \left[2 \ln |\sigma'_w| \right]_a^b = 2 \ln \frac{b}{a} \ , \tag{B.18}$$

which is the same for $[1, 2]$, $[10, 20]$ or $[0.1, 0.2]$. Thus, the prior probability for the weights standard deviation taking values in either of these intervals is equal, and the Bayesian weight values are able to scale with the training output signal.

The same arguing is used for the standard deviation of the additive noise signal σ_n and proves that the Bayesian training process is also robust with respect to scaling of the noise amplitude.

Appendix C

Example signals

On the following pages some examples for the signals in the oscillator-plus-noise model (fig. 5.14 on page 106) are depicted. The plots for each signal comprise the time domain signal, the two-dimensional phase-space representation, as well as the DFT magnitude spectrum.

All signals are shown at a sampling rate of 16 kHz, the embedding lag for the phase-space representation is $M = 13$ samples, and in the phase-space representation the same 600 samples as in the time-domain plot are used. For all synthetic signals both the time-domain signal and the phase-space representation comprises an initial transient due to the initialization of the oscillator by $(N-1)M+1 = 40$ samples from the original signal $x_g(n)$, plotted in dotted lines[1], as well as additional transient signal components due to the synthesis filters $1/H_{\mathrm{lp}}(z)$, $1/A(z)$, and $1/A_{\mathrm{noi}}(z)$.

DFT spectra are computed from a number of 1024 signal samples, where the first 200 samples including the transient signal part for the synthetic signals are skipped. The Hann window function is used.

In the time-domain signal plots of the residual $x_{r\_\mathrm{noi}}(n)$ of the noise-like signal component, the estimated noise amplitude trajectory $\hat{a}_{\mathrm{noi}}(n)$ is plotted in dashed lines, and in the plots for the modulated synthetic noise signal $y_{r\_\mathrm{noi}}(n)$ the predicted noise amplitude $\tilde{a}_{\mathrm{noi}}(n)$ is plotted in dashed lines. Both $\hat{a}_{\mathrm{noi}}(n)$ and $\tilde{a}_{\mathrm{noi}}(n)$ are exaggerated by a factor of 5.

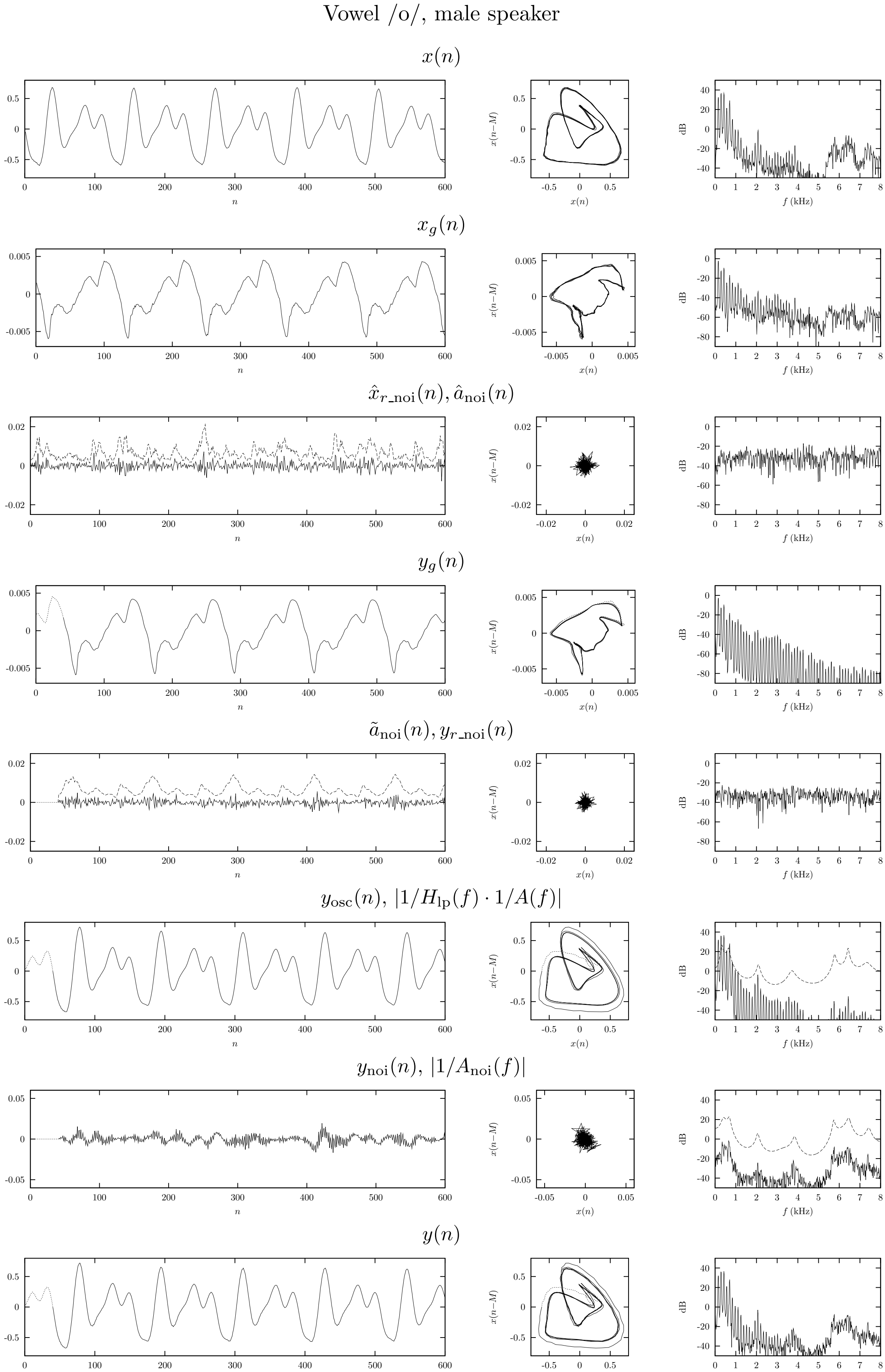
In the DFT spectra of the synthetic oscillatory speech component $y_{\mathrm{osc}}(n)$ and of the synthetic noise-like component $y_{\mathrm{noi}}(n)$ the transfer functions of the according synthesis filters – $|1/H_{\mathrm{lp}}(f) \cdot 1/A(f)|$ and $|A_{\mathrm{noi}}(f)|$, respectively – are plotted in dashed lines.

Inverse filtering is performed with LP of order $N_{\mathrm{LP}} = 18$, pre-emphasis with $k_{\mathrm{em}} = 0.75$, and a one-pole recursive low-pass $H_{\mathrm{lp}}(z)$ with a pole at $k_{\mathrm{lp}} = 0.95$.

The parameters for the oscillator model are: embedding dimension $N = 4$, embedding delay $M = 13$, Gaussian RBF network with $K = 5$ centers per dimension ($N_c = 625$ basis functions), outer edge of center grid $D = 1$, basis function width $d_{\mathrm{BF}} = 1$ (variance of Gaussian basis functions $\sigma_g^2 = 0.5^2$). RBF weights are learned by the Bayesian training algorithm. For the noise amplitude prediction the same network structure as for signal prediction is used, but training is performed with a fixed regularization factor $\lambda = 0.1$ using regularized matrix inversion. A number of $P = 3000$ signal samples were used for training.

The audio files for the example signals presented here are available from my web-page at http://www.spsc.tugraz.at/erhard/publ/diss.html.

[1] The 40 samples for initialization of $y_g(n)$ are from the beginning of $x_g(n)$, right before the signal part used for training and shown in the plots here.



Vowel /o/, male speaker

$x(n)$

$x_g(n)$

$\hat{x}_{r\_\mathrm{noi}}(n), \hat{a}_{\mathrm{noi}}(n)$

$y_g(n)$

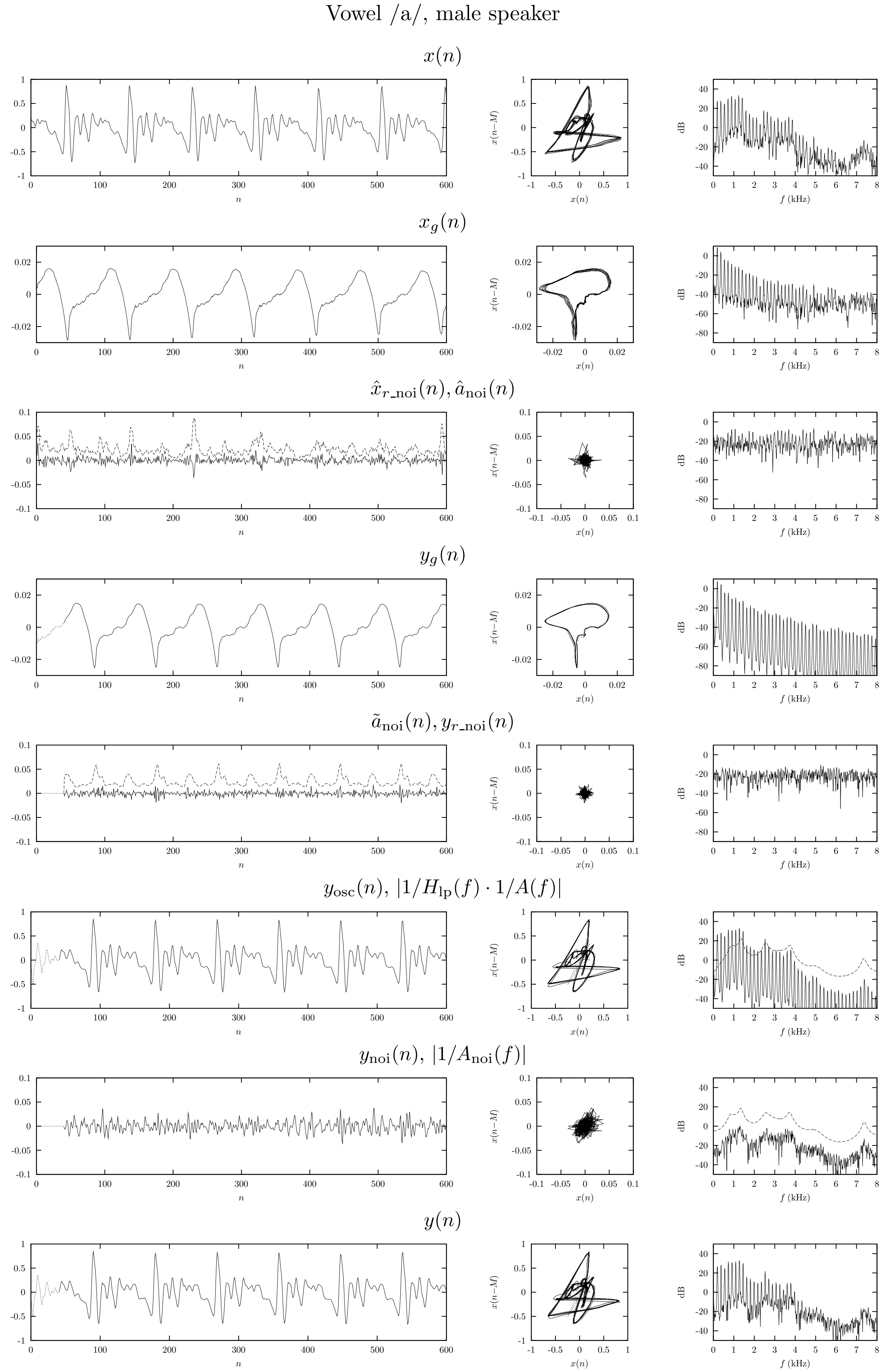
$\tilde{a}_{\mathrm{noi}}(n), y_{r\_\mathrm{noi}}(n)$

$y_{\mathrm{osc}}(n)$, $|1/H_{\mathrm{lp}}(f) \cdot 1/A(f)|$

$y_{\mathrm{noi}}(n)$, $|1/A_{\mathrm{noi}}(f)|$

$y(n)$

Figure C.1: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.



