THE UNIVERSITY
of EDINBURGH

UNIVERSITY OF EDINBURGH

In Search of the Optimal Acoustic Features for Statistical Parametric Speech Synthesis

PhD Thesis

Felipe Espic

Doctor of Philosophy

The Centre for Speech Technology Research
Institute for Language, Cognition and Computation
School of Informatics

May 2019

Declaration of Authorship

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

(Felipe Espic)

UNIVERSITY OF EDINBURGH

Abstract

The Centre for Speech Technology Research
School of Informatics

by Felipe Espic

In the Statistical Parametric Speech Synthesis (SPSS) paradigm, speech is generally represented as acoustic features and the waveform is generated by a vocoder. A comprehensive summary of state-of-the-art vocoding techniques is presented, highlighting their characteristics, advantages, and drawbacks, primarily when used in SPSS. We conclude that state-of-the-art vocoding methods are suboptimal and are a cause of significant loss of quality, even though numerous vocoders have been proposed in the last decade. In fact, it seems that the most complicated methods perform worse than simpler ones based on more robust analysis/synthesis algorithms. Typical methods, based on the source-filter or sinusoidal models, rely on excessive simplifying assumptions. They perform what we call an "extreme decomposition" of speech (e.g., source+filter or sinusoids+noise), which we believe to be a major drawback. Problems include: difficulties in the estimation of components; modelling of complex non-linear mechanisms; a lack of ground truth. In addition, the statistical dependence that exists between stochastic and deterministic components of speech is not modelled.

We start by improving just the waveform generation stage of SPSS, using standard acoustic features. We propose a new method of waveform generation tailored for SPSS, based on neither source-filter separation nor sinusoidal modelling. The proposed waveform generator avoids unnecessary assumptions and decompositions as far as possible, and uses only the fundamental frequency and spectral envelope as acoustic features. A very small speech database is used as a source of base speech signals which are subsequently "reshaped" to match the specifications output by the acoustic model in the SPSS framework. All of this is done without any decomposition, such as source+filter or harmonics+noise. A comprehensive description of the waveform generation process is presented, along with implementation issues. Two SPSS voices, a female and a male, were built to test the proposed method by using a standard TTS toolkit, Merlin. In a subjective evaluation, listeners preferred the proposed waveform generator over a state-of-the-art vocoder, STRAIGHT.

Even though the proposed "waveform reshaping" generator generates higher speech quality than STRAIGHT, the improvement is not large enough. Consequently, we propose a new acoustic representation, whose implementation involves feature extraction and waveform generation, i.e., a complete vocoder. The new representation encodes the complex spectrum derived from the Fourier Transform in a way explicitly designed for SPSS, rather than for speech coding or copy-synthesis. The feature set comprises four feature streams describing magnitude spectrum, phase spectrum, and fundamental frequency; all of these are represented by real numbers. It avoids heuristics or unstable methods for phase unwrapping. The new feature extraction does not attempt to decompose the speech structure and thus the "phasiness" and "buzziness" found in a typical vocoder, such as STRAIGHT, is dramatically reduced. Our method works at a lower frame rate than a typical vocoder. To demonstrate the proposed method, two DNN-based voices, a male and a female, were built using the Merlin toolkit. Subjective comparisons were performed with a state-of-the-art baseline. The proposed vocoder substantially outperformed the baseline for both voices and under all configurations tested. Furthermore, several enhancements were made over the original design, which are beneficial for either sound quality or compatibility with other tools. In addition to its use in SPSS, the proposed vocoder is also demonstrated being used for join smoothing in unit selection-based systems, and can be used for voice conversion or automatic speech recognition.

Acknowledgements

I would like to express my gratitude to:

- Simon King for all his advise, support, scientific freedom, and extensive knowledge provided throughout my PhD studies. Also, for proofreading this thesis and trusting in me as a future doctor.
- Cássia Valentini Botinhão for all the valuable conversations, technical knowledge and support during my PhD.
- Zhizheng Wu for his help and development of the Merlin toolkit, which was essential for the experiments in this thesis.
- Gustav Eje Henter for providing the MUSHRA testing and analysis code used for the experiments in this work.
- Rasmus Dall, for sharing his knowledge and having relevant discussions, especially about frame rate handling and MLPG.
- All my colleagues and ex-colleagues from The Centre for Speech Technology Research (CSTR), especially Oliver Watts, Junichi Yamagishi, Srikanth Ronanki, Sam Ribeiro, Korin Richmond, and Gustav Eje Henter for all the exciting conversations and support.
- Korin Richmond and Paavo Alku for their commitment to reviewing this work, and providing sensible and detailed comments leading to a thesis of much higher quality than the original.
- The Speech Synthesis community, especially the vocoding people who with I have shared knowledge and ideas in conferences, visits, and remotely, especially Lauri Juvela, Fernando Villavicencio, and Gilles Degottex.
- My family, friends, and Valeria Rebolledo for her unconditional support and patience over the last years.

This work was supported by the Chilean National Commission for Scientific and Technological Research (CONICYT) Becas Chile scholarship: PFCHA-Becas - 72150507.

Contents

Chapter 1

Introduction

In spite of the fact that the speech quality achieved by Statistical Parametric Speech Synthesis (SPSS) is comparable to unit selection-based methods, the degree of naturalness produced by SPSS is still low. It is well known that in terms of naturalness, unit selection and hybrid methods outperform SPSS (King and Karaiskos, 2012). According to Zen et al. (2009), its low quality is due to three factors: over-simplified vocoder techniques that cannot generate detailed speech waveforms, over-smoothing of speech parameters, and acoustic modelling inaccuracy.

The effect of over-simplified vocoder techniques is particularly obvious when the unnaturalness persists even when performing simple analysis/synthesis of speech, which is commonly termed "copy-synthesis". It means that even in ideal conditions, namely when the true acoustic parameters are used, these vocoders still fail. Some may produce an artificial sound, which is perceived as "phasiness", "buzziness", "muffled sound", or "transient smearing".

Some vocoders, especially the ones based on sinusoidal models (e.g., Degottex and Stylianou, 2013; Stylianou et al., 1995), work quite well when performing copy-synthesis, generating accurate synthesised speech. However, artefacts are still present in the generated signal. This is attributed to their inability to correctly model aperiodicities of natural speech. Unfortunately, these artefacts are even more noticeable when applying modifications to the acoustic parameters, such as fundamental frequency (F0), spectral envelope, etc.

Also, it has been shown that the quality achievable by vocoders depends on the speaker's voice and style (Degottex and Erro, 2014). Usually, this effect is more prominent when comparing synthesised speech from female and male speakers. Accordingly, there are some vocoders that work better with female than male speech, and vice-versa. So

far, no method to predict the "vocodability" of a speaker has been proposed, thus empirical experimentation is the only way to determine the usability of a particular speaker's voice. This implies a high risk due to the associated costs of professional audio recording and voice building.

The effects of the mentioned drawbacks are increased when the acoustic parameters provided to vocoders are not extracted from natural speech, but inferred from a statistical model, such as Hidden Markov Model (HMM) or Deep Neural Network (DNN).

The large number of features that vocoders extract is another shortcoming. Currently, state-of-the-art vocoders (e.g., Kawahara et al., 1999b) work with a large number of parameters, typically in the order of thousands. It is important to notice that statistical models usually contain some elements with quadratic computational complexity (e.g., a neural network layer), which makes the number of features critical in terms of efficiency.

Regarding all the described imperfections of vocoding, it is of utmost importance to find a method of speech signal analysis/generation that meets well defined requirements. This method must be:

- Robust to different speakers and fundamental frequency fluctuations.
- Capable of producing high speech quality that outperforms state-of-the-art vocoding techniques.
- Aimed at achieving best results when used for acoustic modelling, rather than for speech coding (i.e., high quality copy-synthesis).

1.1 Contributions

The contributions presented in this thesis are:

- Avoiding unnecessary decomposition of speech, such as separation into source-filter, stochastic-plus-deterministic, harmonics-plus-noise, etc., high quality waveform generation can be achieved by applying characteristics of natural speech directly to generate speech.
- Phase spectra features can be extracted from speech signals without using heuristics or iterative algorithms for phase unwrapping. These features exhibit enough consistency to be statistically modelled.
- Real-valued neural networks are capable of learning magnitude and phase spectra directly extracted from natural speech, producing high quality synthesised speech.
- We implemented fast and efficient software from scratch, freely available and open source, written in MATLAB and Python.

1.2 Outputs

Papers:

- C. Valentini-Botinhao, O. Watts, F. Espic, and S.King, "Examplar-based speech waveform generation for text-to-speech," In Proc. 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, December, 2018, (Accepted).
- F. Espic, A. Govender, M. S. Ribeiro, C. Valentini-Botinhao, O. Watts, "The CSTR entry to the Blizzard Challenge 2018," In Proc. Blizzard Challenge, Hyderabad, India, September, 2018.
- O. Watts, C. Valentini-Botinhao, F. Espic, and S. King "Exemplar-based Speech Waveform Generation," in Proc. Interspeech, Hyderabad, India, September, 2018.
- S. Ronanki, M. S. Ribeiro, F. Espic, and O. Watts "The CSTR entry to the Blizzard Challenge 2017," In Proc. Blizzard Challenge, Stockholm, Sweden, August, 2017.
- F. Espic, C. Valentini-Botinhao, and S. King, "Direct Modelling of Magnitude and Phase Spectra for Statistical Parametric Speech Synthesis," in Proc. Interspeech, Stockholm, Sweden, August, 2017.

- F. Espic, C. Valentini-Botinhao, Z. Wu, and S. King, "Waveform generation based on signal reshaping for statistical parametric speech synthesis," in Proc. Interspeech, San Francisco, CA, USA, September, 2016.
- F. Villavicencio, J. Yamagishi, J. Bonada, and F. Espic, "Applying spectral normalisation and efficient envelope estimation and statistical transformation for the Voice Conversion Challenge 2016," in Proc. Interspeech, San Francisco, CA, USA, September, 2016.

Talks:

- F. Espic, C. Valentini-Botinhao, and S. King, "MagPhase Vocoder: Magnitude and phase analysis/synthesis for statistical parametric speech synthesis," UK-Speech Conference, Cambridge, UK, September, 2017.
- S.King, O. Watts, S. Ronanki, Z. Wu, and F. Espic "Tutorial: Deep Learning for Text-to-Speech Synthesis, using the Merlin toolkit," Interspeech Conference, Stockholm, Sweden, August, 2017.
- F. Espic, Talk on Waveform generation based on signal reshaping for statistical parametric speech synthesis. UK-Speech Conference, Sheffield, UK, June, 2016.

Open Source Software:

- MagPhase Vocoder (author): https://github.com/CSTR-Edinburgh/magphase
- WavGenSR (author): https://github.com/CSTR-Edinburgh/WavGenSR
- Snickery (collaborator): https://github.com/CSTR-Edinburgh/snickery
- Merlin (collaborator): https://github.com/CSTR-Edinburgh/merlin

1.3 Background

The problem of the low quality in statistical parametric speech synthesis (SPSS) has been studied by several researchers.

Source-filter separation, which is typically used in SPSS, has been the focus of much study. Titze (2008) showed that the interactions that occur between the glottal airflow and the acoustic vocal tract pressures, lead to harmonic distortion, and modifications in the spectral slope and the spectral ripple. Thus, the filter and source, as they are mutually dependent, cannot be separated. Furthermore, as their interactions are non-linear, a linear operation, as filtering, doesn't provide the best fit for a good approximation to a full separation.

Merritt et al. (2014) analysed the issue by looking for the causes of deterioration in the statistical model as well as in the waveform generation stage. The effect of the separation performed by the source-filter model was investigated by generating synthetic speech using combinations of vocoded, modelled, and natural source and filters. In addition, the effects of over-smoothing, resonance sharpness or modulation scaling of the filter were studied and modelled. Some of the experiments carried out consisted of imposing vocoded vocal tract filters to natural source signals extracted from different excerpts. Then, subjective tests were performed to evaluate speech quality. Some of the findings in this research were: the current filter enhancement techniques are able to recover the quality loss caused by the filter modelling; the dependence between the source and the filter is one of the most important factors that determines the synthetic speech quality.

Another study (Henter et al., 2014) was conducted by using repeated speech from the database REpeated HArvard Sentence Prompts (REHASP), which provides several audio excerpts from the same speaker saying the same prompt. By employing this database, it was possible to combine acoustic features from different repetitions of the same prompt to evaluate the dependency among them. Listening tests were carried out, whose results showed that vocoding, along with the use of Mel-Frequency Cepstral Coefficients (MFCCs) are critical sources of degradation in speech synthesis. Also it was shown that acoustic features (e.g., source and filter) are dependent, breaking the assumption of independence that is usually applied in for example, HMM-based speech synthesis.

It is also worth mentioning that another important source of degradation in SPSS is the over-smoothing applied by the acoustic model (e.g., HMM, DNN), although it is out of scope of this research.

1.3.1 Speech Synthesis Methods

A diversity of speech synthesis methods have been proposed so far, but just a few have been used in practical applications. For this very reason, some methods are left out of scope and are not reviewed in this document (e.g., Toda et al., 2008). Thus, the following sections describe the current state-of-the-art speech synthesis methods.

1.3.1.1 Unit Selection

There are several formulations to perform unit selection. Here we will describe the *Independent-Feature Target-Function Formulation*, which is the one proposed in the original paper describing unit selection speech synthesis by Hunt and Black (1996).

Nowadays, unit selection is the predominant technique for text-to-speech in production, and can be understood as an extension to concatenative synthesis (Taylor, 2009, p. 474). Essentially, the system contains a database of basic speech units (waveforms), each characterized by different features, such as: phonetic transcription, pitch, duration, prosody, emotion, speaking style, intonation, phrasing, etc. During synthesis, a list of features ("specification") is generated. Then, the sequence that suits the specification the best, is selected and concatenated to generate the waveform. Note that all the basic units contained in the database may be used for synthesis.

It is known that the more contiguous the chosen units are in the database, the higher is the output quality. This is because the system would generate less artificial joins, preventing audible "clicks". Also, since the units are closer in the original recordings, the variability in intonation and style of the speaker is minimised. Thus, when using big databases, it is more probable to find contiguous or close units that match the specification, which ensures high quality synthesis.

Hence, the problem to be solved is how to select the unit sequence to generate the highest speech quality. To do so, the formulation proposed by Hunt and Black (1996) is typically used:

$$\hat{U} = \arg\min_{u} \left\{ \sum_{t=1}^{N} T(u_t, s_t) + \sum_{t=1}^{N-1} J(u_t, u_{t+1}) \right\} \tag{1.1}$$

Where $\hat{U}$ is the optimum unit sequence estimate, t is the time index in the sequence. $T(u_t, s_t)$ is the *target cost*, which represents the difference between the unit u_t and the specification item s_t. $J(u_t, u_{t+1})$ is the *join cost*, which serves as a measure of how well two consecutive units join.

Target Function

Every s_t and u_t has a *target feature structure* that is used in the evaluation of the *target function* $T(u_t, s_t)$.

The aim of the *target function* is to evaluate the suitability of the units according to the specification. Basically, this function should return a list of all basic units in the database with an associated cost, which gives the distance between the requested features by the specification and the features of each unit.

The formulation of the *target function* is:

$$T(s_t, u_i) = \sum_{p=1}^{P} w_p \left(T_p(s_t[p], u_i[p])\right) \tag{1.2}$$

Where P is the number of features, s_t is the requested specification at time index t, u_i is ith basic unit contained in the data base, $T_p(s_t[p], u_i[p])$ represents the distance between the feature p of the specification s_t with the feature p of the unit u_i, and w_p is the weight assigned to the feature p.

Join Function

The *Join Function* measures how well two consecutive units join. Every unit contains two additional feature structures, for its left and right boundaries. Depending on how it is formulated, the *Join Function* could return the cost of joining two units, classify whether two units join correctly, or return the probability that two units append adequately.

Different types of acoustic features can be used, such as: *Cepstral Coefficients*, *Linear Prediction Cepstral Coefficients*, *Mel-Frequency Cepstral Coefficients (MFCCs)*, *Line-Spectral Frequencies (LSFs)*, *Fundamental Frequency (f0)*, *Energy*, and so on. In order to measure distance, it is possible to use: *Euclidean*, *Mahalanobis*, or *Kullback-Leibler Divergence* in case the probability distributions are available.

Using acoustic features looks promising. However, these seem to miss some relevant information: the use of only frames that are close to the boundaries does not explain the whole phonetic context, which may include more than one phoneme.

1.3.1.2 Hidden Markov Models (HMM)

An alternative for speech synthesis is the use of machine-learning techniques. In contrast to the unit selection method, during synthesis it infers acoustic features, which are directly used to generate the waveform without employing any audio recording as a source for the synthesised signal.