Vowel /a/, male speaker

$x(n)$

$x_g(n)$

$\hat{x}_{r\_\mathrm{noi}}(n), \hat{a}_{\mathrm{noi}}(n)$

$y_g(n)$

$\tilde{a}_{\mathrm{noi}}(n), y_{r\_\mathrm{noi}}(n)$

$y_{\mathrm{osc}}(n)$, $|1/H_{\mathrm{lp}}(f) \cdot 1/A(f)|$

$y_{\mathrm{noi}}(n)$, $|1/A_{\mathrm{noi}}(f)|$

$y(n)$

Figure C.2: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.

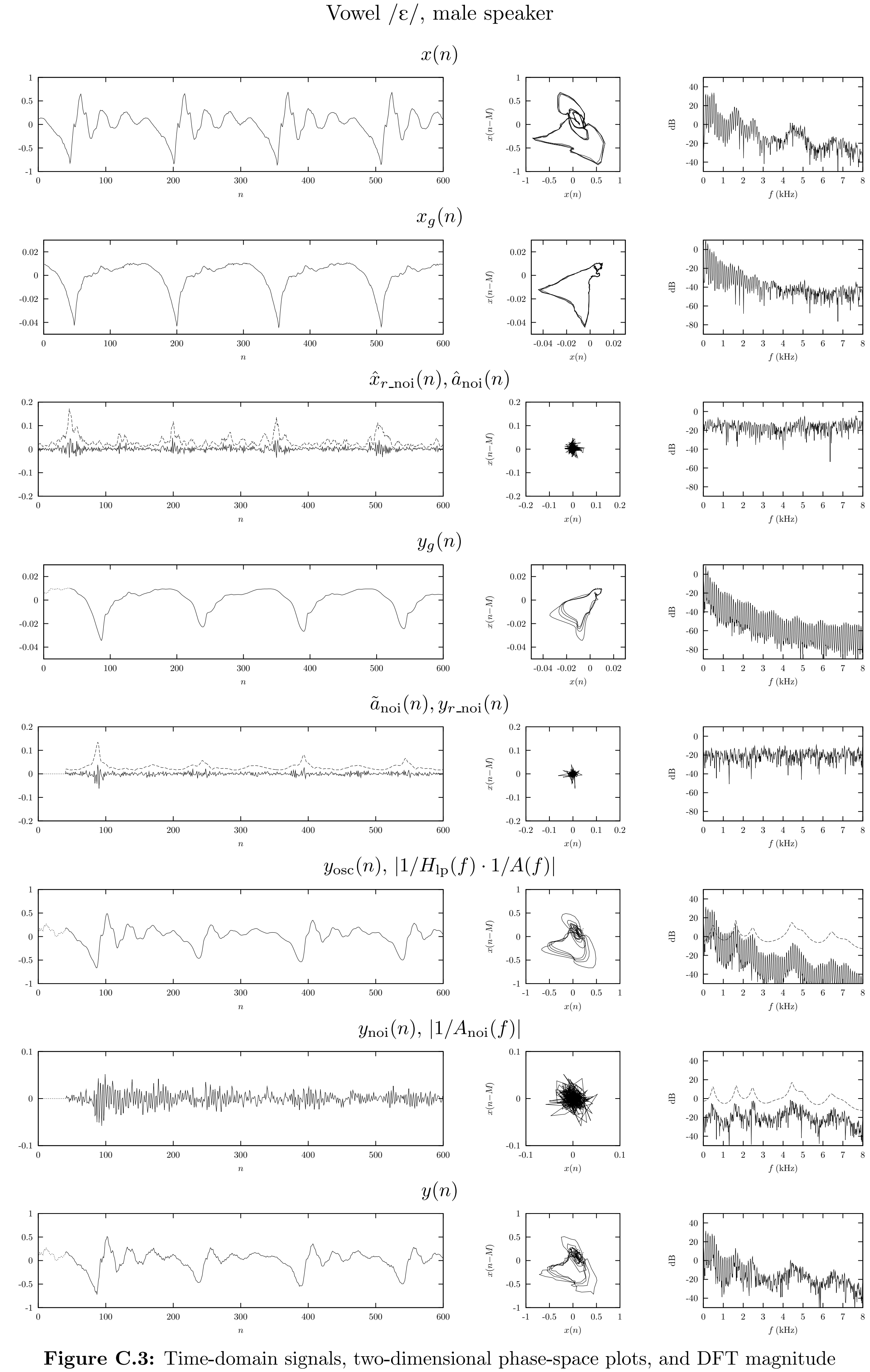
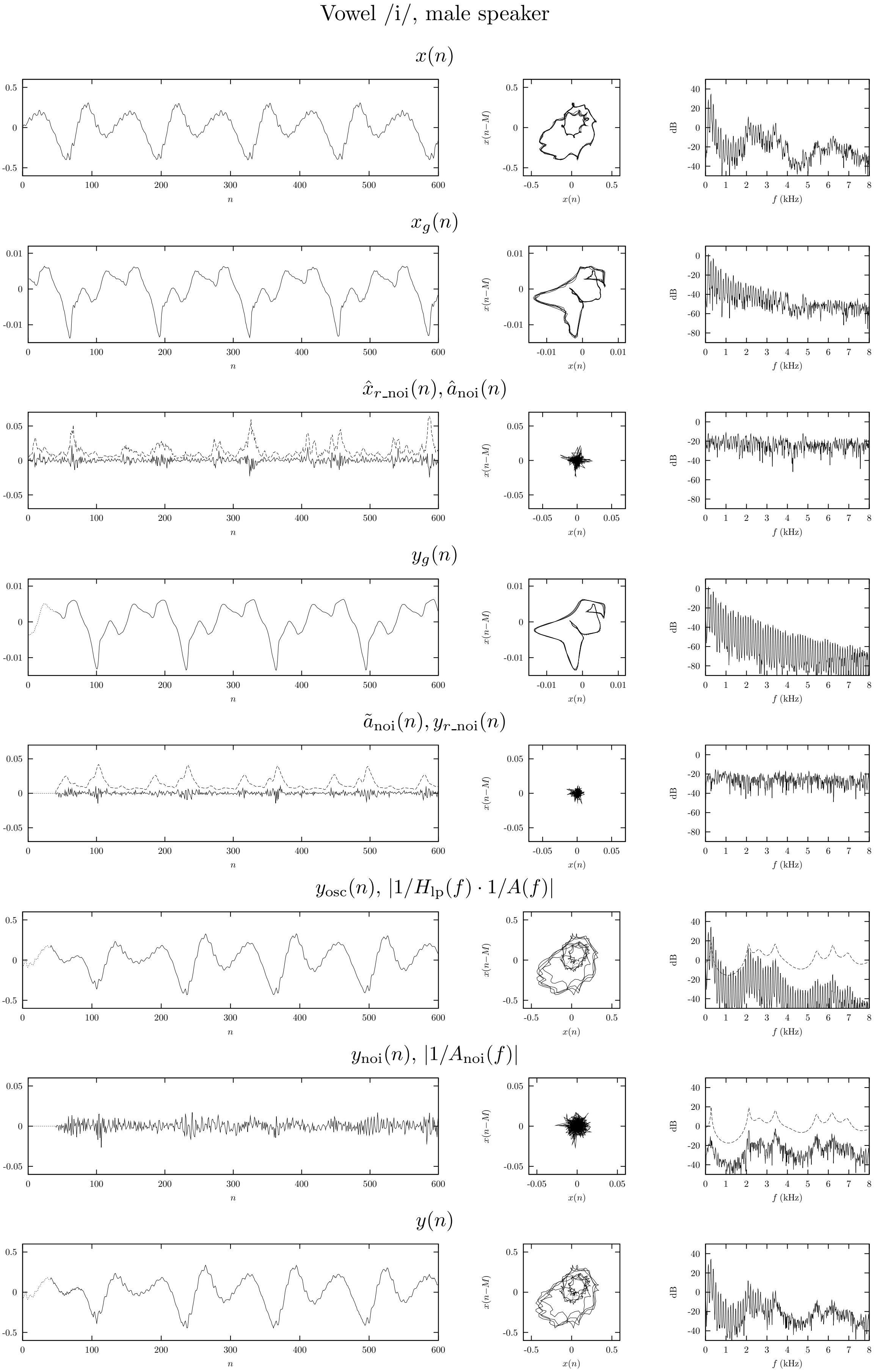


Figure C.3: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.


Vowel /i/, male speaker

$x(n)$

$x_g(n)$

$\hat{x}_{r\_\mathrm{noi}}(n), \hat{a}_{\mathrm{noi}}(n)$

$y_g(n)$

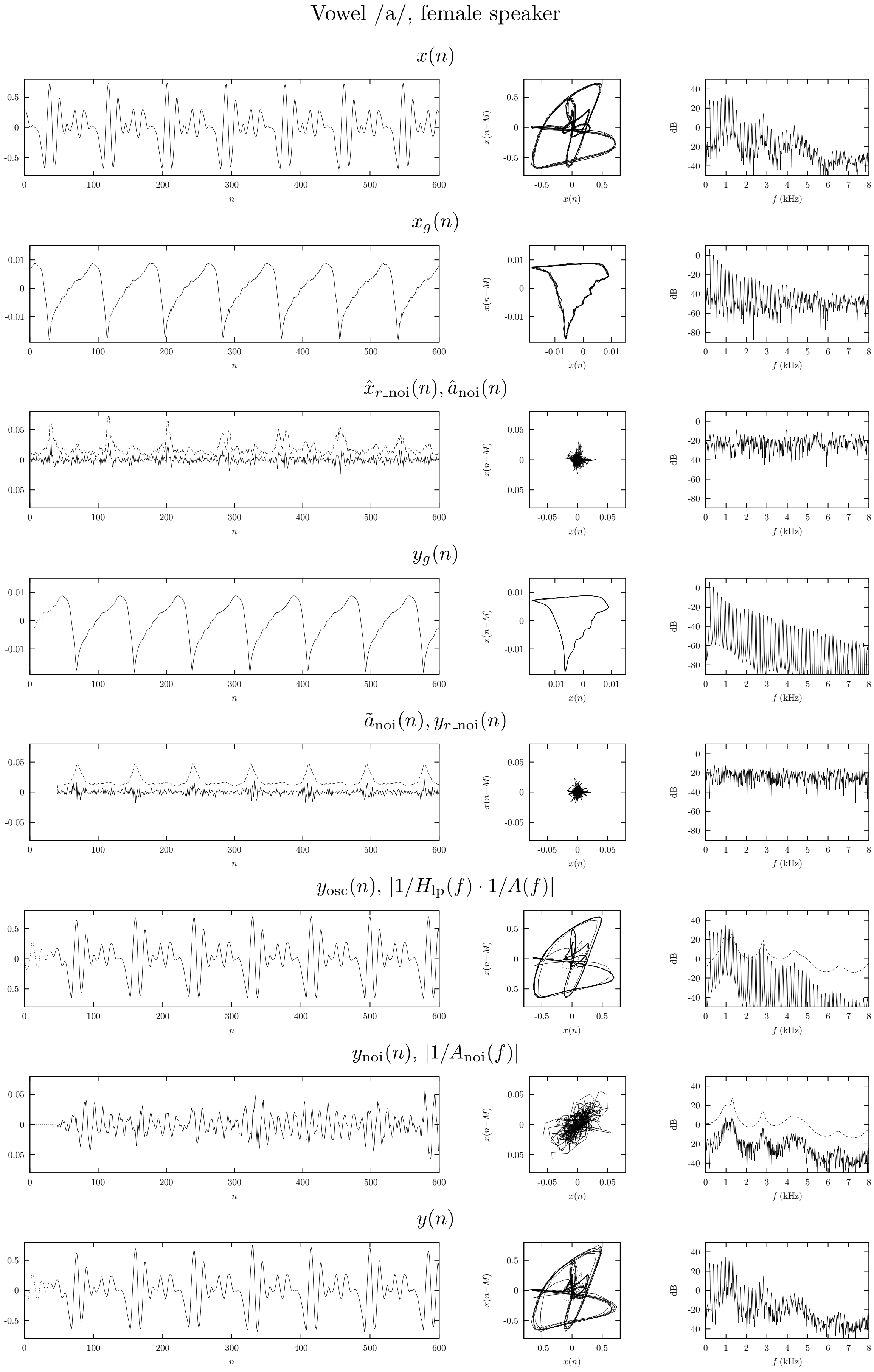
$\tilde{a}_{\mathrm{noi}}(n), y_{r\_\mathrm{noi}}(n)$

$y_{\mathrm{osc}}(n)$, $|1/H_{\mathrm{lp}}(f) \cdot 1/A(f)|$

$y_{\mathrm{noi}}(n)$, $|1/A_{\mathrm{noi}}(f)|$

$y(n)$

Figure C.4: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.



Vowel /a/, female speaker

$x(n)$

$x_g(n)$

$\hat{x}_{r\_\mathrm{noi}}(n), \hat{a}_{\mathrm{noi}}(n)$

$y_g(n)$

$\tilde{a}_{\mathrm{noi}}(n), y_{r\_\mathrm{noi}}(n)$

$y_{\mathrm{osc}}(n)$, $|1/H_{\mathrm{lp}}(f) \cdot 1/A(f)|$

$y_{\mathrm{noi}}(n)$, $|1/A_{\mathrm{noi}}(f)|$

$y(n)$

Figure C.5: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.

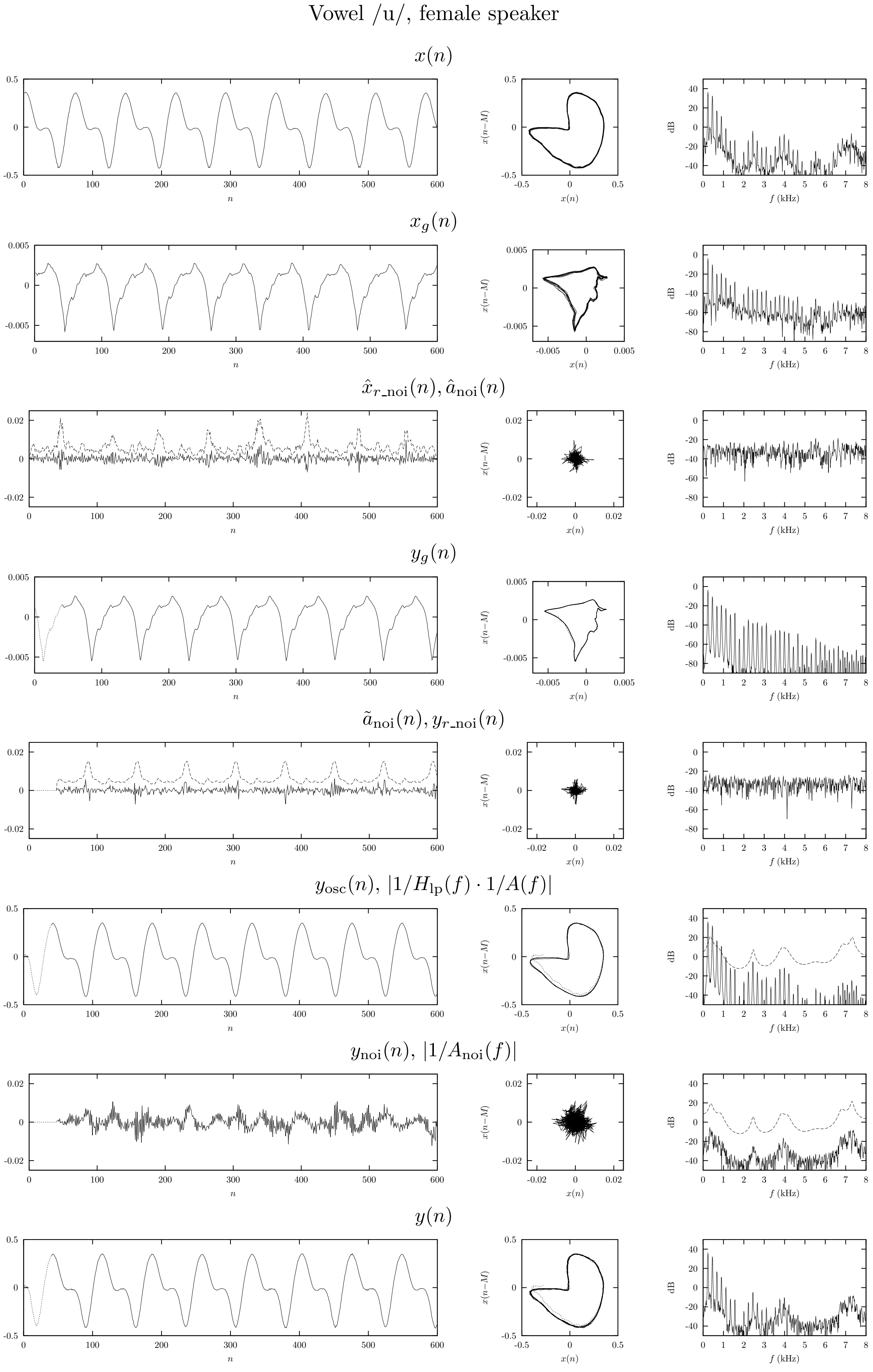


Figure C.6: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.

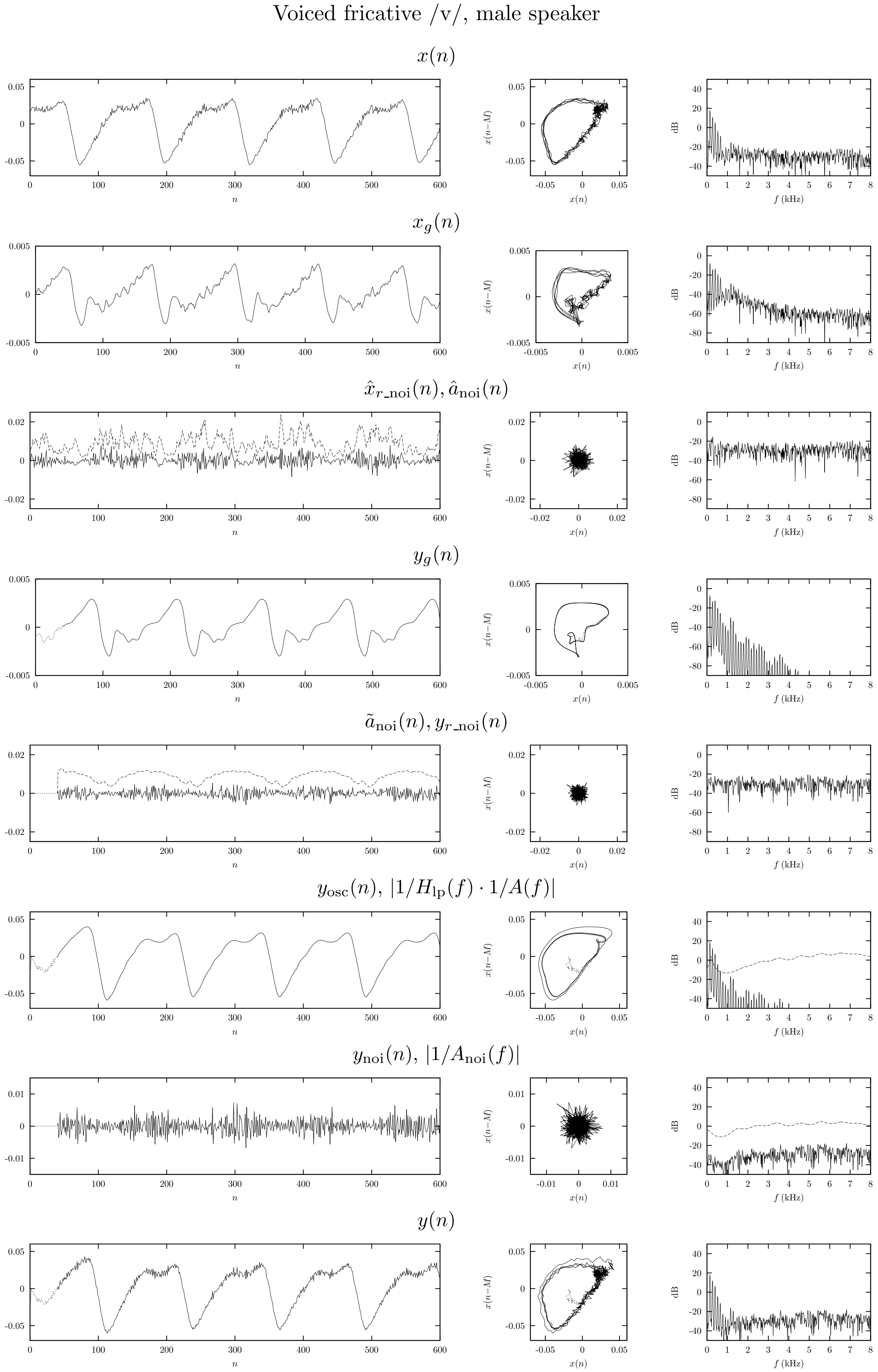
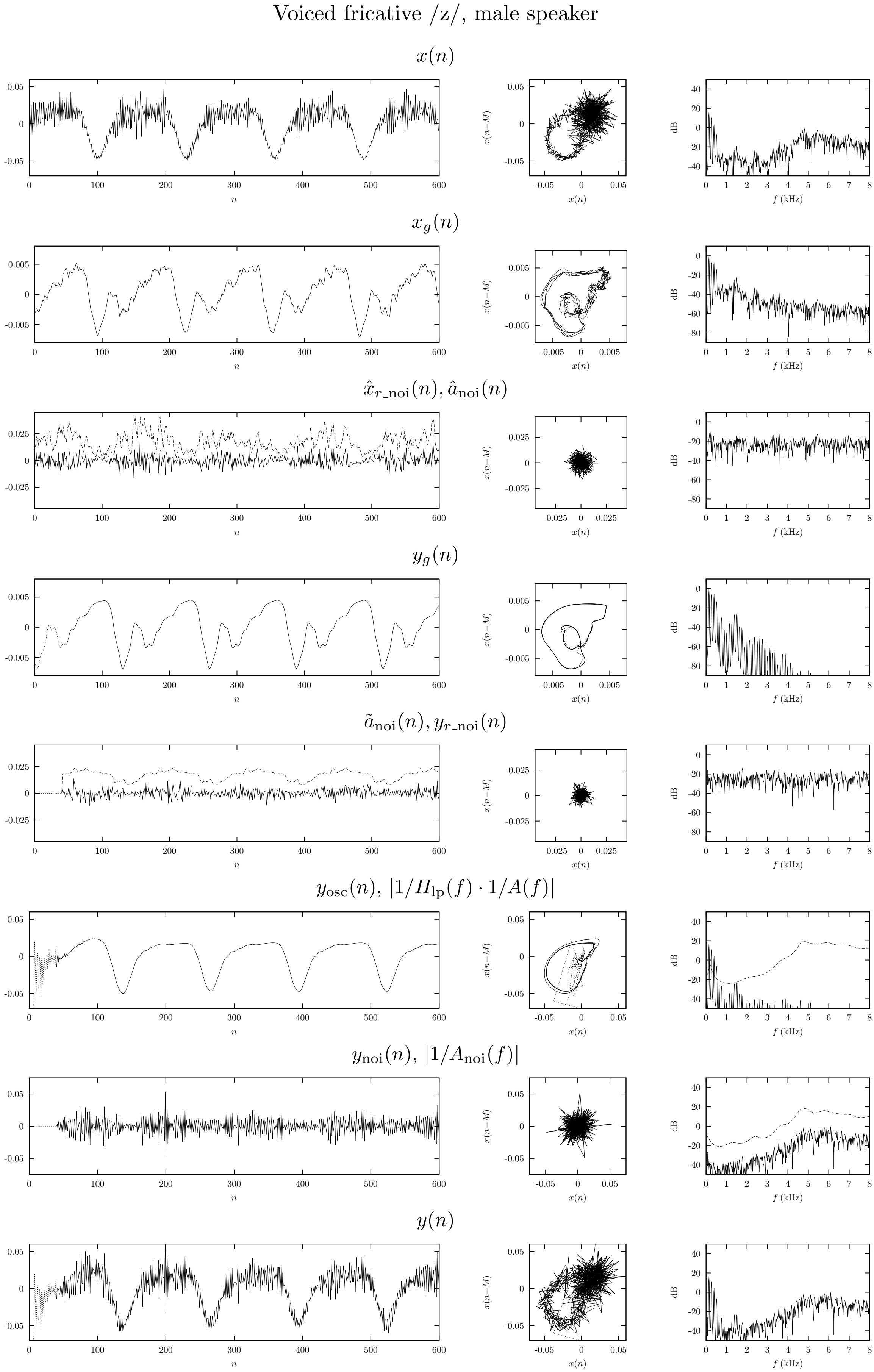