The HMMs are composed by states linked to each other. Every state is represented by an observation distribution. This probability density function (PDF) represents the distribution of the observations that can be emitted by this state (Taylor, 2009, p. 435-438). Generally, these acoustic observations are of types MFCCs, or LSFs extracted from speech data. Usually, in speech synthesis an HMM of five emitting states models a phoneme produced within an specific linguistic context. In order to generate words or phrases, the HMMs are concatenated sequentially.

The observations can be modelled as normal distributions, although it has been observed that this model is too simple (Taylor, 2009, p. 435-438). Hence, Gaussian Mixture Models (GMM) are the most common choice:

$$b(o_t) = \sum_{m=1}^{M} \omega_m \mathcal{N}(o_t; \mu_m, \Sigma_m) \tag{1.3}$$

Where $b(o_t)$ is the probability that the observation o is observed at time t, M is the number of mixture components, ω_m is the weight of the component m, and $\mathcal{N}(o_t; \mu_m, \Sigma_m)$ is the multivariate normal PDF for the mth component with mean μ_m, and covariance matrix Σ_m.

The low correlation among the acoustic features (e.g., MFCCs) implies also low covariance between them. That makes the covariance matrix Σ_m exhibit low values for all the off diagonal elements. Thus, the off diagonal elements can be set to zero, with the objective of simplifying later computations. That is Σ_m is now forced to be a diagonal matrix, reducing the number of model parameters. As a result, not only the system will be more computationally efficient, but also it will need less data to be trained, i.e., less sensitive to data sparsity.

Some researchers have tried using full-covariance matrices, since the forced diagonalisation implies some error due to the fact that correlations between features are ignored (Zen et al., 2008). They use the maximum likelihood linear transform (MLLT) (Gopinath, 1998) to estimate the matrix. However, they have found some disadvantages of using the full-covariance. Firstly, it requires a high amount of training data to reach reliable estimates (data sparsity problem) (Zen et al., 2008). Secondly, its effects

vary by speaker (Yamagishi et al., 2009). Thirdly, the it is not clear when the use of full-covariance matrices is useful, e.g., it seems to work better when used together with global variance (GV) (Toda and Tokuda, 2005; Zen et al., 2008).

Aside from observation probabilities, it is necessary to define the probability of moving from one state to another, a. Finally, the typical formulation of the probability of the sequence of observations O and the sequence of states Q, given the model HMM λ is given by:

$$P(O, Q|\lambda) = a_{q_{(0)}q_{(1)}} \prod_{t=1}^{T} b_{q_{(t)}}(o_t) a_{q_{(t)}q_{(t+1)}} \tag{1.4}$$

Where $a_{q_x q_y}$ is the transition probability of moving from state q_x to state q_y, and $b_{q_{(t)}}(o_t)$ is the observation probability of the observation o_t given the state $q_{(t)}$. However, in a speech synthesis framework, it is necessary to express the transitions a between states as duration-dependent state transitions rather than a simple transition probability. The *Hidden Semi-Markov Model (HSMM)* proposed in Levinson (1986) replaces the typical transition probabilities by a skewed Gaussian probability distribution, which explicitly describes the probability of duration of a state before transitioning to the next one.

Training

In order to train the HMMs, only the duration PDFs, the covariance matrices Σ_m and the means μ_m of the GMMs have to be estimated. However, to perform this task, it is necessary to align the specifications with the observations. In this regard, the HMMs are used in forced speech recognition mode, namely they know the phones in context to synthesise beforehand (Taylor, 2009, p. 447-451).

Context-sensitive modelling assumes that phonemes are affected by linguistic context, which may include: identity, phonetic class of next and previous phonemes, part-of-speech and positional information within a syllable, word, phrase, etc. Thus, a different HMM is built to represent each of these context-dependent phonemes. Decision-trees are used to cluster the data into different contexts, which according to contextual questions (e.g., "is the next phone a stop?", "is the previous phone voiced?") classifies the instances into different clusters. However, this procedure usually doesn't cover all the possible combinations of phonemes within contexts, due to the lack of examples in the database. Also, for this very reason, some models may be built using too few examples. To address these two problems, a process called *State Tying* is applied to build models with limited data or even unseen data by combining distributions of states belonging to different contexts (Young et al., 1994).

Synthesis

The phoneme sequence is given, but there is no indication of which observations to use for the signal generation. Parameters needs to be generated frame-by-frame (e.g., every 5ms) from the state sequence derived from the input phoneme sequence. The number of frames per state is given by the duration probability distribution (HSMM). In principle, this procedure could be performed by random sampling from the state distributions or by emitting the most likely observations (i.e., means). However, the former produces the spectra to change rapidly and randomly, whilst the latter causes piecewise trajectories that sound completely unnatural.

Tokuda et al. (1995b) proposed a method named Maximum Likelihood Parameter Generation (MLPG), which is able to generate smoothly evolving acoustic features, resembling more to natural acoustic parameters. It needs additional features to capture the evolution of the parameters over time. These are the dynamic features delta (Δc_t) and delta-delta coefficients ($\Delta^2 c_t$), which represent the velocity and acceleration of acoustic parameters over time. As a result, the generated trajectories are smoother and the final synthesised speech sounds more natural. The MLPG algorithm defines time indexed observation vectors $O_t = [c_t^T, \Delta c_t^T, \Delta^2 c_t^T]$. The set of these vectors is defined as the matrix $\mathbf{O}$:

$$\mathbf{O} = \mathbf{WC} \tag{1.5}$$

Where $\mathbf{C}$ is the set of static acoustic parameters, and $\mathbf{W}$ is a block-diagonal matrix. By assuming that the observations are normally distributed, the probability of the whole sequence of static acoustic vectors $\hat{\mathbf{C}}$ is given by:

$$\hat{\mathbf{C}} = (\mathbf{W^T \Sigma^{-1} W})^{-1} \mathbf{W^T \Sigma^{-1}} \boldsymbol{\mu} \tag{1.6}$$

Where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix of the observation vectors set $\mathbf{O}$. These matrices are predicted by the model or using global variances for the case of $\boldsymbol{\Sigma}$ (Klimkov et al., 2018).

Acoustic Features

In speech synthesis, several types of acoustic features are used to represent speech. These features are in independent spaces or *streams*. Some examples of typical feature streams are representations of: spectral envelope, fundamental frequency, or aperiodicities. The integration of streams in the HMM is formulated as:

$$b_q(o_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M} \omega_{qms} \mathcal{N}(o_t; \mu_{qms}, \Sigma_{qms}) \right]^{\gamma_s} \tag{1.7}$$

Where S is the number of streams and γ_s is the weight for stream s.

During inference, feature streams are emitted from the HMMs. Then, a vocoder generates the waveform from the predicted acoustic features after passing through the MLPG smoothing.

1.3.1.3 Deep Neural Networks (DNN)

Along with HMM-based methods, neural networks-based speech synthesis is another type of statistical parametric approach, used since a long time ago (e.g., Karaali et al., 1996). However, with the recent progress in hardware and software, Deep Neural Networks (DNN) started as a new trend in SPSS (Zen et al., 2013). In spite of the attractive attributes of HMM speech synthesis when comparing with unit selection (e.g., robustness, small foot print, intelligibility), it exhibits some limitations. The most relevant is the low quality of the generated speech, which is perceived as unnatural.

A typical HMM-based system may take around 50 different linguistic contextual characteristics into account. Unfortunately, the data is not capable of providing all the required linguistic context combinations. For this very reason is that decision tree clustering is employed, since it has shown to be efficient to deal with this type of shortcomings. However, decision tree clustering presents some drawbacks as well. For example, some experiments carried out by Merritt et al. (2015) assessed the perceptual effects of the averaging of parameters across contexts, which showed that this process is very harmful for the perceived speech quality.

DNN-based speech synthesis can be thought of as a method to map linguistic to acoustic features. The linguistic features are grouped into a linguistic specification, which is set as input to the network. The linguistic specification contains information about the current phoneme, its duration, and its phonetic context expressed as answers to binary questions (e.g., "is the next phone a stop?", "is the previous phone a voiced?"). At the output, the network predicts acoustic features that describe the speech waveform at a certain point in time.

There are several DNN-based speech synthesis methods proposed so far. The first successful method was proposed by Zen et al. (2013) which is the base for most SPSS systems nowadays. The acoustic features used for the method are: Mel-Cepstrum, logF0, band-aperiodicities, and a binary voicing decision. In a simple feed-forward network, the mapping (prediction) is separately carried out for each time frame with no information shared among consecutive time steps. As a result, the predicted features don't exhibit smooth trajectories as natural speech does, introducing audible artefacts.

Hence, some smoothing to the acoustic features needs to be used. Usually, the MLPG algorithm is applied as naturally inherited from the HMM-based speech synthesis. However, also some low-pass filters (LPF) have been used to smooth the trajectories instead, making the smoothing process more efficient.

For recurrent neural networks (e.g., RNN, LSTM), the use of smoothing is debatable. Several studies have tried to answer this question by carrying out subjective tests. Zen and Sak (2015) showed that for LSTMs, the use of MLPG was beneficial when using a feed-forward output layer. On the contrary, if a recurrent output layer is applied, the system without MLPG was preferred. Other studies as Wang et al. (2017) showed that MLPG is beneficial when using linear output layers. Furthermore, Klimkov et al. (2018) concluded that MLPG is not useful when using L1 as a loss function, but it is beneficial in other conditions (e.g., L2 loss function). Besides, a simple moving average performed similarly to MLPG. Hence, it is not clear if the use of smoothing of parameters (MLPG or LPF) is beneficial, it seems that the answer to this question is highly sensitive to the architecture design.

Finally, the generated features are processed by a vocoder to synthesise the signal on a frame-by-frame basis. Figure 1.1 shows the original DNN-based system proposed by Zen et al. (2013).

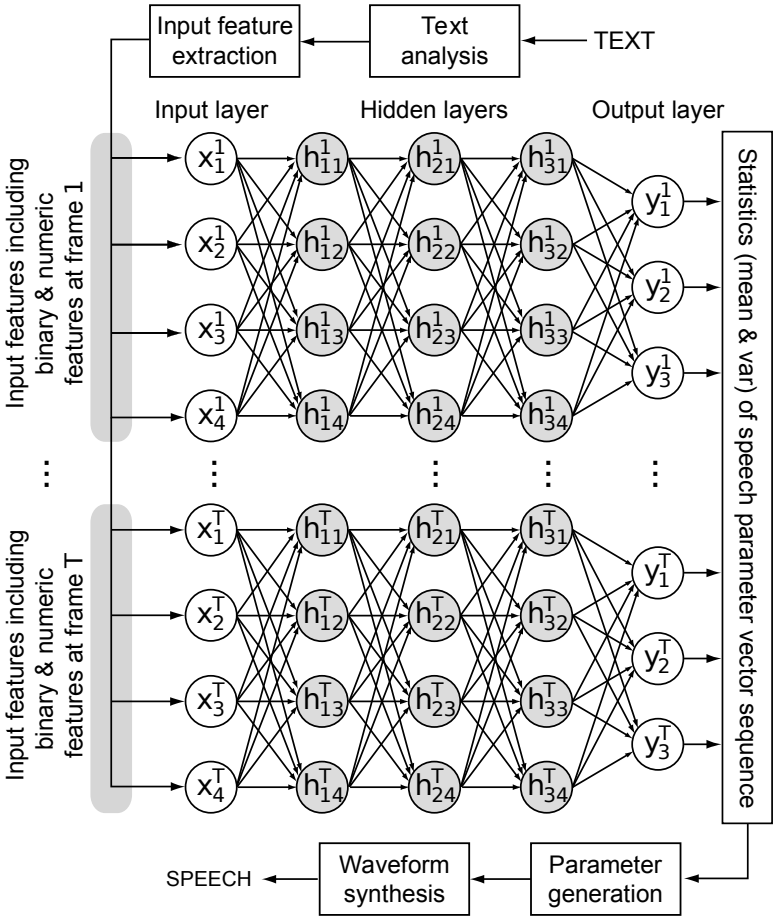

FIGURE 1.1: DNN Speech Synthesis framework. Figure extracted from Zen et al. (2013).

1.3.1.4 Hybrid Methods

Hybrid methods for speech synthesis have attracted attention recently due to the high quality synthetic speech they produce. Basically, these methods combine statistical parametric synthesis with concatenative systems (i.e., unit selection). By doing so, hybrid methods are capable of taking advantage of the benefits of each method, therefore maximising the system performance.

One of the current leading hybrid methods is called *Multiform Synthesis* proposed in Pollet and Breen (2008). It uses a statistical framework to concatenate speech generated from models with pre-stored natural speech segments. It defines two types of units: *template segments*, which are recorded speech sections that have been phonetically and prosodically labelled; and *model segments* that are produced by Hidden Markov Models (HMMs). Although this method was built with HMMs, in principle any other statistical model (e.g., DNN) may be used.

In addition to the typical acoustic parameters, such as spectral envelope and fundamental frequency, etc., the hybrid approach uses the variance of the acoustic parameters of natural speech as observations. By using HMMs it is possible to construct acoustic parameter trajectories from the model segments. Then, the *template segments* with the closest acoustic characteristics to the ones plotted by the trajectories are chosen as unit candidates.

Finally, to generate synthetic speech either a *model segment* or a *template segment* is chosen as a synthesis unit to be concatenated with the units previously selected according to phonologic, and acoustic cues.

A second approach that combines a statistical parametric method with a concatenative system was proposed in Qian et al. (2010), and Qian and Soong (2012). It is called *HMM Trajectory Tiling (HTT).*

For HMM Trajectory Tiling, HMMs are trained by using the Minimum Generation Error (MGE) criterion. In principle, the objective of HMM-based speech synthesis is to generate synthetic acoustic features as close as possible to the ones extracted from natural speech. That is we want to minimise the generation error, which is defined by (Wu and Wang, 2006):

$$l(\mathbf{C}, \lambda) = \sum_{all\mathbf{Q}} P(\mathbf{Q}|\lambda, \mathbf{O}) D(\mathbf{C}, \hat{\mathbf{C}}) \tag{1.8}$$

Where $\mathbf{C}$ is the set of static acoustic features extracted from natural speech, λ the set of model parameters, $\mathbf{Q}$ the state sequence, $\mathbf{O}$ The set of observations, and $D(\mathbf{C}, \hat{\mathbf{C}})$ the Euclidean distance between $\mathbf{C}$ and the predicted static acoustic features $\hat{\mathbf{C}}$. Then, using an iterative algorithm (GPD, See details in Blum (1954)), the optimal model parameters λ are found by:

$$\lambda_{n+1} = \lambda_n - \epsilon_n \frac{\partial l(\mathbf{C_n}, \lambda)}{\partial \lambda}\Big|_{\lambda=\lambda_n} \tag{1.9}$$

Where n is the current iteration index, and ϵ the step size.

Then, formant sharpening is applied to mitigate the over-smoothing that is typical of HMMs. Later, the best segments are selected from an inventory of units depending on the generated acoustic features. The selected units are concatenated according to the cross-correlation between the segments (waveforms) to avoid artefacts around the joins. It is measured in pairs, between two consecutive segments at different time lags. Then, the segments are joined using the time lag that gives the highest cross-correlation. By doing so, the spectral similarity and phase continuity of consecutive units is optimised.

These methods produce high quality synthetic speech. In fact they are considered among the best systems and are widely used in production currently. It seems that the strategy of combining the best of each paradigm improves the perceived quality of the synthesised speech. On one hand, SPSS provides smooth acoustic feature trajectories, high ineligibility, acoustic consistency, whilst unit selection provides high quality natural speech waveforms.

1.3.1.5 New Neural TTS

During the last couple of years, the idea of developing end-to-end systems became a trend. Neural sequence-to-sequence speech synthesis has been shown to be suitable as an approximation to a full end-to-end system; some examples are: Arik et al. (2017); Wang et al. (2017). Both systems are made up by several separated neural networks interconnected with the objective of mapping from a linguistic specification to its corresponding synthesised waveform. However, text processing (i.e., text normalisation and grapheme-to-phoneme) still needs to be performed by typical rule-based methods. For the last stage in the pipeline, a new type of vocoder is typically used, the *neural vocoder*.

To the best of our knowledge, the first attempt to build a neural vocoder was carried out by Tokuda and Zen (2016), who attempted to generate the waveform from a deep neural network directly in the time domain, rather than using a signal processing-based vocoder. The first model that succeeded was the WaveNet (van den Oord et al., 2016), which was able to generate the waveform directly in the time domain, either without conditioning or conditioned by some features, such as: linguistic specification (for TTS), speaker identity, fundamental frequency, Mel-Cepstrum, etc.

Basically, the WaveNet is a deep autoregressive neural network made up of a stack of dilated causal convolutions, which solves for a classification task (softmax output). The classes at the output are the quantisation steps of the PCM signal representation (i.e., uncompressed raw audio). A single audio sample is generated at each generation step, and then the past generated samples are fed into the network as part of the input features. Even though the system is capable of generating speech almost not distinguishable from natural speech, it is highly inefficient, making it impractical for production systems.

Some other systems have been proposed with the goal of predicting speech waveforms directly from neural architectures, but with the degree of efficiency required for production. Recurrent neural networks were used with equivalent success as WaveNet,

but achieving higher computational efficiency (Mehri et al., 2016; Kalchbrenner et al., 2018).

Due to its autoregressive nature, the parallelisation of the WaveNet was not feasible during inference. Hence, the Parallel WaveNet was proposed to make the parallelisation possible, and then the WaveNet fast enough to run in real time applications (van den Oord et al., 2017). Even though the original WaveNet shares the same name with the new parallelised version, their architectures differ greatly. Basically, Parallel WaveNet is trained in a teacher-student fashion to learn "the probability distributions of the probability distributions" of the audio samples. While in generation, a random noise is fed into the input of a series of cascaded networks, which filter the noise successively until reaching the desired quality.

All of the systems named in this section achieve remarkable high quality, even producing synthetic speech that compares to natural speech. However, they still exhibit some drawbacks:

- No appropriate controllability (e.g., it is hard to fix some errors).
- High inefficiency (large number of layers, recurrent architecture, etc.)
- Slow and complicated training.
- Large amount of training data needed.

1.3.2 State-of-the-Art Vocoding Techniques

Most common vocoders used in statistical parametric TTS are based on the source-filter model. The filter is often realised as a minimum phase filter derived by cepstral analysis of a smooth spectrum envelope (Kawahara et al., 1999b). The source is derived directly from the residual (Maia et al., 2007; Drugman et al., 2009), the glottal signal (Raitio et al., 2011) or more commonly from a parametric representation of the excitation signal, simple (pulse train or white noise) (Yoshimura et al., 1999) or mixed (a mixture of periodic and aperiodic signals) (Yoshimura et al., 2001). An alternative to the source and filter paradigm, the sinusoidal vocoders (Hemptinne, 2006; Banos et al., 2008; Hu et al., 2014) are powerful copy-synthesis tools (Hu et al., 2013). Their parameters however are hard to model since their dimensionality vary over time. Both paradigms suffer from poor modelling of aperiodicity components.

The following subsections describe the state-of-the-art vocoding methods, including source-filter and harmonic models that have been used in diverse statistical parametric speech synthesis systems.

1.3.2.1 STRAIGHT

STRAIGHT stands for Speech Transformation and Representation using Adaptive Interpolation of Weight Spectrum. This was conceived as a robust vocoding method that was developed to address the problem of "buzziness" that most other vocoders present. STRAIGHT parametrises speech using three feature streams: $F0$, *spectral envelope* and *aperiodicity measurements*.

The analysis of speech is carried out in several steps:

1. F0 Extraction:
   The fundamental frequency is extracted basically by using an analysing wavelet, whose frequency response resembles a low-pass filter whose cut-off frequency is located in low frequencies. The filter adapts its cut-off frequency to isolate the first harmonic of the signal. The filter cut-off frequency is adapted to isolate the first harmonic of the signal by measuring the signal-to-noise radio and the amplitude (AM), and frequency modulations (FM) of the filtered signal. Thus, if the signal-to-noise ratio is too low, it means that there is no harmonic within the response area, on the contrary if there is more than one harmonic within the response area, the beating between the components will result in AM and/or FM.

During extraction, the $F0$ is represented by a nearly harmonic model, which is aware of the fact that harmonic frequencies of speech are not constant and evolve continuously. This nearly harmonic model is defined as:

$$x(t) = \sum_{k=1}^{K} a_k(t) \cos\left(\int_0^t (k\omega_0(\tau) + \omega_k(\tau))d\tau + \phi_k(0)\right) \tag{1.10}$$

Where $x(t)$ is the speech signal at time t, K is the number of harmonics, $a_k(t)$ is the slowly changing amplitude of the kth harmonic at time t, $\omega_k(\tau)$ is a slowly changing frequency perturbation of the kth component, and $\phi_k(0)$ represents the initial phase of the kth component at time $t = 0$ (Kawahara et al., 1999b).

2. Spectral Envelope Extraction:
   Usually, computing the Short-time Fourier spectrogram is the first step to extract spectral features, e.g., spectral envelope. Unfortunately, the result is prone to spectral interferences produced by signal periodicity. For example, if the frame length is comparable to the pitch cycle, it will produce temporal variations in the analysis, whilst on the contrary if it is too long, it will exhibit frequency domain variations. Hence, STRAIGHT applies a pitch-adaptive smoothing of the spectrum (2nd-order cardinal B-spline) while keeping equivalent resolution for time and frequency domain. It is worth emphasising that the pitch-adaptive algorithm is robust to pitch estimation errors.

3. Aperiodicity Estimation:
   The deviation of periodicity is considered aperiodicity, thus it is measured as the ratio between the spectral power in inharmonic frequencies to the power in the harmonic frequencies. The inharmonic frequencies are regions where the harmonics are not located (Kawahara et al., 2001). In practice, The upper (harmonics energy) and lower (aperiodic energy) spectral envelopes are used as estimates of the harmonic and inharmonic power spectrums, respectively.

The waveform generation is performed by the following steps:

1. Excitation pulses are generated in the time domain following the F0 trajectory.

2. Spectral envelopes are converted into the complex domain by assuming a minimum-phase realisation. Each spectral envelope corresponds to one generated excitation pulse.

3. Some delay needs to be added within each pitch cycle, as speech is not perfectly periodic (i.e., jitter) and the locations of GCIs are time shifted. This is implemented as linear phase rotation (group delay) in the frequency domain per each

pulse. Some degree of randomness is added to the group delay to generate jitter, thus minimising the "buzzy" sound.

4. The aperiodic components are synthesised by shaping white noise following the time-frequency aperiodicities parameter.

5. The generated components are added in the time domain in the locations of the corresponding excitation pulses.

Even though STRAIGHT was developed nearly two decades ago, it is currently the standard vocoder, because of its reliability over different pitch ranges, speaking styles, and speaker characteristics. Furthermore, its robust spectral envelope estimation makes it still one of the state-of-the-art vocoder systems.

1.3.2.2 GlottHMM

As opposed to several other methods that use pulse train and random noise as excitation signal (e.g., STRAIGHT), the GlottHMM method uses actual glottal pulses estimated from real speech data (Raitio et al., 2008, 2011). The glottal pulses are stored in a codebook and are organised gradually according to the speaking style, from *breathy* to *lombard*, so it adapts to the speaking style. As a difference with other vocoders, this method requires one extra stage in addition to the typical analysis and synthesis blocks.

Codebook Preparation
As mentioned a database of different glottal pulses need to be built from pre-recorded voiced speech:

1. The vocal tract filter is estimated by *Iterative Adaptive Inverse Filtering (IAIF)*, and is represented by LSF coefficients.

2. The residual of the signal is obtained by inverse filtering.

3. Glottal Closure Instants (GCIs) are extracted from the differentiated glottal flow signal using a peak picking algorithm.

4. The pulse segments are stretched to set their duration to 25ms each.

5. The glottal pulses are stored in a codebook and ordered by speaking style.

Analysis

1. Again, the vocal tract filter is represented as LSF coefficients, and estimated by the *Iterative Adaptive Inverse Filtering (IAIF)*.
2. The F0 is extracted from the glottal flow (residual) using the autocorrelation method.
3. The spectrum of the glottal flow is parametrised as LSF coefficients (source spectrum).
4. The Harmonic-to-noise ratio (HNR) is computed from the glottal pulses and used as acoustic parameter.
5. The gain for each frame is computed and used as parameter.

Synthesis

1. According to the HNR parameter, glottal pulses are selected from the codebook.
2. The glottal pulses are stretched (resampled) according to the F0 parameter.
3. The glottal pulses are concatenated to form a train of glottal pulses.
4. The spectrum of each glottal pulse is shaped to match the shape of the source spectrum parameter.
5. The signal is filtered by the vocal tract filter.
6. The signal is dynamically scaled according to the gain parameter.

A more advanced version of the vocoder is in Raitio et al. (2014a) and Raitio et al. (2014b), where the codebook is replaced by a deep neural network, which generates the glottal pulses directly.

1.3.2.3 Harmonic Model + Phase Distortion (HMPD)

Degottex and Stylianou (2013) proposed a method based on harmonic modelling aimed at parametric speech synthesis. Alongside the conventional spectral envelope and fundamental frequency ($F0$), this approach uses two extra parameters for phase modelling. One of its main benefits is that it avoids the use of a voiced/unvoiced decision stage, since the stochastic part of the signal is simply modelled as randomness in the phase of harmonics.

HMPD is based on the *Adaptive Harmonic Model (aHM)*, although adds a new method for phase modelling. In addition, an iterative algorithm is proposed to refine the estimation of harmonic frequencies, this is the called *Adaptive Iterative Refinement (AIR)*. The method can be summarized as:

1. aHM analysis is performed to estimate the instantaneous phase of the signal.
2. The minimum-phase component is computed and subtracted from the signal.
3. The local *Phase Distortion (PD)* is calculated.
4. The short-time mean and standard deviation of the PD are obtained and used as additional acoustic features.

According to aHM, the speech signal s within a frame is represented as:

$$s(t) = \sum_{h=1}^{H} a_h e^{j(h\phi_0(t)+\phi_h)} \tag{1.11}$$

Where H is the number of harmonics, a_h is the amplitude of the harmonic h, $\phi_0(t)$ is a real function that adapts the frequency basis of the harmonic model to the waveform frequency basis, and ϕ_h is the instantaneous phase.

The computation of the instantaneous phase takes into account the source-filter model, which comprises the phase of the source and the phase of the vocal tract filter (Degottex and Erro, 2014). It is assumed that the latter is minimum-phase, thus the phase of the vocal tract filter is subtracted from the signal. Then, the relative phase difference between the frequency components (harmonics) is computed, which is the called phase distortion (PD).

Once the PD is obtained for a considerable number of samples, its statistics (mean and standard deviation) are computed to take part in the feature structure. At the end, the speech is characterized by four feature steams: $F0$, *spectral envelope*, the mean of PD, and the standard deviation of PD.

During synthesis, sinusoids are generated for the whole utterance following the $F0$ and the amplitudes given by the spectral envelope. As these parameters are given per frame, they are linearly interpolated to the sample rate beforehand. Also, the phase distortion is applied to the sinusoids by delaying them according to the mean of PD, and the standard deviation of PD (Degottex and Stylianou, 2013).

1.3.2.4 AHOCoder

Ahocoder is the name of the vocoder developed by Erro et al. (2011) in the AHOLAB at the University of Basque Country. This is based on the *Harmonic plus Noise Model (HNM)* (Laroche et al., 1993). Its core is the method to obtain the HNM parameters from Mel-Cepstrum by using complex signal processing algorithms in frame-based processing. In its last modification (Erro et al., 2014), it uses *Quasi-harmonic Modeling (QHM)* to refine $F0$ trajectory estimation (Pantazis et al., 2010). It determines a *maximum voiced frequency* to separate stochastic and deterministic components of speech.

Analysis:

1. F0 is extracted by using the autocorrelation method presented in Boersma (1993).
2. The spectral envelope is extracted by linear interpolation of harmonic amplitudes present in the speech spectrum. Then it is converted into a Mel-Cepstrum (MCEP). The localisation of harmonics in the spectrum is performed by applying the QHM algorithm.
3. The maximum voiced frequency (MVF) parameter is extracted by a variation of the sinusoidal likeness measure (SLM) (Rodet, 1997).

Synthesis:

1. White noise is generated and passed through a high pass filter (HPF), whose cut-off frequency corresponds to the MVF parameter.
2. Sinusoids are generated and added to synthesise each harmonic, whose frequencies depend on the F0 parameter. The number of sinusoids to generate is given by the MVF, which is the maximum possible frequency of a harmonic.
3. The spectral envelope (MCEP) is imposed to the harmonics and noise by assuming a minimum-phase vocal tract filter.
4. The final synthesised speech is made up the addition of the noise and harmonics.

1.3.2.5 Deterministic Plus Stochastic Model (DSM)

Drugman and Dutoit (2012) and Drugman and Raitio (2014) use inverse filtering to extract the residual signal. The inverse filtering is applied with Mel-Generalised Cepstral (MGC) analysis and is modeled using *Principal Component Analysis (PCA)*. This method is described with the following steps:

1. MGC coefficients are estimated per frame, then the residual signal is obtained by inverse filtering.
2. $F0$ is extracted.
3. Glottal Closure Instants (GCI) are obtained by identifying the greatest discontinuity in the residual signal.
4. The frames are windowed using a Blackman window of two periods length and centred at the CGI.
5. To obtain the deterministic component, a *maximum voiced frequency* of 4kHz is used. Then, PCA is applied for dimensionality reduction and inter-feature decorrelation.
6. The stochastic component is computed using the HNM model, as in Stylianou (2001).

During synthesis, the stochastic and deterministic components are resampled and then filtered by using a *Mel-Log Spectrum Approximation(MLSA)* filter (Imai et al., 1983). Then, the frames are concatenated using pitch synchronous overlap-add (PSOLA).

The results show that the method works well in decreasing the buzziness effect, although the authors do not perform any comparison experiment with other state-of-the-art methods or vocoders.

Another model, similar to DSM, is proposed by Csapó and Németh (2013), in which the irregularity of speech is addressed. This effect is called *irregular phonation*, or *glottalization* and appears as abrupt changes in the fundamental frequency (F0), amplitude of the pitch periods, or both. The proposed model uses the parameters: fundamental frequency, RMS energy, locations of prominent values (peaks or valleys) in the windowed frame, and the Harmonic-To-Noise (HNR). As a result, the synthesised speech is perceived as more natural.

1.3.2.6 WaveNet Vocoder

As mentioned in Section 1.3.1.5, the WaveNet (van den Oord et al., 2016) has proved to deliver outstanding speech quality. Different types of features have been successfully used to condition the network. One of the possible conditionings is that acoustic features extracted by a vocoder or predicted by a neural network. Thus, the WaveNet would map acoustic features to raw waveform directly. This application is equivalent to what the synthesis block of a vocoder does.

Some studies (e.g., Tamamori et al., 2017; Shen et al., 2017) have shown success in using the WaveNet as a waveform generator and using acoustic features as input. However it still exhibits practicability issues, basically due to the computational burden that the architecture involves.

Nowadays, there is a whole set of neural vocoders, such as WaveRNN (Kalchbrenner et al., 2018), LPCNet (Valin and Skoglund, 2019), WaveGlow (Prenger et al., 2019), Clarinet (Ping et al., 2019), to name a few.

In spite of the benefits on using this new paradigm for speech synthesis, this topic is out of scope of this PhD thesis, given that they have been published only in the final stages of the work reported in this thesis.

1.3.3 Comparison of Vocoders

An experimental comparison of state-of-the-art vocoders was realised by Hu et al. (2013). The paper shows the comparison of the vocoders working in copy-synthesis mode including two speaking styles: normal and lombard. The description of the evaluated systems is shown in Table 1.1 and the results are presented in Figure 1.2.

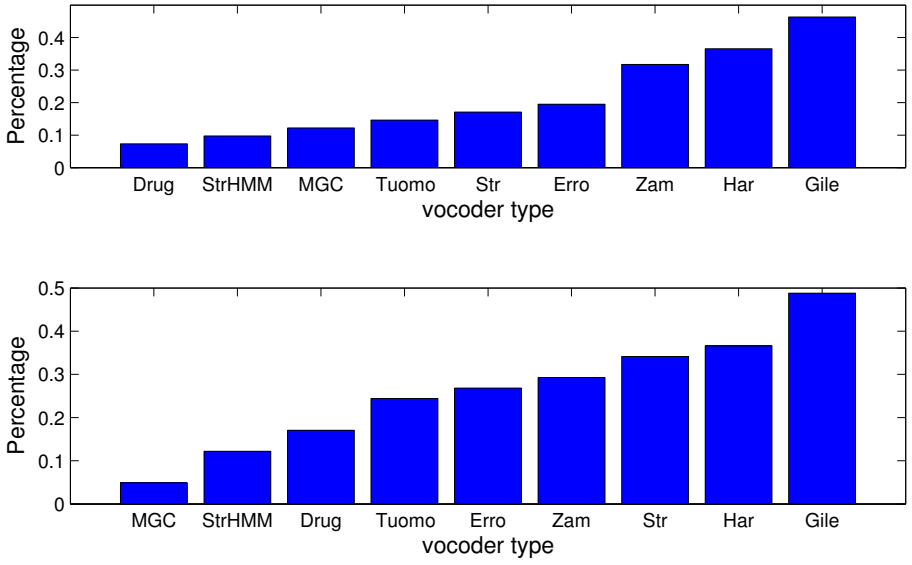

FIGURE 1.2: Preference test results extracted from Hu et al. (2013). Top: Normal, bottom: Lombard.