Figure C.7: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.



Voiced fricative /z/, male speaker

$x(n)$

$x_g(n)$

$\hat{x}_{r\_\mathrm{noi}}(n), \hat{a}_{\mathrm{noi}}(n)$

$y_g(n)$

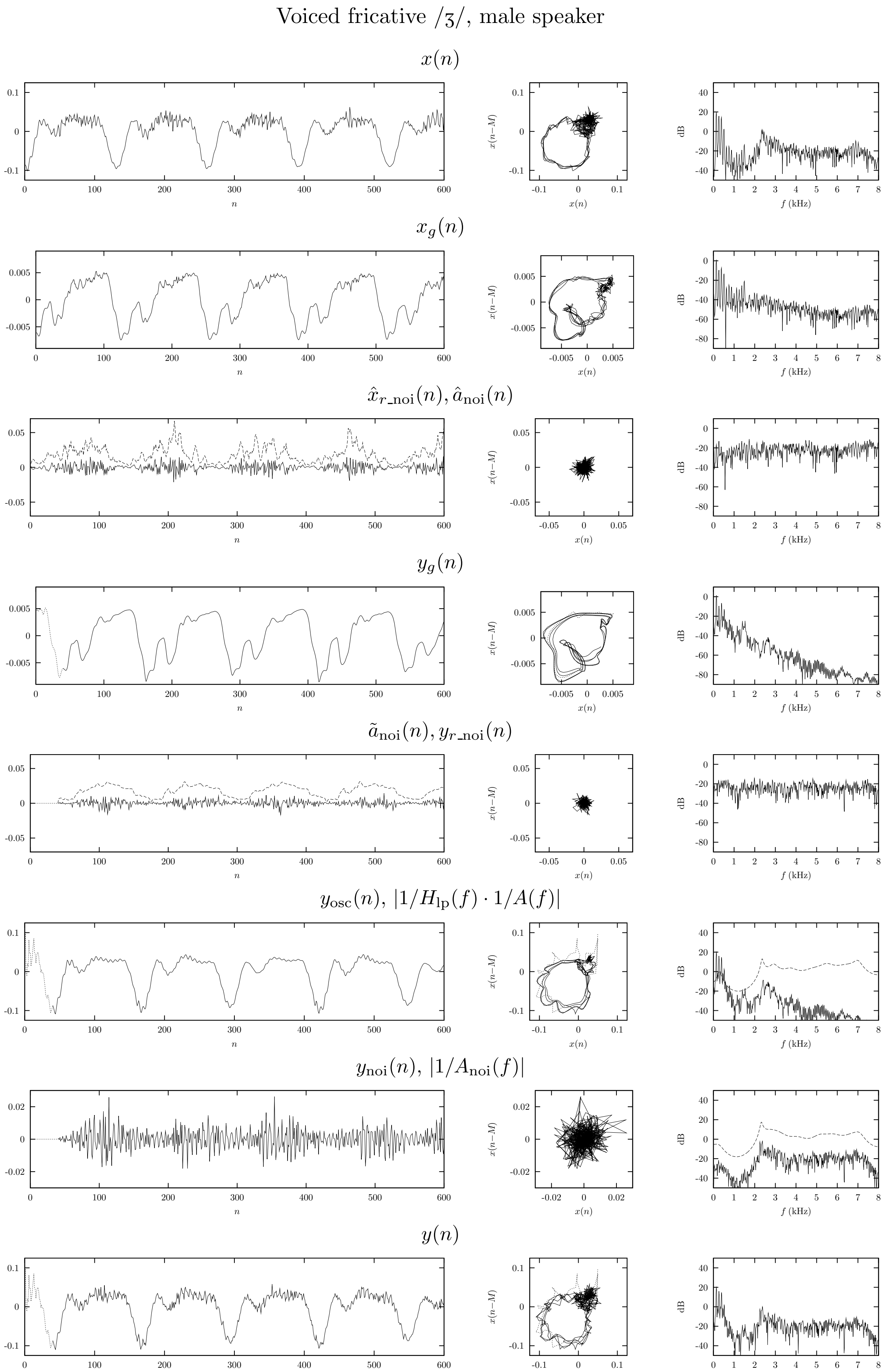
$\tilde{a}_{\mathrm{noi}}(n), y_{r\_\mathrm{noi}}(n)$

$y_{\mathrm{osc}}(n)$, $|1/H_{\mathrm{lp}}(f) \cdot 1/A(f)|$

$y_{\mathrm{noi}}(n)$, $|1/A_{\mathrm{noi}}(f)|$

$y(n)$

Figure C.8: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.



Voiced fricative /ʒ/, male speaker

$x(n)$

$x_g(n)$

$\hat{x}_{r\_\mathrm{noi}}(n), \hat{a}_{\mathrm{noi}}(n)$

$y_g(n)$

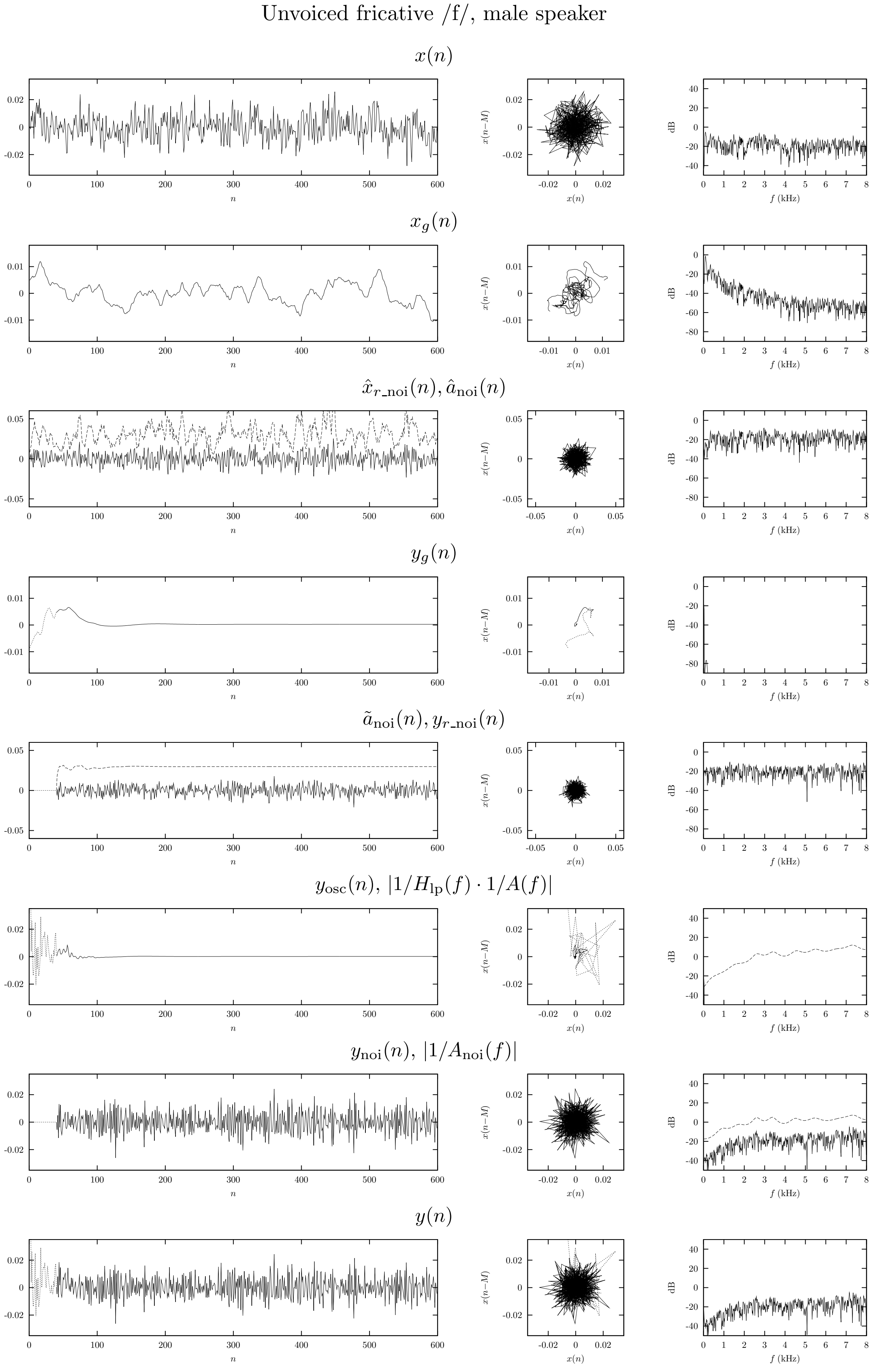
$\tilde{a}_{\mathrm{noi}}(n), y_{r\_\mathrm{noi}}(n)$

$y_{\mathrm{osc}}(n)$, $|1/H_{\mathrm{lp}}(f) \cdot 1/A(f)|$

$y_{\mathrm{noi}}(n)$, $|1/A_{\mathrm{noi}}(f)|$

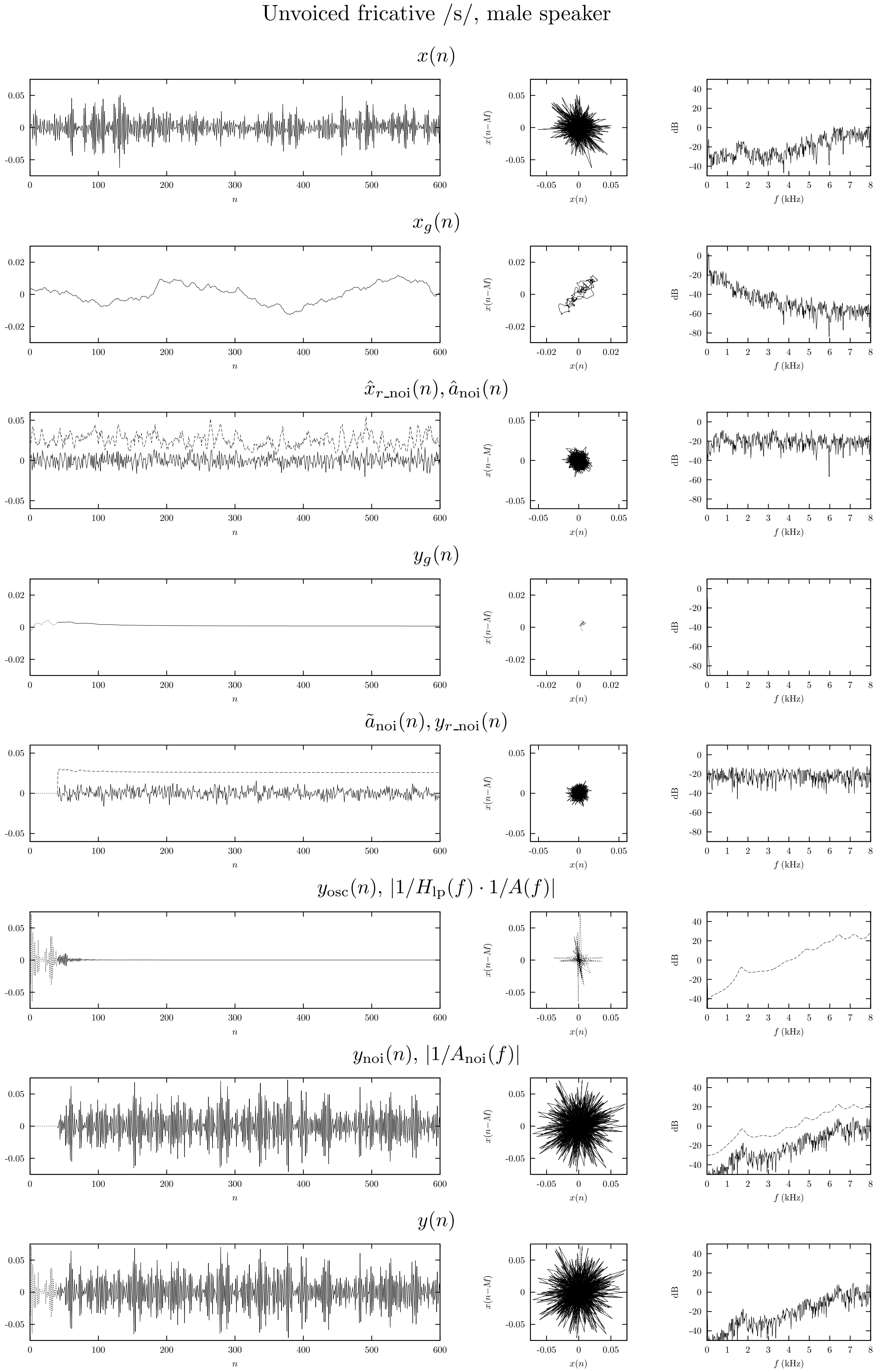
$y(n)$

Figure C.9: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.



Unvoiced fricative /f/, male speaker

Figure C.10: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.



Unvoiced fricative /s/, male speaker

$x(n)$

$x_g(n)$

$\hat{x}_{r\_\mathrm{noi}}(n), \hat{a}_{\mathrm{noi}}(n)$

$y_g(n)$

$\tilde{a}_{\mathrm{noi}}(n), y_{r\_\mathrm{noi}}(n)$

$y_{\mathrm{osc}}(n)$, $|1/H_{\mathrm{lp}}(f) \cdot 1/A(f)|$

$y_{\mathrm{noi}}(n)$, $|1/A_{\mathrm{noi}}(f)|$

$y(n)$

Figure C.11: Time-domain signals, two-dimensional phase-space plots, and DFT magnitude spectra for the input, intermediate, and output signals in the oscillator-plus-noise model.

Bibliography

[ABST93] Henry D.I. Abarbanel, Reggie Brown, John J. Sidorowich, and Lev Sh. Tsimring. The analysis of observed chaotic data in physical systems. *Reviews of Modern Physics*, 65(4):1331–1392, 1993.

[AI96] Naofumi Aoke and Tohruh Ifukube. Two 1/f fluctuations in sustained phonation and their roles on naturalness of synthetic voice. In *Third IEEE International Conference on Electronics, Circuits, and Systems*, volume 1, pages 311–314, 1996.

[AKM98] Rashid Ansari, Dan Kahn, and Marian J. Macchi. Pitch modification of speech using a low-sensitivity inverse filter approach. *IEEE Signal Processing Letters*, 5(3):60–62, March 1998.

[Alk92] Paavo Alku. Glottal wave analysis with pitch synchronous iterative adaptive inverse filtering. *Speech Communication*, 11(2–3):109–118, 1992.

[Ans97] Rashid Ansari. Inverse filter approach to pitch modification: Application to concatenative synthesis of female speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 1623–1626, Munich, Germany, 1997.

[ARK$^+$99] Kai Alter, Erhard Rank, Sonja A. Kotz, Erdmut Pfeifer, Mireille Besson, Angela D. Friederici, and Johannes Matiasek. On the relations of semantic and acoustic properties of emotions. In *Proceedings of the International Congress of Phonetic Sciences*, San Francisco (CA), 1999.

[ARK$^+$00] Kai Alter, Erhard Rank, Sonja A. Kotz, Ulrike Toepel, Mireille Besson, Annett Schirmer, and Angela D. Friederici. Accentuation and emotions - two different systems? In *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*, Belfast, Northern Ireland, September 2000.

[ARK$^+$03] Kai Alter, Erhard Rank, Sonja A. Kotz, Ulrike Toepel, Mireille Besson, Annett Schirmer, and Angela D. Friederici. Affective encoding in the speech signal and in event-related brain potentials. *Speech Communication*, 40(1–2):61–70, April 2003.

[Ata83] Bishnu S. Atal. Efficient coding of LPC parameters by temporal decomposition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 81–84, Boston, 1983.

[Bai97] Gérard Bailly. Learning to speak. Sensori-motor control of speech movements. *Speech Communication*, 22(2-3):251–267, 1997.

[Bai02a] Gérard Bailly. The COST 258 signal generation test array. In Keller et al. [KBM$^+$02], pages 39–51.

[Bai02b] Gérard Bailly. A parametric harmonic+noise model. In Keller et al. [KBM$^+$02], pages 22–38.

[BBD01] Baris Bozkurt, Michel Bagein, and Thierry Dutoit. *From MBROLA to NU-MBROLA.* Multitel-TCTS Lab, Faculté Polytechnique de Mons, Belguim, 2001. http://www.cstr.ed.ac.uk/events/ssw4/papers/111.pdf.

[BBEO03] Pierre Badin, Gérard Bailly, Frédéric Elisei, and Matthias Odisio. Virtual Talking Heads and audiovisual articulatory synthesis. In *Proceedings of the International Congress of Phonetic Sciences*, pages 193–197, Barcelona, Spain, 2003.

[BBK97] Martin Birgmeier, Hans-Peter Bernhard, and Gernot Kubin. Nonlinear long-term prediction of speech signals. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1283–1286, Munich, Germany, 1997.

[BBMR00] Gérard Bailly, Eduardo Rodriguez Banga, Alex Monaghan, and Erhard Rank. The Cost258 signal generation test array. In *Proceedings of the Second International Conference on Language Resources and Evaluation LREC2000*, pages 651–654, Athens, Greece, 2000.

[BBRS98] Pierre Badin, Gérard Bailly, Monica Raybaudi, and Christoph Segebarth. A three-dimensional linear articulatory model based on MRI data. In *ESCA/COCOSDA Workshop on Speech Synthesis*, pages 249–254, Jenolan Caves, Australia, 1998.

[BC05] Alan W. Black and Rob Clark. The Festival speech synthesis system. Available from: http://www.cstr.ed.ac.uk/projects/festival. Last visited: Nov. 2005.

[BCS$^+$99] Mark Beutnagel, Alistair Conkie, Juergen Schroeter, Yannis Stylianou, and Ann Syrdal. The AT&T next-gen TTS system. In *Proceedings of the 137th Meeting of the Acoustical Society of America*, 1999. http://www.research.att.com/projects/-tts.

[BD99] Hans-Peter Bernhard and Georges A. Darbellay. Performance analysis of the mutual information function for nonlinear and linear signal processing. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1297–1300, 1999.

[Ber97] Hans-Peter Bernhard. *The Mutual Information Function and its Application to Signal Processing.* PhD thesis, Vienna University of Technology, 1997.

[Ber98] Hans-Peter Bernhard. A tight upper bound on the gain of linear and nonlinear predictors for stationary stochastic processes. *IEEE Transactions on Signal Processing*, 46:2909–2917, November 1998.

[BHY98] H. Timothy Bunnell, Steve R. Hoskins, and Debra Yarrington. Prosodic vs. segmental contributions to naturalness in a diphone synthesizer. In *Proceedings of the International Conference on Spoken Language Processing*, volume 5, pages 1723–1726, Sydney, 1998.

[Bir95] Martin Birgmeier. A fully Kalman-trained radial basis function network for nonlinear speech modeling. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 259–264, Perth, Australia, 1995.

[Bir96] Martin Birgmeier. *Kalman-Trained Neural Networks for Signal Processing Applications.* PhD thesis, Vienna University of Technology, 1996.

[BK91] Hans-Peter Bernhard and Gernot Kubin. Detection of chaotic behaviour in speech signals using Fraser's mutual information algorithm. In *Proceedings of 13th*

GRETSI Symposium on Signal and Image Processing, pages 1301–1311, Juan-les-Pins, France, September 1991.

[BK94] Hans Peter Bernhard and Gernot Kubin. A fast mutual information calculation algorithm. In M.J.J. Holt et al., editor, *Signal Processing VII: Theories and Applications*, volume 1, pages 50–53. Elsevier, Amsterdam, September 1994.

[BM94] Michael Banbrook and Stephen McLaughlin. Is speech chaotic?: Invariant geometrical measures for speech data. In *Proceedings IEE Colloquium on Exploiting Chaos in Signal Processing*, pages 8/1–8/10, June 1994.

[BM96] Michael Banbrook and Stephen McLaughlin. Dynamical modelling of vowel sounds as a synthesis tool. In *Proceedings of the International Conference on Spoken Language Processing*, volume 3, pages 1981–1984, Philadelphia, PA, 1996.

[BMM99] Michael Banbrook, Stephen McLaughlin, and Iain Mann. Speech characterization and synthesis by nonlinear methods. *IEEE Transactions on Speech and Audio Processing*, 7(1):1–17, January 1999.