Name	Vocoder	Parameters per frame
Drug	DSM	MGC: 30 + f0:1, DSM for residual excitation
Erro	AHOcoder	MFCC: 40 + f0:1, Multi-band excitation
Gile	aHM-AIR[1]	2*k + f0:1, Harmonic excitation
Har	Harmonic Model (HM)	2*k harmonics + f0:1, Harmonic excitation
MGC	Mel-generalized cepstrum	MGC: 24 + f0:1, Pulse plus excitation
Str	STRAIGHT with full band excitation	aperiodicity:1024, spectrum: 1024 + f0:1, Multi-band Mixed Excitation
StrHMM	STRAIGHT with critical band excitation	Band aperiodicity:25 + MGC:39 + f0:1, Multi-band Mixed Excitation
Tuomo	GlottHMM	f0:1, Energy:1, HNR:5, Voice source LSF:10, Vocal tract LSF:30, Using natural pulse
Zam	HM[2]	2*k harmonics + f0:1, Harmonic excitation
Orig	Original Speech	–

TABLE 1.1: Description of the vocoders evaluated in Hu et al. (2013). [1]A previous version of HMPD, which does not perform PD analysis. [2]An HM with fixed number of parameters (harmonics). This makes it suitable for parametric speech synthesis.

The conclusions of the research can be summarized as:

- Sinusoidal vocoders give more consistent performance than source filter vocoders.
- Sinusoidal vocoders are perceptually distinguishable from source filter vocoders.
- Preference test comparisons indicate that sinusoidal vocoders can achieve superior synthesised speech quality.

On the other hand, Bollepalli et al. (2014) made a subjective comparison of several vocoder techniques for laughter synthesis. They experimented using copy-synthesis and HMM-based speech synthesis. The systems under analysis are described in Table 1.2.

System	Parameters	Excitation
MCEP[1]	mcep: 35 + F_0:1	Impulse + noise
STRAIGHT	mcep: 35 + F_0:1 band aperiodicity:21	Mixed excitation + noise
DSM	mcep: 35 + F_0:1	DSM + noise
GlottHMM	F_0:1 + Energy:1 + HNR:5 + source LSF:10 + vocal tract LSF:30	Stored Glottal flow pulse + noise

TABLE 1.2: Description of the vocoders evaluated in Bollepalli et al. (2014).

The results of this study can be summarised as:

- The four vocoders perform relatively well in copy synthesis.
- The speech quality produced by all the vocoders in HMM-based synthesis was significantly lower than in copy-synthesis.
- The methods MCEP and DSM that use simple and robust excitation modelling performed the best.

Another study (Babacan et al., 2014) evaluated the performance of four vocoders in the framework of HMM-based singing synthesis. It is different to speech synthesis because it presents: larger pitch and dynamic ranges, the source-filter interaction is stronger, and voiced sounds are sustained longer. Thus, this research can be seen as a more demanding evaluation than the ones carried out by studies on speech synthesis. The systems under assessment are described in Table 1.3.

Vocoder	Feature	Number of Params
Pulse	F0	1
	MGC coefficients	25
DSM	F0	1
	MGC coefficients	25
HNM	F0	1
	Mel cepstral coefficients	40
	Maximum Voice Frequency	1
GlottHMM	Energy	1
	F0	1
	HNRS	5
	Voice source spectrum	10
	Vocal tract spectrum	30

TABLE 1.3: Description of the vocoders evaluated in Babacan et al. (2014).

The results and conclusions from the subjective measurements can be summarised as:

- The best performance was attained by the HNM followed by GlottHMM and DSM.
- The MGC spectral envelope estimation is distorted by fundamental frequency information for high pitched signals.
- The spectrum exhibits interharmonics, whose magnitudes are comparable to the neighbouring harmonics.
- The evaluated vocoders are only capable of generating harmonics, but interharmonics could be relevant to achieve natural sound.
- The tested vocoders need improvements in their aperiodicity measurement method.

Although some of the results shown in these comparisons tend to show some benefits from using sinusoidal-based over source-filter model vocoders, we will use the STRAIGHT vocoder as a baseline to compare the systems proposed in this thesis. This is due to:

- Most of the recent research done on vocoders use STRAIGHT as a baseline to compare their proposed new vocoders. Arguably, it is the most used vocoder for SPSS (Toda and Tokuda, 2005; Raitio et al., 2011; Degottex and Stylianou, 2013; Drugman and Raitio, 2014; Degottex et al., 2018).
- STRAIGHT has shown to be robust to several conditions in SPSS, such as: gender, speaking style, noise, etc., e.g., Yamagishi et al. (2009); Erro et al. (2014); Bollepalli et al. (2014); Tamamori et al. (2017).
- In most of the papers presenting new vocoders that perform comparisons against STRAIGHT, the proposed methods are not undoubtedly better than STRAIGHT (Drugman and Dutoit, 2012; Degottex and Stylianou, 2013; Erro et al., 2014).

There are some related researches about spectral speech representations, from which we can name the method proposed by Maia et al. (2013), which attempts to represent speech by means of the complex cepstrum. However, it relies on strong assumptions and the system is not tested against any known baseline. Also, the work by Takaki et al. (2017) focuses on the direct prediction of FFT magnitude spectrum by a neural network. Nevertheless, as just the magnitude spectrum is defined, the phase spectrum need to be synthesised by using the Grifin-Lim algorithm (Griffin and Lim, 1984) yielding low speech quality.

Recently, other vocoders have been proposed and other comparisons among vocoders have been carried out. Among them, we highlight the good results achieved by the new GlottDNN (Airaksinen et al., 2018) and Pulse Model in Log-domain (PML) (Degottex et al., 2018) in the extensive evaluation of different vocoders performed by Airaksinen et al. (2018). However, as these studies have been published just recently, they are out of the scope of this thesis, but are mentioned here to give the reader an up-to-date overview of current potential vocoders.

1.4 Conclusion

Although several vocoding techniques have been proposed during the last decade, artefacts are still prominent in statistical parametric speech synthesis. Phenomena described as "buzziness", "phasiness" and "muffled sound" are still sources of unnaturalness. In fact, STRAIGHT is currently considered the standard reference to compare with, even though it was proposed nearly two decades ago. Its robustness and high quality have not been clearly outperformed by newer techniques. This shows that the improvements achieved by state-of-the-art waveform generators and vocoders are not sufficient, despite the large number of methods that have been proposed, and the high quality of the studies that evaluate them.

As mentioned, we can distinguish two main types of speech analysis/generation: source-filter decomposition, and harmonic models. Although the latter seems to achieve higher quality, they still fail especially in aperiodicity and phase modelling.

Both vocoding methods perform an extreme decomposition, which breaks the dependence of speech components. For instance, it has been shown that the source-filter separation degrades the resulting synthesised speech (Henter et al., 2014). Furthermore, even if we assumed that the relation between the source and the filter was a pure linear convolution, it would be very difficult to estimate the vocal tract filter. On the other hand, the assumption that the aperiodicity of speech signals can be modelled as a naive stochastic process totally independent of harmonic components, seems to be quite weak and a cause of artificial sound. In addition, the dependence of the phase with other components of speech is not well understood, and hard to model. Thus, attempting to build a naive phase model can be disadvantageous, leading to unnatural sound.

The source-filter and harmonic models have not been capable of substantially improving the quality of synthetic speech despite many refinements over a long period of time, implying that these techniques are probably reaching their upper bound in terms of perceived quality. This suggests that the waveform analysis and generation should be addressed from another point of view not making the assumptions and simplistic approximations applied by current methods.

This thesis is structured as follows: Chapter 2 presents a new waveform generation method that is not based on either source-filter model nor harmonic models, which intends to preserve the characteristics of natural speech by the modification of recorded speech signals for synthesis. The system shows that with simplicity, and least possible decomposition of speech signals, speech quality is improved. Chapter 3 describes a

whole new vocoder, MagPhase, inspired on the results obtained from the previous chapter. The vocoder encodes magnitude and phase spectrum in a simple way, achieving superior speech quality when comparing to the state of the art. Chapter 4 shows improvements to the original design of the MagPhase vocoder mainly in terms of efficiency. The use of the proposed vocoder in applications, other than SPSS, such as join smoothing in unit selection-based speech synthesis, speech recognition, and exemplar-based speech synthesis is presented in Chapter 5. Chapter 6 presents the final conclusions, including a general summary, future works and final remarks.

Chapter 2

Waveform Generation based on Signal Reshaping

2.1 Motivation

As stated in chapter 1, current state-of-the-art methods don't achieve substantial improvements for SPSS. We think that it signifies that this problem should be approached from another point of view. After performing some analyses and aggregating characteristics of the evaluated vocoding methods, we found some generalities, which are described in the following paragraphs.

2.1.1 Extreme Decomposition

The current methods, either source-filter separation or harmonic model-based, perform an extreme decomposition of the structures of speech, which seems to worsen the perceived synthesised speech quality.

2.1.1.1 Source-Filter Separation

For the methods based on source-filter separation, it is assumed that speech production is a linear time-invariant (LTI) process within an individual frame of analysis. Hence, they do not take into account some known properties of speech. For instance, the vibration of vocal tract walls affects the characteristics of the acoustic cavity (Hanna et al., 2012). Depending on the viscosity, voice intensity, and excitation frequency, the acoustics of the vocal tract change within an analysis frame.

Furthermore, it is well known that the shape of the vocal tract cavity changes at each glottal closure instant (GCI) and glottal opening instant (GOI) (Barney et al., 2007). This is caused by the movement of the glottal folds. When closed, the vocal tract is closed by the glottis, changing the frequency and phase response of the vocal tract. This process occurs once per pitch cycle, thus invalidating the assumption of invariability during an analysis frame. Therefore, classical methods that rely on this assumption, such as linear prediction, cepstral analysis, etc., are not able to deliver a good estimate of the vocal tract filter.

There are some techniques that model the change of the vocal tract shape every GCI/-GOI epoch, such as those of Raitio et al. (2014a), and Suni et al. (2010). However, the estimation of GCIs/GOIs, and the filters for the closing and opening segments, is very complex and requires several additional parameters to be tuned. Thus, due to the complexity of the task, these methods have shown to be suboptimal, and although they usually outperform the baselines, they do not achieve substantial perceptible improvement.

Unfortunately, even if the mentioned complications were solved or if speech production were truly an LTI process, the problem of filter estimation has still not been resolved. One of the main issues is the undefined number of degrees of freedom that filters present. The number of feed-forward (zeros) and feed-back paths (poles) must be predefined by the designer, who chooses these values according to his or her previous knowledge. However, it is impossible to be sure about these values. Moreover, it is difficult to define which zeros are located inside or outside the unit circle, affecting the phase response (e.g., minimum, maximum, or mixed-phase).

Generally, the criteria used to estimate poles are based on the minimization of the residual energy. These criteria are usually suitable for applications such as audio data compression or speech coding, but they were not conceived for the estimation of glottal source signals and vocal tract filter. Although several methods have been proposed to address this problem, their accuracy has not been confirmed due to the lack of reference glottal flow signals (Drugman et al., 2012).

2.1.1.2 Harmonic-based Models

Even though it has been shown that harmonic-based models produce higher quality synthetic speech than methods based on source-filter separation, the decomposition carried out is suboptimal due to:

- Models based only on sinusoids cannot accurately represent the stochastic components of speech, which are therefore perceived as "musical artifacts". Unfortunately, stochastic components are always present in speech signals, especially at high frequencies, even during clean vowels. Thus, this shortcoming is quite relevant and cannot be neglected.

- One solution is the use of random noise as source of the stochastic components of speech. This is the called Harmonic plus Noise Model (HNM), which estimates the characteristics of the random noise along with the harmonics. Usually, this task is simplified by defining a maximum voiced frequency (MVF) to constrain the random noise to higher frequencies than MVF, and harmonics to lower frequencies. This boundary, the MVF, is commonly fixed at around 4kHz, although some methods dynamically estimate it over time. It is an oversimplified model, since it is well known that harmonics and stochastic components overlap in a more complex fashion covering a wide range of frequencies. Some methods address this issue by describing the mix of deterministic (harmonics) and stochastic components in the frequency domain. Several representations have been proposed, for instance, aperiodicity measurement (e.g., STRAIGHT (Kawahara et al., 1999b)) is expressed as a magnitude spectrum difference, or phase distortion (e.g., HMPD (Degottex and Erro, 2014)) that models the degree of phase randomness in the sinusoids.

Despite the good results obtained with these techniques, unnaturalness persists, which is usually perceived as "musical artefacts" or "separated sound sources". It is obvious that both deterministic and stochastic components of speech are produced by the interaction between the air flow and the vocal folds. Therefore, these components may share statistical dependence, and be highly correlated. Still, there is not any harmonic-based method that accounts for this, thus the resulting speech is perceived as two different sources additively mixed rather than a single process of sound generation.

The extreme decomposition of speech signals seems to be inconvenient for accurate speech modelling. There are several poorly understood phenomena of speech production, thus the simplistic assumptions used to address them seem to degrade the resulting synthesised speech rather than being beneficial.

2.1.2 Common Characteristics of Speech Production

Usually, a high number of acoustic parameters is used in statistical speech synthesis, since speech generation, as a complex process, requires many features to be modelled, such as fundamental frequency, spectral envelope, aperiodicity measurements, etc. Nevertheless, if we compare and observe some examples of speech signals we know intuitively that there are some characteristics shared by different phonemes and speakers.

It is worthwhile to study and question the complexity of aperiodicities that are usually represented by a high number of frequency bands, which may be excessive. It is not clear what the functionality of aperiodicities is in terms of speaker differentiation or how these vary among different instances of the same phoneme uttered by the same speaker. That suggests that the high dimensional aperiodicity measurements could eventually be removed or replaced by a simpler representation for either acoustic modelling or vocoding. Perhaps, only one low dimensional aperiodicity parameter is necessary to define the general aperiodicity characteristics of a speaker or speaking style. This topic will be studied in Section 2.2.

2.2 Preliminary Experiments with State-of-the-art Vocoders

After the comprehensive literature review summarised mainly in Section 1.3, our next step was performing a preliminary evaluation and experimentation with state-of-the-art vocoders. The aim of it is to have a personal insight of the speech quality derived by the vocoders, and practical experience running the tools.

In order to test the vocoders, statistical parametric speech synthesis systems HTS (Zen et al., 2007) and a DNN-based system called Merlin (Wu et al., 2016) developed at CSTR, along with copy-synthesis were used. Also, recorded utterances as described in Section 2.5 were employed as test samples.

The experimentations with the vocoders that showed the best performances are summarized in the following paragraphs:

- STRAIGHT

  The standard system STRAIGHT produced quite satisfactory results in terms of quality and robustness. Overall, its performance remains consistent either in copy synthesis or in statistical speech synthesis. That is, even though it does not achieve the best performance in copy synthesis when compared to the other vocoders under test, it does show a more consistent performance in both situations: using generated parameters (HMMs, DNN) or those extracted directly from natural speech. The principal perceivable drawback of this vocoder is the "buzziness" that it produces, which was present in all of the evaluation sentences.

  During our analyses, we observed that the aperiodicity measurements (spectrums) exhibit a rather simple shape resembling to the spectrum of white noise passed through a simple high-pass filter, as seen in the Figure 2.1. Furthermore, if these aperiodicity measurements are predicted by an acoustic model (i.e., DNN), their shape would look even more smooth and simple. That suggests that the whole representation could be approximated by just one value, the "cut-off frequency" of the high-pass filter.

  Also, we observed a high correlation between the F0 contour and the time varying "cut-off" frequency of the aperiodicity spectrogram. Figure 2.2 shows this, which also suggests that the whole aperiodicity spectrogram can be approximated by only using the F0 contour as a guide.

- HMPD

  It is known that, in general, methods based on harmonic models achieve superior quality when compared with vocoders based on source-filter decomposition. Thus,

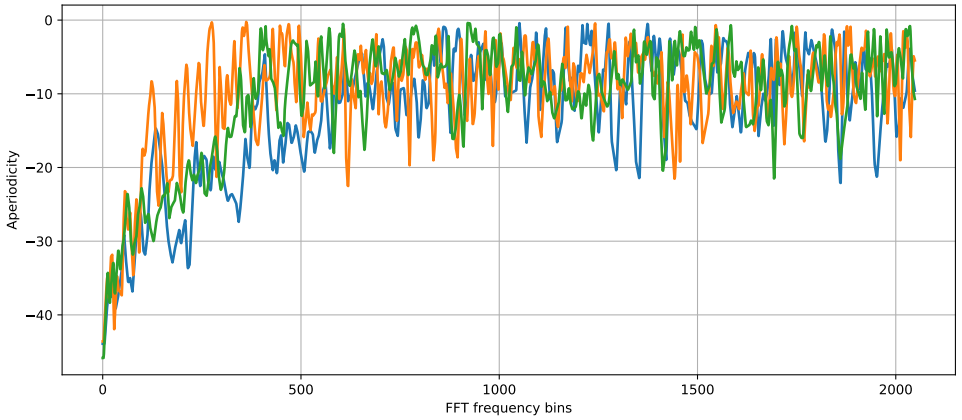

FIGURE 2.1: Three examples of aperiodicity spectrums taken from different voiced speech instances uttered by the same speaker. It is clear that they look as a high-pass filtered white noise. All aperiodicities extracted by STRAIGHT.