[BT94] Alan W. Black and Paul A. Taylor. CHATR: A generic speech synthesis system. In *Proceedings of the International Conference on Computational Linguistics*, pages 983–986, 1994.

[BTC97] Alan W. Black, Paul Taylor, and Richard Caley. The Festival speech synthesis system: System documentation. Technical Report HCRC/TR-83, Human Computer Research Centre, University of Edinburgh, University of Edinburgh, Scotland, UK, 1997.

[CA83] Barbara E. Caspers and Bishnu S. Atal. Changing pitch and duration in LPC synthesized speech using multipulse excitation. *Journal of the Acoustic Society of America*, 73:S5, 1983.

[CA87] Barbara E. Caspers and Bishnu S. Atal. Role of multi-pulse excitation in synthesis of natural-sounding voiced speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 2388–2391, Dallas, Texas, 1987.

[Cam94] Nick Campbell. Prosody and the selection of units for concatenative synthesis. In *Proc. of ESCA/IEEE Workshop on Speech Synthesis*, pages 61–64, New York, USA, 1994.

[ČBC98] Jan Černocký, Geneviève Baudoin, and Gérard Chollet. Segmental vocoder – going beyond the phonetic approach. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Seattle, WA, 1998.

[CBM03] Adnène Cherif, Lamia Bouafif, and Mounir Mhamdi. Analysis of pathological voices by speech processing. In *Seventh International Symposium on Signal Processing and Its Applications*, volume 1, pages 365–367, July 2003.

[CC99] Jing-Dong Chen and Nick Campbell. Objective distance measures for assessing concatenative speech synthesis. In *Proceedings of the European Conference on Speech Communication and Technology*, volume 2, pages 611–614, 1999.

[Chi95] Donald G. Childers. Glottal source modeling for voice conversion. *Speech Communication*, 16:127–138, 1995.

[CHNV82] Leon O. Chua, Martin Hasler, Jacques Neirynck, and Philippe Verburgh. Dynamics of a piecewise-linear resonant circuit. *IEEE Transactions on Circuits and Systems*, 29(8):535–547, August 1982.

[CMU02a] Mark R. Cowper, Bernard Mulgrew, and Charles P. Unsworth. Nonlinear prediction of chaotic signals using a normalized radial basis function network. *Signal Processing*, 82:775–789, 2002.

[CMU02b] Mark R. Cowper, Bernard Mulgrew, and Charles P. Unsworth. Radial basis functions: Normilized or un-normalized? In *Proceedings of the European Signal Processing Conference*, volume I, pages 349–355, Toulouse, France, 2002.

[Coo93] Perry R. Cook. SPASM, a real-time vocal tract physical model controller; and singer, the companion software synthesis system. *Computer Music Journal*, 17(1):30–44, 1993.

[DA00] Carlo Drioli and Federico Avanzini. Model-based synthesis and transformation of voiced sounds. In *Proceedings of the Cost G-6 Conference on Digital Audio Effects*, pages 45–48, Verona, Italy, December 2000.

[DC97] Wen Ding and Nick Campbell. Optimising unit selection with voice source and formants in the CHATR speech synthesis system. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 537–540, Rhodes, Greece, 1997.

[Det83] Helmut Dettweiler. *Automatische Sprachsynthese deutscher Wörter mit Hilfe von silbenorientierten Segmenten.* PhD thesis, Technische Universität München, 1983.

[dK94] Guus de Krom. *Acoustic Correlates of Breathiness and Roughness.* PhD thesis, Utrecht University, Utrecht, 1994.

[DL93] Thierry Dutoit and H. Leich. MBR-PSOLA: Text-to-speech synthesis based on an MBE re-synthesis of the segments database. *Speech Communication*, 13:435–440, 1993.

[DLR77] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(2):1–38, 1977.

[dP92] Bert de Vries and José C. Príncipe. The gamma model - a new neural model for temporal processing. *Neural Networks*, 5(4):565–576, 1992.

[dPG00] Marcelo de Oliveira Rosa, José Carlos Pereira, and Marcos Grellet. Adaptive estimation of residue signal for voice pathology diagnosis. *IEEE Transactions on Biomedical Engineering*, 47(1):96–104, January 2000.

[EK96] Yasuo Endo and Hideki Kasuya. A stochastic model of fundamental period perturbation and its application to perception of pathological voice quality. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 772–775, Philadelphia (PA), 1996.

[EMPSB96] Samir El-Masri, Xavier Pelorson, Pierre Saguet, and Pierre Badin. Vocal tract acoustics using the transmission line matrix (TLM) method. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 953–956, 1996.

[Fan70] Gunnar Fant. *Acoustic Theory of Speech Production.* Mouton, The Hague, Paris, 1970.

[Fan95] Gunnar Fant. The LF-model revisited. Transformations and frequency domain analysis. In *Quarterly Progress Status Report*, number 2-3, pages 119–156. Speech Transmission Laboratory/Royal Institute of Technology, Stockholm, Sweden, 1995.

[FLL85] Gunnar Fant, Johan Liljencrants, and Qi-Guang Lin. A four parameter model of glottal flow. In *Quarterly Progress Status Report*, number 4, pages 1–13. Speech Transmission Laboratory/Royal Institute of Technology, Stockholm, Sweden, 1985.

[FMV97] Marcos Faúndez, Enric Monte, and Francesc Vallverdú. A comparative study between linear and nonlinear speech prediction. In José Mira, Roberto Moreno-Díaz, and Joan Cabestany, editors, *Biological and Artificial Computation: From Neuroscience to Technology*, volume 1240 of *Lecture Notes in Computer Science*, pages 1154–1163. Springer, 1997.

[FNK$^+$98] Attila Ferencz, István Nagy, Tünde-Csilla Kovács, Maria Ferencz, and Teodora Raţiu. The new version of the ROMVOX text-to-speech synthesis system based on a hybrid time domain-LPC synthesis technique. In *Proceedings of the International Conference on Spoken Language Processing*, volume 5, pages 1971–1974, Sydney, 1998.

[FNK$^+$99] Attila Ferencz, István Nagy, Tünde-Csilla Kovács, Teodora Raţiu, and Maria Ferencz. On a hybrid time domain-LPC technique for prosody superimposing used for speech synthesis. In *Proceedings of the European Conference on Speech Communication and Technology*, volume 4, pages 1831–1834, Budapest, 1999.

[Fra89] Andrew M. Fraser. Information and entropy in strage attractors. *IEEE Transactions on Information Theory*, 35(2):245–262, 1989.

[Fri91] Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–141, 1991.

[Hay94] Simon Haykin. *Neural Networks. A Comprehensive Foundation.* Macmillan College Publishing Company, New York, Toronto, Oxford, 1994.

[HB96] Andrew Hunt and Alan W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 373–376, Atlanta (GA), 1996.

[Hei82] Josef Heiler. Optimized frame selection for variable frame rate synthesis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 586–588, 1982.

[Her91] Dik J. Hermes. Synthesis of breathy vowels: Some research methods. *Speech Communication*, 10:497–502, 1991.

[Hes83] Wolfgang Hess. *Pitch Determination of Speech Signals.* Springer Series in Information Sciences. Springer, Berlin-Heidelberg-New York-Tokyo, 1983.

[HK98] Herbert Haas and Gernot Kubin. A multi-band nonlinear oscillator model for speech. In *Proceedings of the 32nd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 1998.

[HK05] Martin Hagmüller and Gernot Kubin. Poincaré sections for pitch mark determination. In *Proceedings of the ISCA Tutorial and Research Workshop on Non-Linear Speech Processing (NOLISP)*, pages 107–113, Barcelona, Spain, April 2005.

[HKM01] Rainer Hegger, Holger Kantz, and Lorenzo Matassini. Noise reduction for human speech signals by local projection in embedding spaces. *IEEE Transactions on Circuits and Systems*, 48(12):1454–1461, December 2001.

[HKS99] Rainer Hegger, Holger Kantz, and Thomas Schreiber. Practical implementation of nonlinear time series methods: The TISEAN package. *CHAOS*, 9:413–435, 1999.

[HM84] Michael L. Honig and David G. Messerschmitt. *Adaptive Filters: Structures, Algorithms, and Applications*. Kluwer Academic Publishers, Boston-The Hague-London-Lancaster, 1984.

[Hol81] Sverre Holm. 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, pages 118–120, Atlanta (GA), 1981.

[HP98a] John H.L. Hansen and Bryan L. Pellom. An effective quality evaluation protocol for speech enhancement algorithms. In *Proceedings of the International Conference on Spoken Language Processing*, volume 7, pages 2819–2822, Sydney, 1998.

[HP98b] Simon Haykin and José Príncipe. Making sense of a complex world. *IEEE Signal Processing Magazine*, 15(3):66–81, May 1998.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[HSW90] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5):551–560, 1990.

[ICI$^+$98] Akemi Iida, Nick Campbell, Soichiro Iga, Fumito Higuchi, and Michiaki Yasumura. Acoustic nature and perceptual testing of corpora of emotional speech. In *Proceedings of the International Conference on Spoken Language Processing*, volume 4, pages 1559–1562, Sydney, 1998.

[IF72] Kenzo Ishizaka and James L. Flanagan. Synthesis of voiced sounds from a two-mass model of the vocal cords. *Bell Systems Technical Journal*, 51:1233–1267, 1972.

[IGW$^+$03] Khalil Iskarous, Louis M. Goldstein, Douglas H. Whalen, Mark K. Tiede, and Philip E. Rubin. CASY: The Haskins configurable articulatory synthesizer. In *Proceedings of the International Congress of Phonetic Sciences*, pages 185–188, Barcelona, Spain, 2003.

[ISK$^+$72] Fumitada Itakura, Shuzo Saito, Tsunehiko Koike, Hiroshi Sawabe, and Masaaki Nishikawa. An audio response unit based on partial autocorrelation. *IEEE Transactions on Communications*, 20(4):792–797, August 1972.

[JM98] Kevin Judd and Alistair Mees. Embedding as a modeling problem. *Physica D*, 120(3-4):273–286, September 1998.

[JS00a] Philip J.B. Jackson and Christine H. Shadle. Aero-acoustic modelling of voiced and unvoiced fricatives based on MRI data. In *Proceedings of 5th Speech Production Seminar*, pages 185–188, Kloster Seeon, Germany, May 2000.

[JS00b] Philip J.B. Jackson and Christine H. Shadle. Frication noise modulated by voicing, as revealed by pitch-scaled decomposition. *Journal of the Acoustic Society of America*, 108(4):1421–1434, October 2000.

[JS01] Philip J.B. Jackson and Christine H. Shadle. Pitch-scaled estimation of simultaneous voiced and turbulence-noise components in speech. *IEEE Transactions on Speech and Audio Processing*, 9(7):713–726, October 2001.

[JSCK03] Matthias Jilka, Ann K. Syrdal, Alistair D. Conkie, and David A. Kapilow. Effects on TTS quality of methods of realizing natural prosodic variations. In *Proceedings of the International Congress of Phonetic Sciences*, pages 2549–2552, Barcelona, Spain, 2003.

[JZ02] Jack J. Jiang and Yu Zhang. Chaotic vibration induced by turbulent noise in a two-mass model of vocal folds. *Journal of the Acoustic Society of America*, 112(5):2127–2133, 2002.

[KAK93] Gernot Kubin, Bishnu S. Atal, and W. Bastiaan Kleijn. Performance of noise excitation for unvoiced speech. In *Proc. of IEEE Workshop on Speech Coding for Telecommunication*, pages 1–2, St.Jovite, Québec, Canada, October 1993.