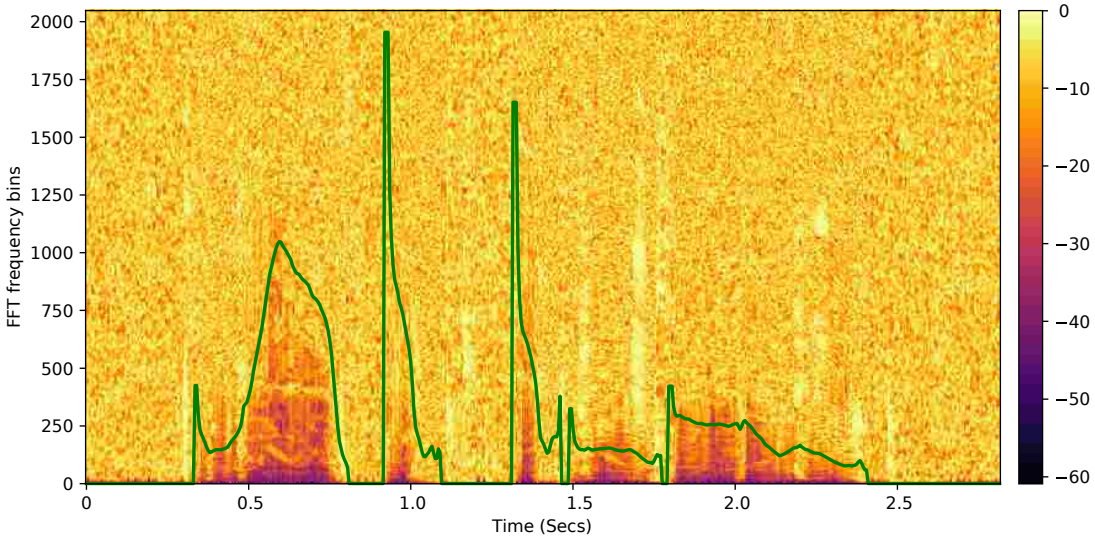

FIGURE 2.2: Example of aperiodicity spectrogram in a utterance. Green curve: Corresponding F0 contour (scaled for illustration purposes). All acoustic features were extracted by STRAIGHT.

HMPD shows superior quality in copy-synthesis. In fact, the reconstruction of the signal from parameters extracted from natural speech can be considered as virtually perfect in most test utterances.

Nevertheless, if the parameters are estimated rather than extracted, HMPD generates artefacts which are mainly perceived as "musical" noise at high frequencies, where the aperiodic components are localised. We strongly believe that these artefacts are produced by the vocoder rather than the acoustic modelling, since the amplitude modulation of the "musical" noise runs at a higher rate, and the parameter generation process ensures very smooth feature trajectories (actually sometimes over-smoothed). Furthermore, this type of artefact is typical of some vocoders.

- GlottHMM

  Although GlottHMM is based on a quite complex and detailed decomposition that models the real acoustic process of speech production, the resulting perceived quality is not close enough to HMPD and STRAIGHT, at least in this informal experimentation. For the test utterances with both parametric speech synthesis and copy-synthesis, this vocoder delivers high quality synthetic speech, but producing a "vocal fry" sound. High frequency components of "buzziness", typical of source-filter vocoders are removed, but some similar artefact is added in the mid frequencies. The modest quality shown by this vocoder in this experimentation can be attributed to the high complexity involved in expert tuning of parameters.

Overall, the perceived quality produced by the analysed vocoders correlates with the studies reviewed in Section 1.3.3, and the conclusions presented in section 2.1.1. The methods based on harmonic models outperform the ones that apply source-filter separation, and the extreme decomposition seems to be the main factor of degradation. It is worth mentioning that according to these informal tests, HMPD seems to produce the best speech quality, and STRAIGHT shows a very robust performance.

2.3 Objectives

After reviewing the literature and experimenting informally with the state-of-the-art vocoders, we observe that:

1. Vocoders have not achieved significant improvement during the last decade.
2. Vocoders apply extreme decomposition to the structures of speech.
3. Dependency between stochastic and deterministic processes of speech production has not been modelled.
4. Correlation between F0 and the aperiodicity spectrum has not been exploited to improve sound quality and dimensionality reduction.
5. Many processes of speech production are not well understood, and so are approached by simplistic inaccurate models.
6. Harmonic-based models sound better than methods that perform source-filter separation.
7. The most complex methods perform worse than the ones that use simple and robust excitation modelling.

Therefore, the method that is proposed in this chapter should meet the following requirements:

a) Avoid performing unnecessary or extreme decomposition of speech, such as source-filter separation, or stochastic plus deterministic modelling.

b) Take into account the observed correlation between F0 and aperiodicity spectrum shapes.

c) Focus the design into making a good method for parametric speech synthesis rather than an excellent "speech codec" for copy-synthesis.

Thus, at the end of this work, it should be possible to claim that the proposed method:

- Starts a completely new class of waveform generators, that does not fit into harmonic-based models or source-filter separation.
- Achieves superior sound quality in parametric speech synthesis.
- Uses fewer parameters than conventional techniques.

2.4 Desirable Characteristics

In order to conceive the proposed method, the first objective is to meet requirement a), which states the avoidance of making unnecessary decompositions of speech signals. However, is there any way to synthesise speech without making extreme decompositions? This question is one of the most important problems that this work has to answer.

Natural speech has all the characteristics and high quality that we require for speech synthesis, but it cannot be used directly to synthesise, since it is impossible to build a speech database containing infinite combinations of prosody, message, etc. Hence, in order to cover all of these cases, two different approaches are being used currently: statistical modelling of speech features, and the concatenation of natural speech audio units. The ideal method should be based on natural speech signals as much as possible, avoiding decompositions and meeting the required acoustic parameters by some other means. Table 2.1 shows a comparison of the basic characteristics of the proposed and classic methods.

Our main goal here is to use a stored natural speech signal (the base signal) and "reshape" its characteristics to match the predictions from an acoustic model; we aim to achieve this with the least possible modification.

Phonemes can be broadly classified into voiced and unvoiced. In order to constrain the analysis, we focused only on voiced speech for the first stage of research. Then, we need to define the characteristics of the speech signal that will be reshaped, which should present the closest features to the target speech. Thus, the signal to be reshaped, the so called "base signal" must:

	Methods		
	Vocoding	**Concatenative**	**Proposed**
Signal origin	Artificial	Natural	Natural
Generation	Synthetic ensemble	Concatenation of audio clips	Adaption of natural speech
Easiest task	Meet required prosody, text	Meet natural acoustic properties	Meet natural acoustic properties
Hardest task	Meet natural acoustic properties	Meet required prosody, text	Meet required prosody, text

TABLE 2.1: Comparison of the basics of the methods based on vocoding for statistical parametric speech synthesis, concatenative, and the proposed method (reshaping).

- *Be voiced speech*
  Since the objective of this first stage is to synthesise voiced speech, the base signal must be of this type to minimize the modifications applied by the reshaping process.

- *Keep its acoustic properties as constant as possible*
  The more stable the acoustic properties are, the easier is its analysis and modification, thus it is less probable to generate artefacts.

- *Be as long as possible*
  If the audio clip is long enough to cover whole words or phrases, that would avoid the concatenation of audio clips within words, preventing audible joins.

The perfect candidate to meet the desirable characteristics for the base signal is recorded speech. To obtain this, some recording sessions were carried out.

2.5 Recordings

The signals were recorded in a hemi-anechoic chamber with professional audio equipment. One male and one female speaker were recorded at 96kHz, 24bits with 2 microphones: One small-diaphragm condenser and a headset microphone were used to have more options to choose from. Then, the signal recorded by the small-diaphragm condenser mic was chosen, due to cleaner characteristics of the signal. The recordings included:

- *Phonemes*
  The speakers were asked to utter 5 different phonemes /a/, /e/, /i/, /o/, /u/, keeping the sound properties as stable and as long as possible. Also, they uttered each phoneme at 3 different pitches: One normal, that is as they would deliver in normal speaking; one at a higher pitch; and one at a lower pitch. The speakers decided the specific pitches according to how they felt comfortable. These recordings are intended to be used as base signals, which will be reshaped to meet acoustic requirements during synthesis. It is worth to emphasize that we have recorded several signals for experimentation purposes. However, the system will work with only one base signal at a time.

- *Utterances*
  Each speaker pronounced 2 utterances in 2 different intonation styles; normal and happy. Later, these utterances were used to test the system by performing copy-synthesis.

2.6 Proposed Approach

For simplicity, the first design of the proposed method was focused on the generation of voiced speech only. The general diagram describing this approach is depicted in Figure 2.3. The following subsections will provide detailed explanations of the processes in the diagram.

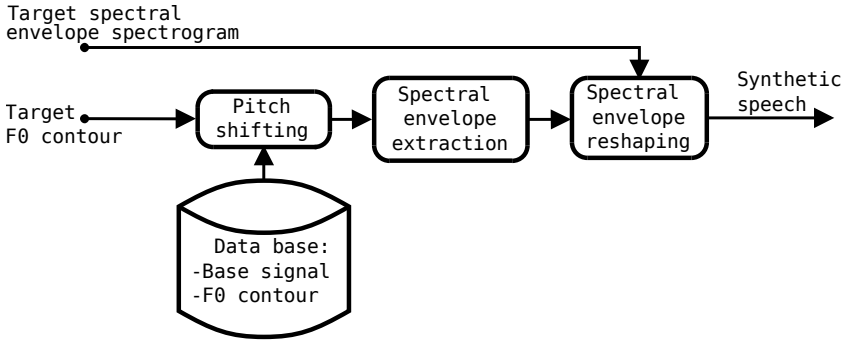

FIGURE 2.3: Block diagram of the basic proposed system (voiced speech only).

2.6.1 Acoustic Parameters

In the input of the proposed waveform generator, we have acoustic parameters, either predicted by an SPSS system (acoustic model), or directly extracted from natural speech by a feature extractor (i.e., STRAIGHT). As the system needs to reshape the characteristics of a *base* signal, it will need to change the F0 contour and the spectral structure of it. This implies that it will need at least the target F0 contour and the target spectral envelope as input parameters.

Aperiodicity measurements are acoustic parameters commonly used in speech synthesis. Therefore, their suitability for the proposed framework has to be considered. We have observed that the spectral shape of aperiodicity measurements is highly correlated with the fundamental frequency, as explained in Section 2.2. Thus, we decided to not incorporate aperiodicity measurements into our proposed system, which we expect will be naturally shaped by the pitch fluctuations.

2.6.2 Pitch Shifting

The database is made up by **just one base signal, plus its previously extracted F0 contour**. As mentioned, the base signal presents approximately constant F0 and spectral structure as seen in Figure 2.5 (a). The first stage on the process consists of modifying the base signal, such that its F0 contour matches the target F0 contour.

There are several pitch shifting techniques that may be suitable, including ones based on a phase vocoder, overlap and add (OLA), pitch synchronous overlap and add (PSOLA), etc.

In general, the listed techniques are intended to change the pitch whilst keeping the original duration of acoustic events and/or spectral structure of the signal. However, this is not a must for the proposed system at this stage, since the duration of acoustic events is not intended to match the target parameters yet. Also, the spectral structure will be imposed by a posterior stage on the process, thus it is not necessary to care about it here. However, it is critical for the proposed method to keep the natural characteristics of the base signal as much as possible. Consequently, pitch shifting based on nonuniform resampling is applied, which is the simplest, most natural and cleanest technique for pitch shifting, and ensures the absence of audible artefacts. By using nonuniform resampling, we are able to shrink/stretch the audio signal dynamically over time to meet the target F0 contour, and avoiding audible artefacts.

Nonuniform audio resampling is the resampling of the audio data by a time-varying resampling ratio, which is updated at every time step. As a result, the playing speed of the base signal is warped, and thus its pitch. This effect is perceived as changing the speed of rotation in a turntable ("scratching"). As a consequence, not only the pitch is modified, but also the spectral structure and aperiodicities. However, as mentioned earlier we don't need to care about the spectral structure, since it is something that will be reshaped at a subsequent stage. For the aperiodicities, we expect that these will be naturally shaped by the variation of F0, as explained in Section 2.2.

Still, some critical factors must be taken into account:

- *Sample rate*
  When increasing the pitch of the base signal, aliasing will be inevitably produced, since the high frequency components will fold to lower frequencies if they exceed the Nyquist frequency. To avoid this issue in a simple and efficient fashion, the output sample rate was set to the half of the recordings' sample rate, i.e., 48kHz.

  Hence, even though aliasing is produced in case of upsampling, the folded frequencies will be placed over 24kHz, making them inaudible.This only holds if the output (target) pitch is not greater than twice the original pitch (base signal).

- *Signal length*
  In addition, when increasing the pitch, the signal gets shorter and vice-versa. Thus, it is important to care about the length of the base signal, since it should be capable of covering voiced segments in their entirety.

Ideally, the nonuniform resampling should be applied by interpolation using the system's analog impulse response (i.e., sinc function) as kernel. However, it is easier and more efficient to use *spline* interpolation, which achieves a good approximation. After the nonuniform resampling, the signal is passed through an anti-aliasing filter to be finally downsampled to the target sample frequency (e.g., 48kHz). Figure 2.4 illustrates an example of pitch shifting, where you can see the high accuracy of the method. Also, Figure 2.5 shows the spectral structure of the base signal before (a) and after (b) pitch shifting. It is worth to notice how the formants and valleys are warped following the new F0 contour and the attenuation of high frequencies due to the anti-aliasing filter.

2.6.3 Spectral Envelope Reshaping

Even though the original base signal has a spectral envelope approximately constant over time, after applying pitch shifting it gets warped following the target F0 contour as seen in Figure 2.5 (b). Since the base signal is already pitch shifted, the next step is to reshape its spectral envelope spectrogram to the target spectral envelope spectrogram.

In order to perform the reshaping of the spectral envelope spectrogram, we firstly need to estimate the spectral envelope spectrogram of the pitch shifted base signal, which as seen in 2.5 (b) presents a distorted shape. We use STRAIGHT to perform the feature extraction. Then, the difference between the target spectral envelope spectrogram and the pitch shifted base signal's spectral envelope spectrogram in Log-domain is computed. This describes the modification that has to be applied on top of the pitch shifted base signal. Usually, the spectral envelope, as well as the fundamental frequency are provided at a constant rate frame-by-frame (e.g., by a TTS acoustic model).

The adaptation of spectral envelope can be achieved by applying some type of filtering to the pitch shifted base signal. For our first implementation, the spectral difference

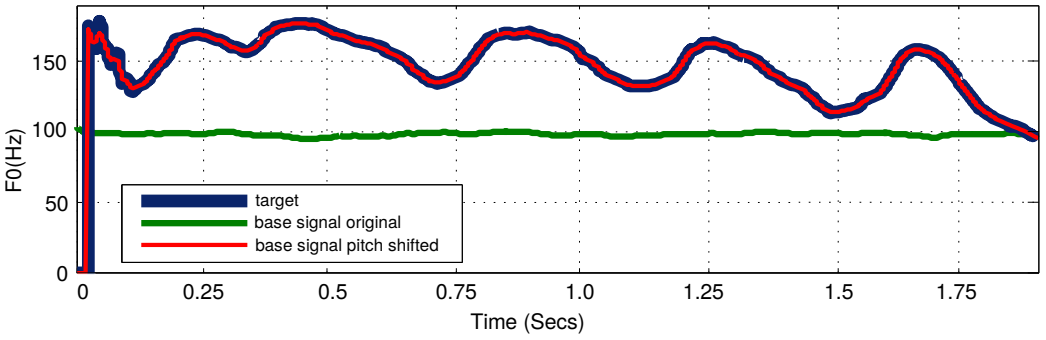

FIGURE 2.4: Example of pitch shifting of the base signal. Blue: Target F0 contour extracted from the utterance "We were a year ago." Green: Original F0 contour of the base signal. Red: F0 contour extracted from the base signal after pitch shifting.

was converted to time domain by using the Inverse Fast Fourier Transform (IFFT). Then, the resulting linear-phase impulse response is convolved with the signal on a frame-by-frame basis (constant frame rate), and then the resulting filtered frames are overlapped and added (OLA). The impulse response of the filter dynamically changes frame by frame according to the spectral difference. As a result, it produces synthesised speech at the output (Figure 2.5 (c)).

2.6.4 Extending the proposed Method to Unvoiced Speech

As mentioned, the system previously described only works for voiced speech, that is vowels, nasals, etc. The next step is to provide support for any type of phoneme to be able to synthesise complete utterances. The diagram in Figure 2.6 shows the complete proposed system which is capable of synthesising any type of speech waveform, or silence.