[KBA92] Matthew B. Kennel, Reggie Brown, and Henri D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6):3403–3411, March 1992.

[KBM$^+$02] Eric Keller, Gérard Bailly, Alex Monaghan, Jack Terken, and Mick Huckvale, editors. *Improvements in Speech Synthesis, COST 258: The Naturalness of Synthetic Speech.* John Wiley & Sons, 2002.

[Kez95] Thomas Keznikl. Modifikation von Sprachsignalen für die Sprachsynthese, (Modification of speech signals for speech synthesis, in German). In *Fortschritte der Akustik—DAGA'95*, pages 983–986. Deutsche Gesellschaft für Akustik, Oldenburg, Germany, 1995.

[KH94] W. Bastiaan Kleijn and Jesper Haagen. Transformation and decomposition of the speech signal for coding. *IEEE Signal Processing Letters*, 1(9):136–138, 1994.

[KH95a] W. Bastiaan Kleijn and Jesper Haagen. A speech coder based on decomposition of characteristic waveforms. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 508–511, Detroit, Michigan, 1995.

[KH95b] W. Bastiaan Kleijn and Jesper Haagen. Waveform interpolation for coding and synthesis. In W. Bastiaan Kleijn and Kuldip K. Paliwal, editors, *Speech Coding and Synthesis*, pages 175–207. Elsevier, 1995.

[KK94] Gernot Kubin and W. Baastian Kleijn. 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, pages 453–456, Adelaide, South Australia, 1994.

[KKP03] Heinz Köppl, Gernot Kubin, and Gerhard Paoli. Bayesian methods for sparse RLS adaptive filters. In *Thirty-Seventh IEEE Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1273–1277, 2003.

[Kla80] Dennis Klatt. Software for a cascade/parallel formant synthesizer. *Journal of the Acoustic Society of America*, 67:971–995, 1980.

[KS95] Gudrun Klasmeyer and Walter F. Sendlmeier. Objective voice parameters to characterize the emotional content in speech. In *Proceedings of the International Congress of Phonetic Sciences*, Stockholm, Sweden, 1995.

[Kub86] Gernot Kubin. Wave digital filters: Voltage, current, or power waves? In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Tampa (FL), March 1986.

[Kub95] Gernot Kubin. Nonlinear processing of speech. In W. Bastiaan Kleijn and Kuldip K. Paliwal, editors, *Speech Coding and Synthesis*, pages 557–610. Elsevier, Amsterdam, 1995.

[Kub96a] Gernot Kubin. 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, pages 267–270, Atlanta (GA), 1996.

[Kub96b] Gernot Kubin. Voice processing—beyond the linear model. In *PRORISC/IEEE Workshop on Circuits, Systems and Signal Processing*, pages 393–400, Mierlo, NL, 1996.

[Kub98] Gernot Kubin. Signal analysis and speech processing. In A. Prochazka et al., editors, *Signal Analysis and Prediction*, pages 375–394. Birkhaeuser, Boston, 1998.

[KV98] Ester Klabbers and Raymond Veldhuis. On the reduction of concatenation artefacts in diphone synthesis. In *Proceedings of the International Conference on Spoken Language Processing*, volume 5, pages 1983–1986, Sydney, 1998.

[KV01] Ester Klabbers and Raymond Veldhuis. Reducing audible spectral discontinuities. *IEEE Transactions on Speech and Audio Processing*, 9(1):39–51, 2001.

[LHVH98] N.J.C. Lous, G.C.J. Hofmans, Raymond N. Veldhuis, and A. Hirschberg. A symmetrical two-mass vocal-fold model coupled to vocal tract and trachea, with application to prosthesis design. *Acta Acustica*, 84(5):1135–1150, 1998.

[LJP03] Andrew C. Lindgren, Michael T. Johnson, and Richard J. Povinelli. Speech recognition using reconstructed phase space features. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 66–63, Hong Kong, 2003.

[LJP04] Andrew C. Lindgren, Michael T. Johnson, and Richard J. Povinelli. 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, pages 533–536, Montreal, Quebec, Canada, 2004.

[LLW01] Henry Leung, Titus Lo, and Sichun Wang. Prediction of noisy chaotic time series using an optimal radial basis function neural network. *IEEE Transactions on Neural Networks*, 12(5):1163–1172, September 2001.

[LMM00] Douglas Leith, Iain Mann, and Steve McLaughlin. A nonlinear model to synthesize voiced speech with natural pitch variations. In *Fifth IMA International Conference on Mathematics in Signal Processing*, University of Warwick, UK, December 2000.

[LS01] Hui-Ling Lu and Julius O. Smith, III. Estimating glottal aspiration noise via wavelet thresholding and best basis thresholding. In *IEEE Workshop an the Application of Signal Processing to Audio and Acoustics*, pages 11–14, October 2001.

[LSM93] Jean Laroche, Yannis Stylianou, and Eric Moulines. HNS: Speech modification based on a harmonic+noise model. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 550–553, 1993.

[Luc93] Jorge C. Lucero. Dynamics of the two-mass model of the vocal folds: Equilibria, bifurcations, and oscillation region. *Journal of the Acoustic Society of America*, 94(6):3104–3111, December 1993.

[LZL03] Jianmin Li, Bo Zhang, and Fuzong Lin. 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)*, pages 259–265, Sacramento, CA, November 2003.

[Mac92a] David J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[Mac92b] David J.C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):698–714, 1992.

[Mac92c] David J.C. MacKay. A practical Bayesian framework for backprop networks. *Neural Computation*, 4(3):448–472, 1992.

[Mac99] David J.C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11(5):1035–1068, July 1999.

[Mak75] John Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, April 1975.

[MAK$^+$93] Marian Macchi, Mary Jo Altom, Dan Kahn, Sharad Singhal, and Murray Spiegel. Intelligibility as a function of speech coding method for template-based speech synthesis. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 893–896, Berlin, Germany, 1993.

[Man99] Iain Mann. *An Investigation of Nonlinear Speech Synthesis and Pitch Modification Techniques.* PhD thesis, University of Edinburgh, 1999.

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

[MB95] Alan V. McCree and T. P. Barnwell. A mixed exitation LPC vocoder model for low bit rate speech coding. *IEEE Transactions on Speech and Audio Processing*, 3(4):242–250, July 1995.

[MC81] John I. Makhoul and Lynn K. Cosell. Adaptive lattice analysis of speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(3):654–659, June 1981.

[MC90] Eric Moulines and Francis Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9:452–467, 1990.

[MC96] Michael W. Macon and Mark A. Clements. Speech concatenation and synthesis using an overlap-add sinusoidal model. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 361–364, 1996.

[MD89] John Moody and Christian J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

[MG76] John D. Markel and Augustine H. Gray, Jr. *Linear Prediction of Speech.* Springer, Berlin, Heidelberg, New York, 1976.

[MGS95] Dirk Michaelis, Tino Gramss, and Hans Werner Strube. Glottal-to-noise excitation ratio – a new measure for describing pathological voices. *Acta Acustica*, 81:700–706, 1995.

[MK93] Alan V. McCree and W. Bastiaan Kleijn. Mixed exitation prototype waveform interpolation for low bit rate speech coding. In *Proc. of IEEE Workshop on Speech Coding for Telecommunications*, pages 51–52, October 1993.

[MM98] Iain Mann and Steve McLaughlin. A nonlinear algorithm for epoch marking in speech signals using Poincaré maps. In *Proceedings of the European Signal Processing Conference*, volume 2, pages 701–704, September 1998.

[MM99] Iain Mann and Steve McLaughlin. Stable speech synthesis using recurrent radial basis functions. In *Proceedings of the European Conference on Speech Communication and Technology*, volume 5, pages 2315–2318, Budapest, Hungary, 1999.

[MM01] Iain Mann and Stephen McLaughlin. Synthesising natural-sounding vowels using a nonlinear dynamical model. *Signal Processing*, 81(8):1743–1756, 2001.

[Moo92] Francis C. Moon. *Chaotic and Fractal Dynamics.* Wiley, New York, 1992.

[Moo96] Todd K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, November 1996.

[MQ86] Robert J. McAuley and Thomas F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34(4):744–754, August 1986.

[NA00] Shrikanth Narayanan and Abeer Alwan. Noise source models for fricative consonants. *IEEE Transactions on Speech and Audio Processing*, 8(2):328–344, March 2000.

[Nea96] Radford M. Neal. *Bayesian Learning for Neural Networks.* Number 118 in Lecture Notes in Statistics. Springer, 1996.

[NPC99] Karthik Narasimhan, José C. Príncipe, and Donald G. Childers. Nonlinear dynamic modeling of the voiced excitation for improved speech synthesis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 389–392, Phoenix, Arizona, 1999.

[Oli80] Joseph Olive. A scheme for concatenating units for speech synthesis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 568–571, 1980.

[OM02] Daragh O'Brian and Alex Monaghan. Shape invariant pitch and time-scale modification of speech based on a harmonic model. In Keller et al. [KBM+02], pages 64–75.

[OS76] Joseph P. Olive and N. Spickenagel. Speech resynthesis from phoneme-related parameters. *Journal of the Acoustic Society of America*, 59(4):993–996, April 1976.

[O'S87] Douglas O'Shaughnessy. *Speech Communication: Human and Machine*. Addison-Wesley, 1987.

[OSS$^+$04] Asli Ozdas, Richard G. Shiavi, Stephen E. Silverman, Marilyn K. Silverman, and D. Mitchell Wilkes. Investigation of vocal jitter and glottal flow spectrum as possible cues for depression and near-term suicidal risk. *IEEE Transactions on Biomedical Engineering*, 51(9):1530–1540, September 2004.

[PCL89] Neal B. Pinto, Donald G. Childers, and Ajit L. Lalwani. Formant speech synthesis: Improving production quality. *IEEE Transactions on Speech and Audio Processing*, 37(12):1870–1887, December 1989.

[Pea98] Steve Pearson. A novel method of formant analysis and glottal inverse filtering. In *Proceedings of the International Conference on Spoken Language Processing*, volume 3, pages 1079–1082, Sydney, 1998.

[Per88] José Carlos Pereira. AC analysis of the three-mass model of the larynx. In *Proc. IEEE Conference on Engineering in Medicine and Biology*, volume 3, pages 1068–1069, New Orleans, 1988.

[Per89] José Carlos Pereira. Some results from the three-mass model of the larynx. In *Proc. IEEE Conference on Engineering in Medicine and Biology*, volume 3, pages 835–836, 1989.

[PG89] Tomaso Poggio and Federico Girosi. A theory of networks for approximation and learning. A.I. Memo 1140, Massachusetts Institute of Technology, 1989.

[PG90] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, September 1990.

[PM02] Vassilis Pitsikalis and Petros Maragos. Speech analysis and feature extraction using chaotic models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 533–536, Orlando (FL), USA, 2002.

[PMdM99] Carmen Peláez-Moreno and Fernando Díaz de María. Backward adaptive RBF-based hybrid predictors for CELP-type coders at medium bit-rates. In *Proceedings of the European Conference on Speech Communication and Technology*, volume 3, pages 1475–1478, Budapest, Hungary, 1999.

[PNR$^+$03] Michael Pucher, Friedrich Neubarth, Erhard Rank, Georg Niklfeld, and Qi Guan. Combining non-uniform unit selection with diphone based synthesis. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 1329–1332, Geneve, Switzerland, 2003.

[Por94] Thomas Portele. *Ein phonetisch-akustisch motiviertes Inventar zur Sprachsynthese deutscher Äusserungen.* PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 1994.

[PQR99] Michael D. Plumpe, Thomas F. Quartieri, and Douglas A. Reynolds. Modeling of the glottal flow derivative waveform with application to speaker identification. *IEEE Transactions on Speech and Audio Processing*, 7(5):569–586, September 1999.

[PWK98] José Príncipe, Ludong Wang, and Jyh-Ming Kuo. Non-linear dynamic modelling with neural networks. In Ales Prochazka, Jan Uhlir, Peter W. J. Rayner, and Nick G. Kingsbury, editors, *Signal Analysis and Prediction*, pages 275–289. Birkhäuser, Boston, 1998.

[Qua02] Thomas Quatieri. *Discrete-Time Speech Signal Processing: Principles and Practise.* Prentice Hall, 2002.

[Ran99] Erhard Rank. Exploiting improved parameter smoothing within a hybrid concatenative/LPC speech synthesizer. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 2339–2342, Budapest, 1999.

[Ran00] Erhard Rank. Über die Relevanz von alternativen LP-Methoden für die Sprachsynthese. In *Fortschritte der Akustik, DAGA-2000*, Oldenburg, 2000.

[Ran01] Erhard Rank. Synthese von Vokalen mit einem Oszillatormodell unter Berücksichtigung der stimmlosen Anregung. In Wolfgang Hess and Karlheinz Stöber, editors, *Zwölfte Konferenz Elektronische Sprachsignalverarbeitung ESSV*, Rüdiger Hoffmann: Studientexte zur Sprachkommunikation, Band 22, pages 136–143, 2001.

[Ran02] Erhard Rank. Concatenative speech synthesis using SRELP. In Keller et al. [KBM$^+$02], pages 75–86.

[Ran03] Erhard Rank. Application of Bayesian trained RBF networks to nonlinear timeseries modeling. *Signal Processing*, 83(7):1393–1410, July 2003.

[RK01] Erhard Rank and Gernot Kubin. Nonlinear synthesis of vowels in the LP residual domain with a regularized RBF network. In *Lecture Notes in Computer Science*, volume 2085, pages 746–753. Springer, 2001.

[RK03] Erhard Rank and Gernot Kubin. Towards an oscillator-plus-noise model for speech synthesis. In *Proceedings of the ISCA Tutorial and Research Workshop on Non-linear Speech Processing (NOLISP)*, Le Croisic, France, May 2003.

[Ros71] Aaron Rosenberg. Effect of glottal pulse shape on the quality of natural vowels. *Journal of the Acoustic Society of America*, 49:583–590, 1971.

[RP98] Erhard Rank and Hannes Pirker. VieCtoS—speech synthesizer, technical overview. Technical report, Österreichisches Forschungsinstitut für Artificial Intelligence, Wien, 1998.

[SA79] Christine H. Shadle and Bishnu S. Atal. Speech synthesis by linear interpolation of spectral parameters between dyad boundaries. *Journal of the Acoustic Society of America*, 66(5):1325–1332, November 1979.

[SA83] Celia Scully and E. Allwood. The representation of stored plans for articulatory coordination and constraints in a composite model of speech production. *Speech Communication*, 2(2-3):107–110, 1983.

[Sag88] Yoshinori Sagisaka. Speech synthesis by rule using an optimal selection of non-uniform synthesis units. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 679–682, New York City, USA, 1988.

[Sau92] Tim Sauer. A noise reduction method for signals from nonlinear systems. *Physica D*, 58(1-4):193–201, September 1992.

[SCAA01] Robert J. Schilling, James J. Carroll, and Ahmad F. Al-Ajlouni. Approximation of nonlinear systems with radial basis function neural networks. *IEEE Transactions on Neural Networks*, 12(1):1–15, January 2001.

[Sch90] Jean Schoentgen. Non-linear signal representation and its application to the modelling of the glottal waveform. *Speech Communication*, 9(3):189–201, 1990.

[Sch92] Jean Schoentgen. Glottal waveform synthesis with Volterra shaping functions. *Speech Communication*, 11:499–512, 1992.

[Scu86] Celia Scully. Speech production simulated with a functional model of the larynx and the vocal tract. *Journal of Phonetics*, 14:407–413, 1986.

[Scu90] Celia Scully. Articulatory synthesis. In W. J. Hardcastle and A. Marchal, editors, *Speech Production and Speech Modelling*, pages 151–186. Kluwer Academic Publishers, Dortrecht, The Netherlands, 1990.

[SdG91] Jean Schoentgen and Raoul de Guchteneere. An algorithm for the measurement of jitter. *Speech Communication*, 10:533–538, 1991.

[Ser89] Xavier Serra. *A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition.* PhD thesis, CCRMA, Department of Music, Stanford University, 1989.

[SG97] Jean Schoentgen and Raoul De Guchteneere. Predictable and random components of jitter. *Speech Communication*, 21:255–272, 1997.

[SK98] Jan Skoglund and W. Bastiaan Kleijn. On the significance of temporal masking in speech coding. In *Proceedings of the International Conference on Spoken Language Processing*, volume 5, pages 1791–1794, Sydney, 1998.

[SL00] Karl Schnell and Arild Lacroix. Bestimmung von Rohrmodellparametern aus Sprachsignalen. In *Fortschritte der Akustik - DAGA 2000*, 2000.

[SL01] Karl Schnell and Arild Lacroix. Inverse filtering of tube models with frequency dependent tube terminations. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 2467–2470, Aalborg, Denmark, 2001.

[SL03] Karl Schnell and Arild Lacroix. Generation of nasalized speech sounds based on branched tube models obtained from separate mouth and nose outputs. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume I, pages 156–159, Hong Kong, 2003.

[SLM95] Yannis Stylianou, Jean Laroche, and Eric Moulines. High-quality speech modification based on a harmonic + noise model. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 451–454, Madrid, Spain, 1995.

[SS01] Yannis Stylianou and Ann K. Syrdal. Perceptual and objective detection of discontinuities in concatenative speech synthesis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 837–840, Salt Lake City (UT), USA, 2001.

[ST95] Brad Story and Ingo Titze. Voice simulation with a body-cover model of the vocal folds. *Journal of the Acoustic Society of America*, 97(2):1249–1260, 1995.

[Sto74] Mervyn Stone. Cross-validation choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36:111–147, 1974.

[Stö02] Karlheinz Stöber. *Bestimmung und Auswahl von Zeitbereichseinheiten für die konkatenative Sprachsynthese.* PhD thesis, University of Bonn, Germany; P. Lang, Frankfurt, 2002.

[Str78] Hans Werner Strube. Fast straight-line-train fitting algorithm for application with Olive's speech-coding method. *Journal of the Acoustic Society of America*, 63(5):1636–1637, May 1978.

[Sty96] Yannis Stylianou. Decomposition of speech signals into a deterministic and a stochastic part. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 1213–1216, Philadelphia (PA), USA, 1996.

[Sty01] Yannis Stylianou. Applying the harmonic plus noise model in concatenative synthesis. *IEEE Transactions on Speech and Audio Processing*, 9(1):21–29, 2001.

[SV01] Mark Swerts and Raymond Veldhuis. The effect of speech melody on voice quality. *Speech Communication*, 33(4):297–303, 2001.

[SYC91] Tim Sauer, James A. Yorke, and Martin Casdagli. Embedology. *Journal of Statistical Physics*, 65(3/4):579–616, 1991.

[TA77] Andreĭ N. Tikhonov and Vasiliĭ Y. Arsenin. *Solutions of Ill-posed Problems.* W.H. Winston, 1977.

[Tak81] Floris Takens. On the numerical determination of the dimension of an attractor. In D. Rand and L. S. Young, editors, *Dynamic Systems and Turbulence*, volume 898 of *Warwick 1980 Lecture Notes in Mathematics*, pages 366–381. Springer, Berlin, 1981.

[TBW$^+$03] Mark Thomson, Simon Boland, Mike Wu, Julien Epps, and Michael Smithers. Decomposition of speech into voiced and unvoiced components based on a state-space signal model. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume I, pages 160–163, Hong Kong, 2003.

[Ter02a] Dmitry E. Terez. Robust pitch determination using nonlinear state-space embedding. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 345–348, Orlando (FL), USA, 2002.

[Ter02b] Jacques Terken. Variability and speaking styles in speech synthesis. In Keller et al. [KBM$^+$02], pages 199–203.

[Tip01] Michael E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[Tit88] Ingo Titze. The physics of small-amplitude oscillation of the vocal folds. *Journal of the Acoustic Society of America*, 83:1536–1552, 1988.

[TL88] Jack Terken and G. Lemeer. Effects of segmental quality and intonation on quality judgements for texts and utterances. *Journal of Phonetics*, 16:453–457, 1988.

[TNH94] Jes Thyssen, Henrik Nielsen, and Steffen Duus Hansen. Non-linear short-term prediction in speech coding. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 185–188, 1994.

[Tow91] Brent Townshend. Nonlinear prediction of speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 425–428, Toronto, ON, 1991.

[TS97] Ingo Titze and Brad Story. Acoustic interaction of the voice source with the lower vocal tract. *Journal of the Acoustic Society of America*, 101(4):2234–2243, 1997.

[Vap95] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New-York, 1995.

[Vel98] Raymond Veldhuis. A computationally efficient alternative for the LF model and its perceptual evaluation. *Journal of the Acoustic Society of America*, 103(1):566–571, 1998.

[vK70] Wolfgang van Kempelen. *Mechanismus der menschlichen Sprache nebst Beschreibung einer sprechenden Maschine*. Grammatica Universalis 4. Herbert E. Brekle, Verlag Frommann-Holzboog, 1970. Including a facsimile of the original publication (J. V. Degen, 1791).

[VMJ96] Maurílio N. Vieira, Fergus R. McInnes, and Mervyn A. Jack. Robust f_0 and jitter estimation in pathological voices. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 745–748, Philadelphia (PA), 1996.

[Whi36] Hassler Whitney. Differentiable manifolds. *Annals of Mathematics*, 37:645–680, 1936.

[WM98] Johan Wouters and Michael W. Macon. A perceptual evaluation of distance measures for concatenative speech synthesis. In *Proceedings of the International Conference on Spoken Language Processing*, volume 6, pages 2747–2750, Sydney, 1998.

[WM00] Johan Wouters and Michael W. Macon. Spectral modification for concatenative synthesis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 941–944, 2000.

[WM01] Johan Wouters and Michael W. Macon. Control of spectral dynamics in concatenative speech synthesis. *IEEE Transactions on Speech and Audio Processing*, 9(1):30–38, 2001.

[WMG79] David Y. Wong, John D. Markel, and Augustine H. Gray, Jr. Least squares glottal inverse filtering from the acoustic waveform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-27(4):350–355, August 1979.

[Wok97] Wolfgang Wokurek. Time-frequency analysis of the glottal opening. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 1435–1438, Munich, Germany, 1997.

[YH95] Paul V. Yee and Simon Haykin. A dynamic regularized Gaussian radial basis function network for nonlinear, nonstationary time series prediction. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3419–3422, Detroit (MI), USA, 1995.

[YH01] Paul V. Yee and Simon Haykin. *Regularized Radial Basis Function Networks: Theory and Applications*. Wiley, 2001.

[Zar02] Christopher J. Zarowski. Limitations on SNR estimator accuracy. *IEEE Transactions on Signal Processing*, 50(9):2368–2372, September 2002.