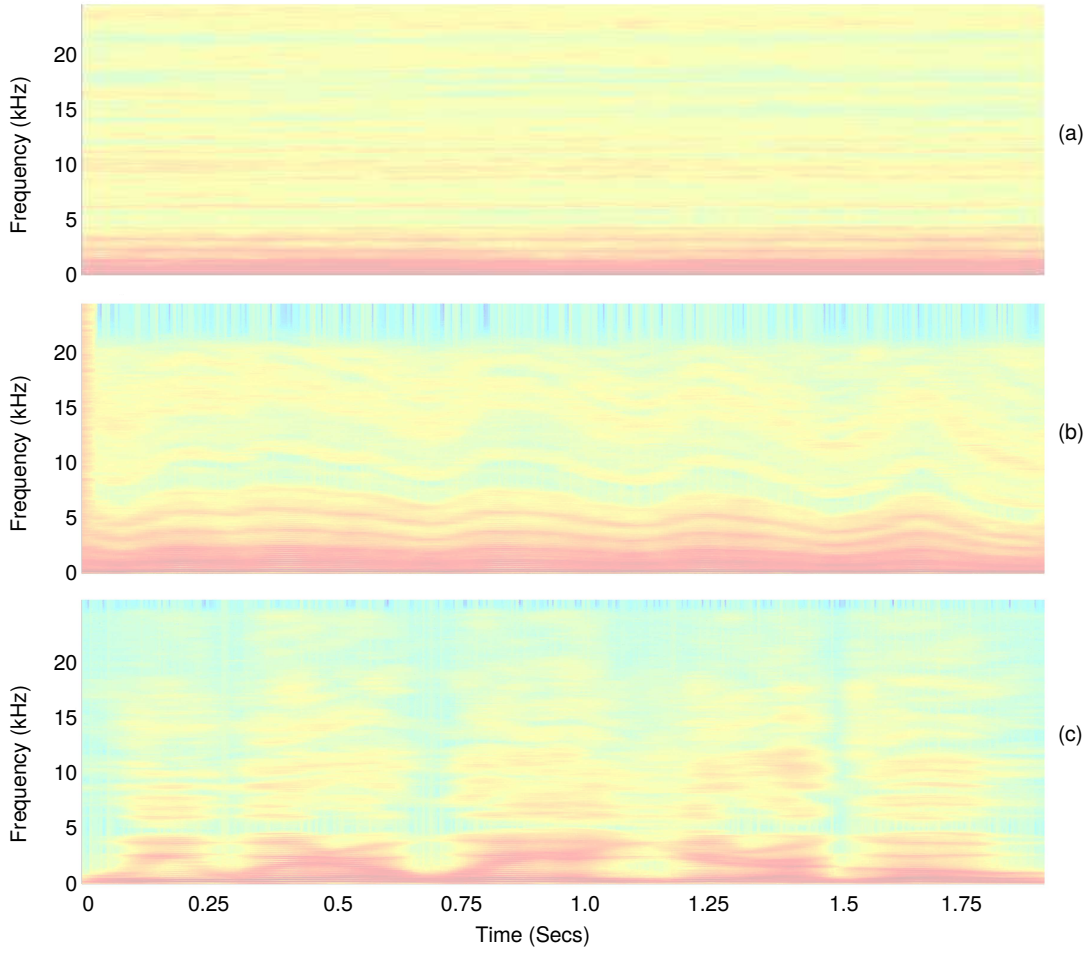

FIGURE 2.5: Log-magnitude spectrograms in different stages of the waveform generation process for the target utterance "We were are a year ago." (a) Original base signal (phoneme /a/). (b) Base signal after pitch shifting. (c) Synthesised speech, i.e., base signal after pitch shifting and spectral reshaping.

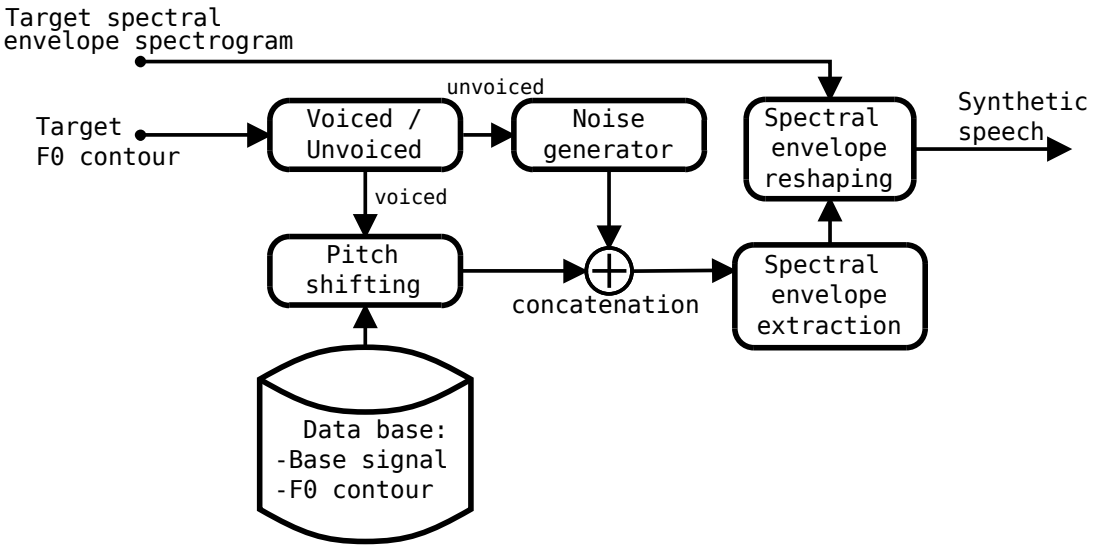

FIGURE 2.6: Block diagram of the complete proposed system capable of synthesising voiced and unvoiced speech.

The acoustic features are segmented into voiced and unvoiced segments according to the voicing decision derived from the target F0 contour, as seen in the example of Figure 2.7. The segments are variable length, from one to several frames each. For each voiced segment, the pitch shifted base signal is generated applying the pitch shifting process described in Section 2.6.2. For unvoiced segments, the base signal is generated by a white noise generator. Then, the energy of the segments is normalised to keep constant energy through the whole utterance, thus the segments can be concatenated to form a voiced/unvoiced base signal. Finally, the spectral envelope of this new base signal is reshaped by using the method described in Section 2.6.3.

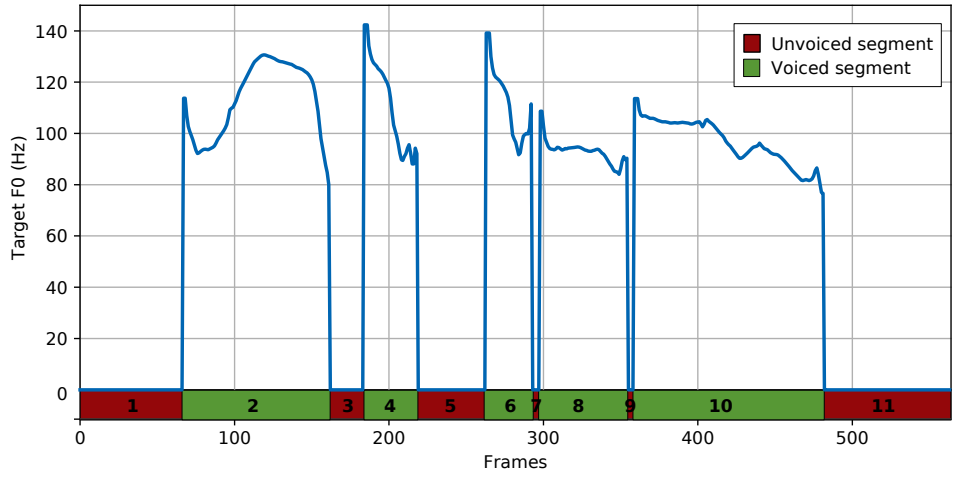

FIGURE 2.7: Segmentation of a utterance into voiced and unvoiced segments. In this example 11 segments were identified.

2.7 Improvements

Some modifications were implemented to improve the quality delivered by the waveform generator. Also, the method was used in other applications other than SPSS. The following subsections describe these modifications and new applications.

2.7.1 Modifications to the Spectral Envelope Reshaping

Although the adaptation of the spectral envelope by using the dynamic filter explained in section 2.6.3 is quite effective, it still produces some unwanted artefacts, which can be perceived as "metallic", "phasiness", or "ringing" effect. Thus, we attempted other ways to perform the spectral envelope reshaping to remove the unwanted artefacts.

2.7.1.1 Filter Modifications

Several filters were implemented, intended to improve the perceived quality of synthesised speech, reducing the artefacts generated by the filter plus OLA described in Section 2.6.3. The first hypothesis is that OLA is suboptimal, since in most of the cases the signal reconstruction is not perfect. The accuracy of the analysis/synthesis by using OLA relies on several factors such as the chosen window type, hop length, window size, phase coherence, etc. Moreover, since the filter changes frame by frame, OLA distorts the phase coherence between consecutive frames, which translates into more artefacts.

Dynamic Convolution Filter

This method consists of convolving progressively modified impulse responses with each audio sample of the signal, in the time domain. Thus, the process accurately approaches to how a filter progressively evolves over time, smoothly, sample-by-sample. It was necessary to interpolate the differential magnitude spectrum over the time axis to generate the values for each frequency bin, sample-by-sample. Hence, a different spectral envelope is obtained for each audio sample. Then, linear-phase impulse responses are generated from them using IFFT. Finally, these are convolved with the signal in the time domain, sample-by-sample. As a result, the "phasiness" or "metallic" sound is reduced compared to the original filter design, but is still present.

Dynamic Convolution Filter with Pitch-Synchronous Coefficients Update

Another filter design was attempted based on the previous *Dynamic Convolution Filter*. The algorithm was modified to make the impulse responses change at every pitch cycle, rather than at each audio sample. The filter coefficient updates are synced with the pitch periods, hopefully hiding the perceptible artefacts produced as a result of the continuously varying filter.

For its implementation, pitch cycle locations were derived from the F0 of the base signal. Then, epoch locations were placed on the highest value of the waveform on every pitch cycle. It was a rather easy and robust task due to the steadiness of the characteristics of the base signal. All of this operation was done beforehand, off-line.

During synthesis, this time the *Dynamic Convolution Filter* is applied updating its coefficients on the epoch locations, instead of at every audio sample. In spite of the optimism on this new approach, it added a new artefact to the synthesised speech, generated by abrupt changes in the spectral envelope, which was perceived as "granularity". This artefact was more severe than the "phasiness" produced by the previous filter methods.

Pitch-Synchronous Overlap-Add (PSOLA)

The update of filter coefficients at pre-marked epoch locations did not result in good speech quality, since it generated abrupt changes in the spectral envelope. Hence, PSOLA was applied to smooth these abrupt changes that occur once per cycle. Thus, the update of filter coefficients would be smooth while keeping the synchronisation with pitch cycles. However, this method was not able to reduce the "granularity" produced by the previous approach.

Mel-Log Spectrum Approximation(MLSA)

The MLSA filter (Imai et al., 1983) is widely used in speech synthesis applications, due to the good quality it can achieve as a time-varying synthesiser filter, low complexity, and its capability of synthesising speech directly from Mel-Cepstra. It is an IIR filter exhibiting a minimum-phase response. We tried the MLSA filter implementation included in the Speech Signal Processing Toolkit (SPTK[1]). As a result, we found that the MLSA along with the *Dynamic Convolution Filter* were the ones that achieved the

[1]Available at: http://sp-tk.sourceforge.net/

best quality, overall. Still producing some "phasiness", but acceptable compared to other vocoders.

2.7.1.2 Phase Analysis

Despite the good results obtained with the described filtering techniques, the method still generates artefacts, perceived as "phasiness"or "chorus effect". One strategy to address this issue is to analyse the phase distortion produced by the filter. FFT based spectrograms were used to analyse the signal at the input of the filter (pitch shifted base signal + noise) and at the output (synthesised speech).Then, by computing the phase distortion, which is basically the difference between the two phase spectrograms, we could observe how the filter distorts the phase, so generating the unwanted artefacts. An example of the analysis of phase distortion is shown in Figure 2.8.

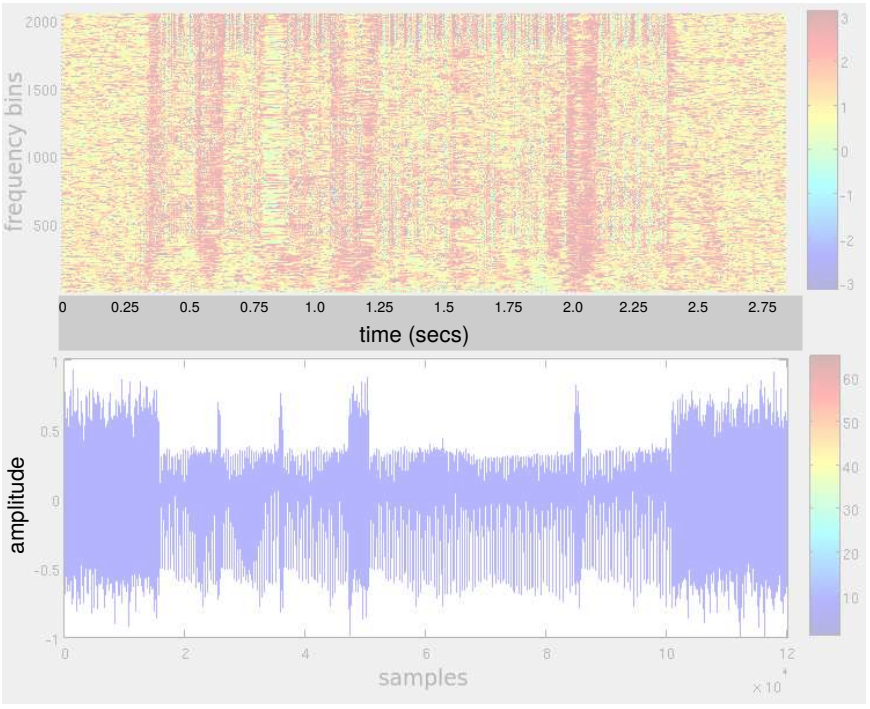

FIGURE 2.8: Phase distortion example. Top: Phase distortion spectrogram. Bottom: Input signal in the time domain.

From this analysis, some interesting observations can be made:

- No matter the type and characteristics of the dynamic filter, it always produces phase distortion:

  We tried different filter configurations and types as described in Section 2.7.1.1 with several utterances, and always they generated phase distortion as the one shown in Figure 2.8 (Top).

- The more complex the frequency response of the filter is, the more phase distortion is produced. We observed that complex filters, the ones that exhibit abrupt changes in the frequency response (e.g., a large and narrow resonances), notably produce higher phase distortion.

- The faster the filter changes over time, the more phase distortion is generated:

  As the filters dynamically evolve over time, we assessed how much the speed of the variation affects the phase distortion. Hence, we tried different filters evolving at different speeds. As a result, we observed that the slower varying filters produced less phase distortion.

- In the extreme case that the filter is zero-phase[2] and the values of its coefficients remain constant over time, some phase distortion is still produced:

  We tried several zero-phase filters, whose response keep constant during the whole utterance (i.e., no time-varying), and phase distortion measured from the FFT spectrogram (as in Figure 2.8 (Top)) is still produced. This contradicts what we know about zero-phase filters, which shouldn't distort the phase of the signal.

- Phase distortion increases in spectro-temporal regions with low magnitude spectrum: We observed that there is some correlation between phase distortion and magnitude spectrum. That is frequency bins with lower magnitude spectrum (before and/or after filtering), tended to exhibit more phase distortion.

Therefore, we can conclude that the phase distortion shown by spectrograms is due to two factors:

1. The FFT-based spectrograms themselves contain "errors" in measuring the phase spectrum. It is guaranteed that a linear phase filter, with proper delay compensation, does not introduce any phase distortion. Therefore, for the case of the linear-phase filter with constant coefficients, the only possible cause of the apparent phase distortion is that it is introduced by the analyser (spectrogram).

2. Rapidly changing filters generate more phase distortion than slowly evolving filters.

3. Because of the inaccuracy of spectrograms, we cannot quantify and describe the phase distortion produced by dynamic filters at this point in our research.

The fact that spectrograms appear to reveal "errors" is a consequence of doing FFT analysis, and specifically is an effect of the use of an analysis window. The spectrum

[2]Zero-phase filter: Linear-phase filter whose phase slope is 0.

of an analysis window has a certain width, which distorts the values corresponding to adjacent frequency bins. This mainly affects frequency bins with lower magnitudes.

However, although it was observed that spectrograms show distorted phase values, it is still possible to perfectly recover the original signal from the spectrogram. That means that what we call "errors" in phase calculations are actually not errors from the mathematical point of view. That is, although the values of phase shown by spectrograms do not accurately represent the signal, they are perfectly accurate for the sinusoidal model of the Fourier theorem, even in an OLA framework.

2.7.2 Use of Natural Speech to Synthesise Unvoiced Phonemes

As explained in Section 2.6.4, unvoiced speech segments are synthesised by using random noise as base signal. However, similarly to the case of voiced speech, we thought that real speech recordings are more likely to deliver a natural sound.

Hence, some unvoiced phonemes were recorded by a male speaker (/f/, /s/, /ʃ/). While recording, the speaker was asked to keep the acoustic characteristics stable for as long as possible. Thus, three audio clips (one of each phoneme) were selected to form a small database. The magnitude spectrogram is computed for each of the three candidates, then, the average of the magnitude spectrogram of each of the three units is computed and stored as well.

When synthesising unvoiced segments, the Euclidean distance between the average of the magnitude spectrum of the target spectral envelope and the average of the magnitude spectrum of each of the three candidates is computed. Then, the unit candidate with the minimum euclidean distance to the target spectral envelope is the one chosen as a base signal.

Finally, its spectral envelope is shaped to match the target by following the process described in Section 2.6.3.

2.7.3 Modification of Aperiodicities using a Comb Filter

Several different methods were tried to modify aperiodicities of voiced segments. They were based on the idea that the energy-localised between harmonics, in the frequency domain, would correspond to a stochastic process. Thus, one way to modify it is by applying some comb-shaped filter that would adjust the energy ratio between the energy of harmonics (deterministic), and the energy between them (stochastic). That would control the degree of "randomness" of speech at different frequency bands.

After several trials using different implementations of comb-shaped filters, the results were not satisfactory, and the quality achieved was far from being perceived as natural.

2.7.4 Mel-Frequency Smoothing

Any target spectral envelope derived via MCEPs has reduced resolution at higher frequencies. On the contrary, the spectral envelope of the base signal is full resolution at all frequencies. This mismatch in resolution was addressed by applying Mel-scale smoothing to the base signal spectral envelope. This makes the spectral subtraction more consistent, ensuring that the data to be processed share the "same domain".

2.7.5 Spectral Enhancement

Spectral envelopes tend to be over-smoothed because of the extraction method and/or statistical modelling. To alleviate this, target log spectral envelopes are raised to a power greater than 1 (e.g., 1.1) to enhance peaks.

After analysing all the potential improvements to the original design, we chose the *Use of Natural Speech to Synthesise Unvoiced Phonemes*, *Mel-Frequency Smoothing*, and *Spectral Enhancement* to be part of the final waveform generator. Thus the complete diagram of the waveform generator employing the new improvements is shown in Figure 2.10.

2.7.6 Pitch Shifting - Spectral Envelope Modification Swap

The system so far, as depicted in Figure 2.6 is impractical, since it needs to extract the spectral envelope of the base signal during synthesis (runtime), which is computationally expensive. Thus, some modifications to the method were applied to make it more efficient.

The generation of voiced and unvoiced segments was totally separated in two different blocks (left and right panels in Figure 2.10). For voiced segments, the stages pitch shifting and spectral envelope modification were swapped, in order to avoid performing the spectral envelope extraction during runtime. To do so, it was necessary to add new steps:

1. *Time-Frequency Stretching of Target Spectral Envelope*

   We need to modify the target spectral envelope spectrogram so that its underlying F0 contour matches the one from the base signal. This is carried out to

"pre-correct" for the posterior pitch shifting that will be applied in a later stage. It is done by moving each frame of the target spectral envelope spectrogram closer together or further apart in time, according to the local ratio between base signal F0 and target F0. As a consequence, the spectral envelope on each frame needs to be modified accordingly: stretched if F0 was increased, or shrunk if F0 was decreased. All of these modifications, in time and frequency domain, are applied by using a cubic spline interpolator. As a result, we have a warped target spectral spectrogram, whose underlying f0 coincides to the one of the base signal. Figure 2.9 shows an example of this process.

2. *Spectral Envelope Modification*

   The filter is applied as described in Section 2.6.3 using the warped target spectral envelope spectrogram as target.

3. *Pitch Shifting*

   The pitch shifting is performed as described in Section 2.6.2. As mentioned, it stretches/shrinks the spectral envelope of the base signal according to the pitch modification. However this frequency domain scaling was "pre-corrected" given as a result the expected spectral structure in the synthesised signal.

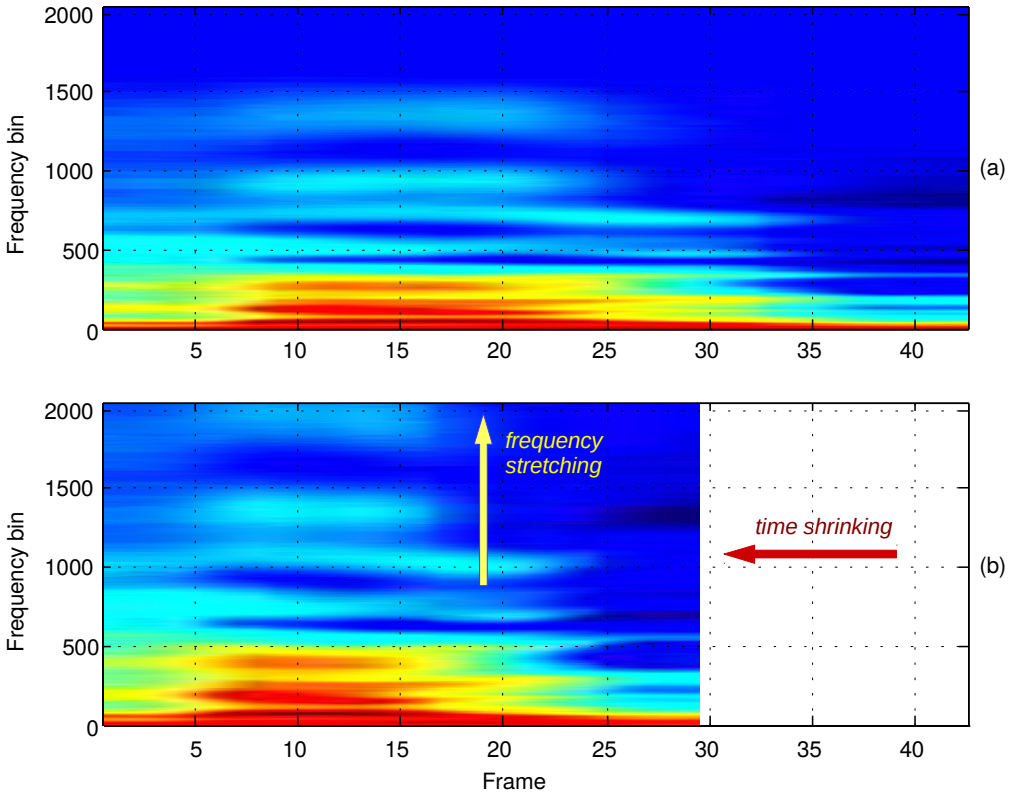

FIGURE 2.9: Example of time-frequency stretching of the target spectral envelope of one voiced segment. (a) Target spectral envelope spectrogram. (b) Warped target spectral envelope, stretched to match the base signal's F0. In this example, the target F0 is lower than the base signal F0, so the result is that the duration of the warped target spectral envelope sequence has become shorter, whilst it is stretched in frequency domain.

2.8 Final Proposed Method

Figure 2.10 summarises the final proposed waveform generator which will be tested by the experiments in Section 2.9. At its input it has the target F0 contour and the target spectral envelope spectrogram of a whole utterance. Firstly, the *spectral enhancement* is applied to the target spectral envelope spectrogram (See Section 2.7.5). Then, according to the voicing decision, implicit in the F0 contour, the data is divided into voiced and unvoiced segments, which are processed in completely separated blocks:

- *Voiced Segment Generation*

  Firstly, the target spectral envelope spectrogram is time-frequency stretched according to the target and base signal's F0 contours. Then, the spectral envelope modification of the base signal is performed. Later, the filtered base signal is pitch shifted so the resulting waveform has the target pitch. See Section 2.7.6 for details on all of these processes.

- *Unvoiced Segment Generation*

  The synthesis of unvoiced segments is performed as detailed in Section 2.7.2.

Finally, the voiced and unvoiced segments are concatenated in waveform domain to form the speech for the whole utterance.

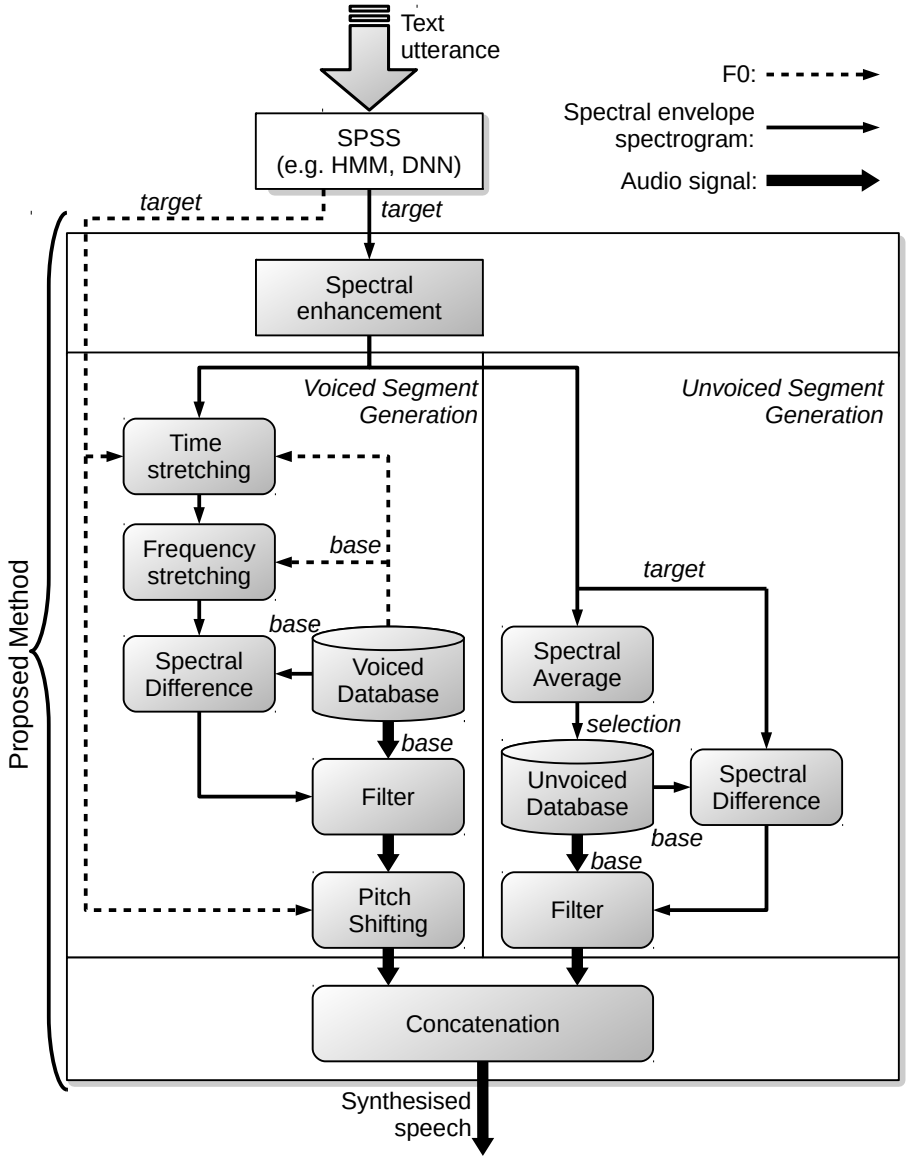

FIGURE 2.10: Block diagram of the complete proposed method including the unvoiced database, in the framework of SPSS.

2.9 Experiments

Even though objective evaluations can give a good approximation for error of predictions, they do not necessarily correlate with the perceived naturalness of synthetic speech. Accordingly, only subjective evaluations were carried out.

2.9.1 Evaluation

Two English DNN-based SPSS voices were built using the Merlin toolkit (Wu et al., 2016). The architecture of the network consisted on 4 feed-forward layers, of 1025 units each, plus an SLSTM layer in top of 512 units. The voices were one female "Laura" and one male "Nick". The female voice was built with 4500, 60 and 67 sentences for training, validation and testing, respectively. To build the male voice, 2400, 70 and 72 sentences for training, validation, and testing were used, respectively. Only 5 base signals were used as a total, all of them recorded by speakers other than the ones used to build the DNN voices. The duration of the base signals were: /f/=2.8 secs., /s/=4.4 secs., /ʃ/=2.6 secs., /a/ female=4.6 secs., /a/ male=6.0 secs. For each experiment only 4 base signals were used, one voiced, and 3 unvoiced.

Thirty native English-speaking university students evaluated the systems using a MUSHRA-like[3] listening test. The listening tests were carried out in sound-proof booths, each containing a desktop computer connected to an audio interface Focusrite iTrack Solo to feed the headphones Beyerdynamic DT 770, which would were worn by the listeners. Each of the subjects evaluated 30 MUSHRA screens (30 different sentences). Each sentence was randomly chosen from the test sets. Half of the sentences were the male voice and half the female voice for each participant. They were asked to assess the naturalness of six different stimuli randomly ordered in each screen (See Appendix A for full details on the instructions). The stimuli are described in Table 2.2.

Subjects were obliged to give at least a score of 100 to one stimulus per screen before proceeding to the next screen.

2.9.2 Results

The evaluation of one subject was dismissed due to inconsistent scores. For instance, natural speech was given a score below 30% several times. Holm-Bonferroni correction was applied because of the large number of systems to compare. To test statistical significance, the Wilcoxon Signed Rank test at $p<0.05$ was utilised.

[3]Code available at `http://dx.doi.org/10.7488/ds/1316`

Configuration	Description
Nat	Natural speech (the hidden reference).
STR	STRAIGHT (baseline)
SR_all	Signal Reshaping with "ideal" settings: matched-gender voiced base signal, linear-phase filtering, and Mel-scale spectral smoothing (all = all settings ideal)
SR_gen	as SR_all but base voiced signal is from the opposite gender to target (gen = mismatched gender)
SR_dp	as SR_all but filtering is not linear phase. MLSA filter (Imai et al., 1983) is used. (dp = distorted phase)
SR_ns	as SR_all but without Mel-warped spectral smoothing of base signal spectral envelopes (ns = no smoothing)

TABLE 2.2: Stimuli randomly ordered in each MUSHRA screen. Note that all stimuli were generated using the same system architecture (TTS), with the exception of Nat.

	system					
	Nat	STR	SR_all	SR_gen	SR_dp	SR_ns
female	99.9	38.3	42.9	42.6	43.7	41.9
male	99.7	50.5	48.5	48.6	51.6	45.9

TABLE 2.3: Average MUSHRA score per system in evaluation.

2.9.2.1 Female Voice

The results for the female voice are shown in Table 2.3 and Figures 2.11-2.12. All variants of the proposed method significantly outperformed the baseline, STRAIGHT in terms of absolute score. System SR_dp is significantly preferred in both rank and absolute score. SR_all and SR_dp work significantly better than SR_ns in terms of absolute score; SR_dp was significantly preferred over SR_ns in the rank score.

2.9.2.2 Male Voice

Table 2.3 and Figures 2.13-2.14 show the results for the male voice. SR_dp performed significantly better than the baseline, STRAIGHT, in terms of the rank analysis, although in absolute score there is no significant difference. SR_ns is significantly worse than all other systems under test.

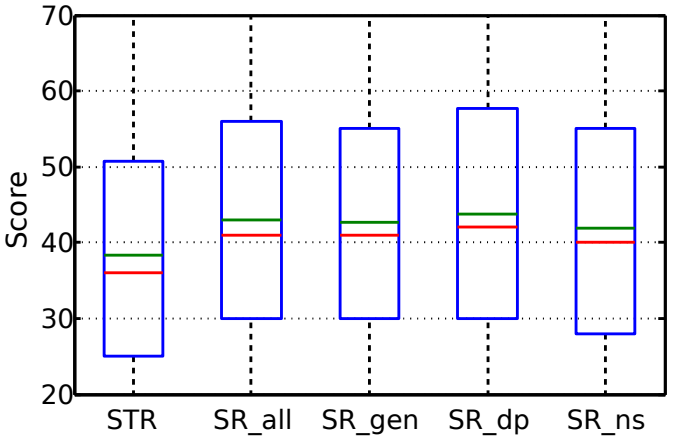

FIGURE 2.11: Results for the female voice, absolute scores. Natural speech is omitted (mean score is approx. 100) and the vertical scale is limited to 20–70, for clarity.

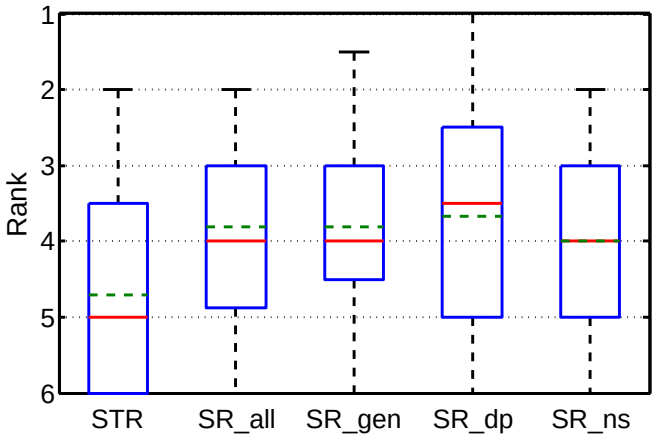

FIGURE 2.12: Results for the female voice, rank order. Derived from absolute scores within each MUSHRA screen. Natural speech is omitted (rank is approx. 1).

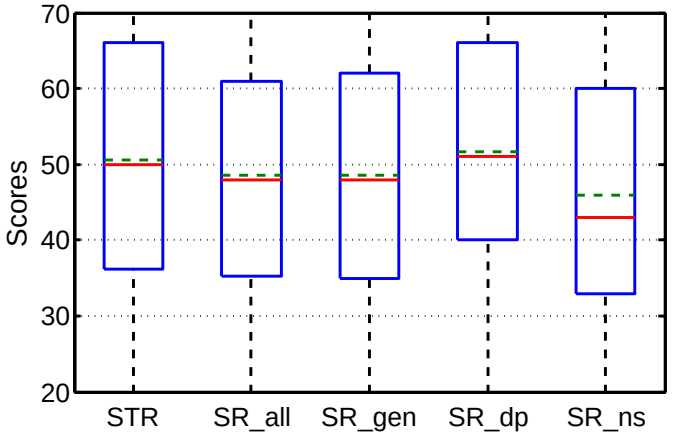

FIGURE 2.13: Results for the male voice, absolute scores. Natural speech is omitted (mean score is approx. 100) and the vertical scale is limited to 20–70, for clarity.

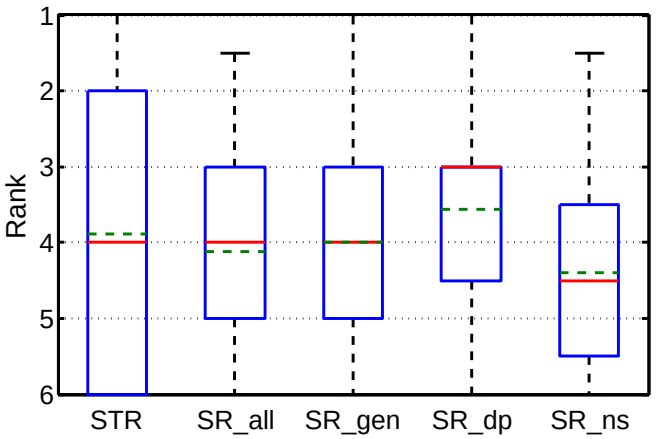

FIGURE 2.14: Results for the male voice, rank order. Derived from absolute scores within each MUSHRA screen. Natural speech is omitted (rank is approx. 1).

2.10 Conclusion

We have proposed a new paradigm for waveform generation which, unlike typical methods, does not intend to decompose waveforms. Instead it reshapes a base signal using filtering and pitch manipulation. Even though the proposed waveform generation uses spectral envelope as input paramater, which usually in our experiments was provided by the vocoder STRAIGHT, its sound quality highly differs to typical vocoders. This is mainly due to the characteristics retained in the base signal, which typical vocoders don't model:

- Natural aperiodicities from natural speech, and not emulated by using random noise or other techniques commonly applied by typical vocoders. We believe that it is critical especially during voicing, since the aperiodic energy should be to some degree correlated with pitch and synchronised with GCIs.
- Natural local fluctuations in pitch and natural inharmonicity are retained. Typical vocoders make simplistic modelling of these features by using either perfectly spaced pulse trains or sinusoids. However, these parameters show structured local variations in natural speech, which require complex modelling.
- In the proposed method, the spectral envelope modification and pitch shifting procedures are designed carefully to keep the named characteristics unmodified as much as possible in order to keep the naturalness in the base signal.

We built two SPSS voices in different configurations of our proposed method, and carried out subjective measurements to measure the performance of the proposed method and compare with the baseline system. The system SR_dp showed the best performance, overall. It is significantly better than the baseline, STRAIGHT, in both absolute score

and rank order for the case of the female data. For the male data, the system SR_dp only outperforms the baseline in rank order. Overall, these results show that the proposed method clearly tends to perform better than the baseline, STRAIGHT. The difference in the relative performance for the female and male voices could be due to:

- Decreasing, compared to increasing the F0 of the base signal creates worse artefacts. By increasing F0, natural aperiodicities present in the base signal are shifted towards higher frequencies, where aperiodicities are usually located, avoiding significant perceptible artefacts, but when decreasing F0, natural aperiodicities are moved towards lower frequencies, adding aperiodic components where they are not usually positioned.
- It is known that STRAIGHT is generally worse for female than male voices.

Notably, the distorted phase variant (SR_dp) using the MLSA filter outperformed the linear phase variant (SR_all). We hypothesise that this is due to the length of the filter tails usually produced by linear phase filters, especially the long pre-delay that is commonly perceived as unnatural.

Also, another interesting result is given by the SR_gen system, which performs highly competitively as shown by the scores for the male and female voices. This implies that the speaker chosen to provide the voiced base signal is not critical in the final speech quality. In fact, even if the voiced base signal comes from a speaker of the opposite gender, the speech quality is not remarkably degraded.

The proposed method does not require any parameter related to aperiodicities. When doing copy-synthesis, the synthesised speech has the characteristics of the speaker and the original utterance, producing an almost identical copy of the signal even though base signals are form other speakers. This suggests that perhaps it is not mandatory to explicitly manipulate aperiodicities. Hence, one advantage of the proposed waveform generator is that it needs only spectral envelope and F0 as acoustic parameters, which is fewer than commonly used by conventional vocoders. That means that the SPSS acoustic model has fewer acoustic features to predict from the input text features, making the inference task easier.

As mentioned, from the different filter implementations presented on this work for spectral envelope adaptation, the MLSA filter achieves the best sound quality. Nevertheless, it is worth emphasizing that artefacts are still present, which are perceived as “phasiness”or “chorus effect”. This is assumed to be an issue related to the phase distortion produced by the continuously changing filter.

Generally speaking, filtering adds weighted and delayed versions of the input signal to itself, which inevitably produces external, sometimes spurious components. An exception to this occurs when some signal components are cancelled out by the filter. To achieve this useful property, it is necessary to estimate the components of the signal that we wish to remove. However, filter estimation is a very difficult task that is still open to research.

Future work includes the application of the method to hybrid speech synthesis, join smoothing in concatenation-based systems.

Chapter 3

MagPhase Vocoder: Direct Modelling of Magnitude and Phase Spectra

3.1 Motivation

From the results of the previous chapter, we infer that:

- Removing unnecessary decomposition in speech modelling is beneficial to achieve higher quality.
- Even though natural speech was used for waveform generation, speech quality was not dramatically improved.
- Because the waveform generator relies on conventional speech features derived from STRAIGHT, its perceived sound quality resembles STRAIGHT.
- Conventional speech features are suboptimal.

All of these statements suggest that in order to improve speech quality in SPSS, it is necessary to propose not only a new waveform generator, but a whole vocoder comprising both feature extraction to extract optimal acoustic features, and a new waveform generator to achieve higher speech quality.

In this chapter we propose a method to model speech directly from its Discrete Fourier Transform (DFT). In this approach, we map the complex-valued Fourier transform to a set of four real-valued components that represent the magnitude and the phase spectra.

We do not perform any of the typical decompositions of the speech structure, such as source-filter separation, harmonics+noise, or any other derived method.

3.2 Goals and Challenges

The goals of the proposed method are to:

- Avoid estimation steps to the greatest extent possible.
- Extract features that are consistent so they can be used for statistical modelling (e.g., they can be safely averaged).
- Get rid of the "phasiness" and "buzziness" of typical vocoders.
- Use a conventional real-valued deep learning architecture as typically employed in SPSS.

There are several problems that need to be solved to achieve the proposed goals. A first obstacle is that neural networks typically used in SPSS only deal with real-valued data, whilst the FFT derived spectrum is complex-valued. An exploratory study on complex-valued neural networks for SPSS was presented by Hu et al. (2016), unfortunately not showing competitive results. An obvious option is the use of real and imaginary parts of the FFT spectrum as separated real-valued feature streams. Unfortunately, by doing so, phase values are dominated by spectral bins with higher magnitudes, introducing an undesired bias for statistical modelling. Also, this would make the phase representation inconsistent, e.g., the network may see different numerical values for the same phase information.

In addition, aperiodic components of the spectrum are hard to estimate and model due to their quasi-random nature; a statistical regressor would tend to over-smooth the spectra.

For phase modelling, the first difficulty comes from time location. Depending on which point in time the phase is measured, its value will differ substantially. Namely, it is necessary to "normalise" the delay over all the measured phase spectra to make the extracted phase values consistent across all the speech database. Group delay normalisation may be applied to achieve this, but algorithms to compute group delay are heuristics, iterative, unstable, or inaccurate, making the task error prone, e.g., Murthy and Gadde (2003). For this very reason, the use of unwrapped phase is pointless due to its inconsistency across the speech data, not to mention wrapped phase, which is known to be highly inconsistent. See the examples in Figures 3.1 and 3.2 for clarification. All of these issues are studied and explained in detail in Section 3.3.1.

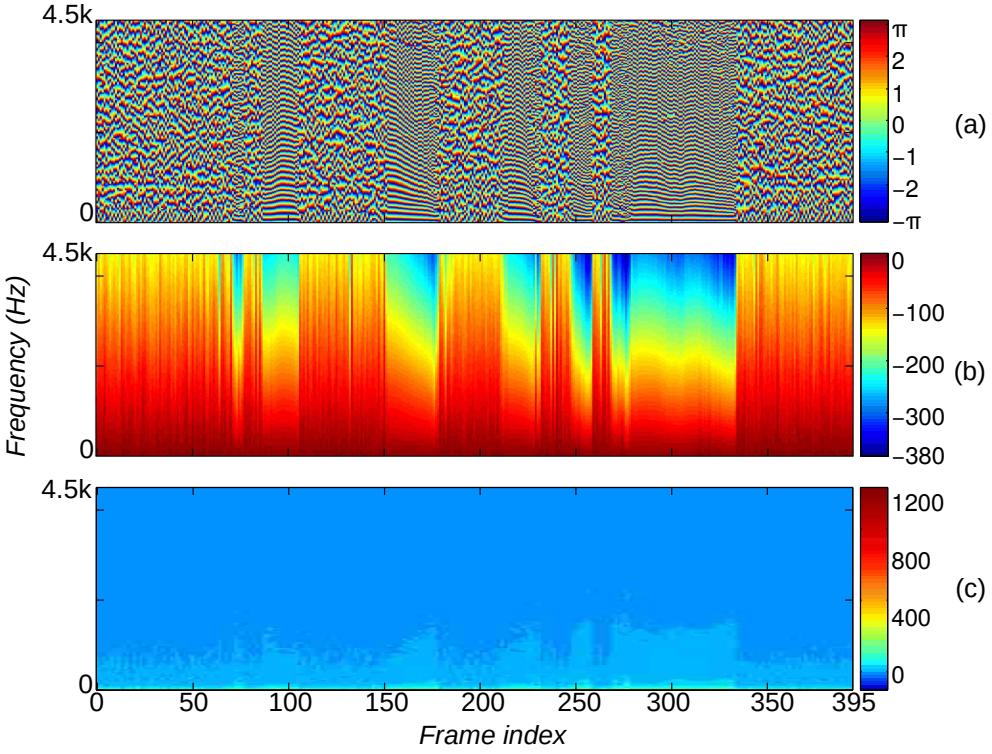

FIGURE 3.1: Examples of typical phase representations extracted from a utterance. The plots show the lack of recognisable patterns that may be successfully used in statistical modelling. (a) Wrapped phase. (b) Unwrapped phase. (c) Group delay.

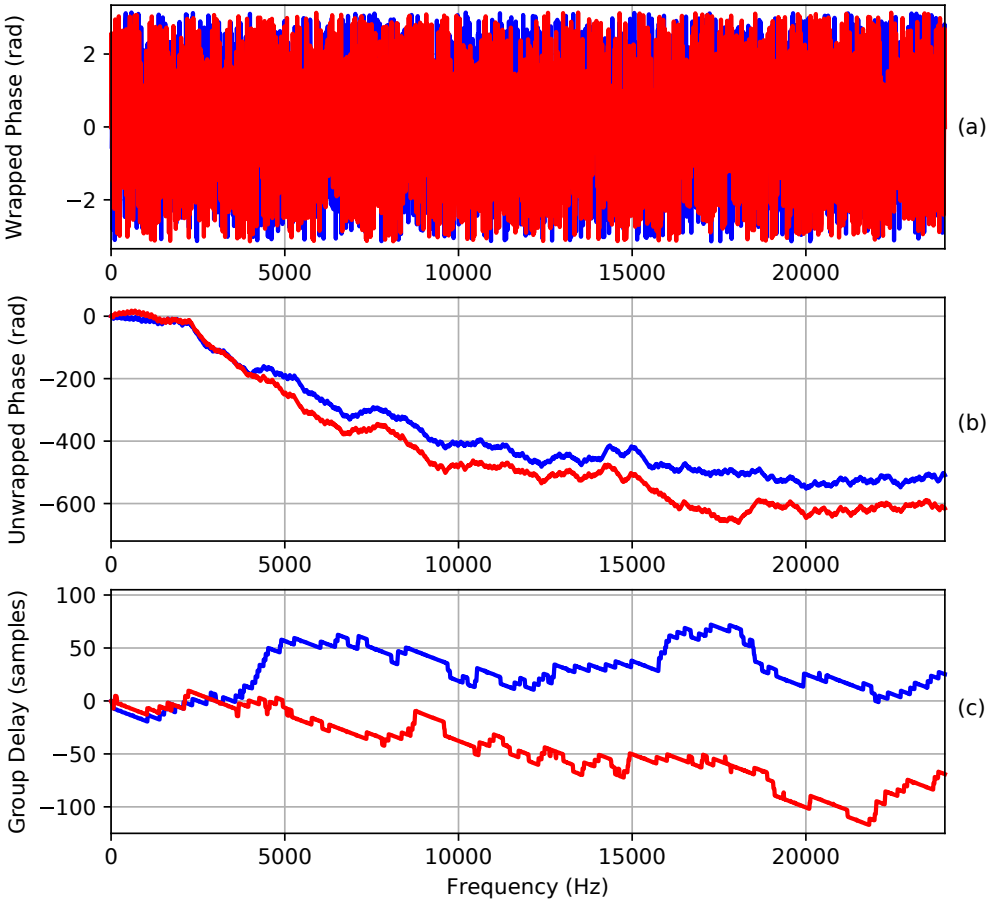

FIGURE 3.2: Spectral phase representations of two consecutive frames of a vowel. For a consistent representation, we expect to have similar spectral contours for consecutive frames. However, the plots show how inconsistent these representations can be by giving totally different curves for the analysed frames.(a) Wrapped phase. (b) Unwrapped phase. (c) Group delay.

3.3 Proposed Method

As we know, vocoders are basically made of two blocks, *Analysis* (feature extraction) and *Synthesis* (waveform generation). For our proposed method is the same.

3.3.1 Analysis

The analysis is carried out frame-by-frame, whereby four features are extracted for each analysis frame, pitch-synchronously. These are:

- **Fundamental Frequency (f_0):** A one dimensional feature that contains the fundamental frequency of the current analysis frame. Also, it indirectly represents the time lapse between two consecutive epoch locations (GCIs).
- **Magnitude Spectrum (M):** Directly obtained from the FFT coefficients extracted from the current analysis frame. Its dimensionality depends on the selected FFT length (dimension = 1 + FFT length / 2).
- **Normalised Real Spectrum (R)**: The normalised real part of the FFT coefficients. Its dimensionality depends on the selected FFT length (dimension = 1 + FFT length / 2).
- **Normalised Imaginary Spectrum (I)**: The normalised imaginary part of the FFT coefficients. Its dimensionality depends on the selected FFT length (dimension = 1 + FFTlength / 2).

The derivation of the introduced acoustic features is explained in the following subsections. A diagram of the whole analysis procedure is depicted in Figure 3.3.

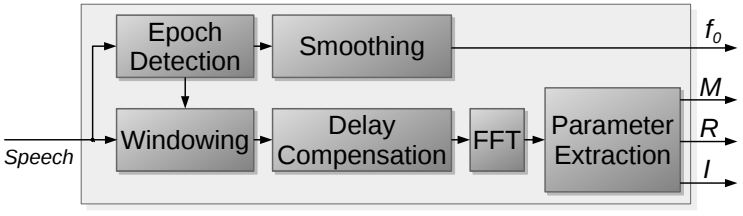

FIGURE 3.3: General diagram of the analysis process. Speech is parametrised into four acoustic feature streams: fundamental frequency (f_0), magnitude spectrum (M), normalised real spectrum (R), and normalised imaginary spectrum (I).

3.3.1.1 Fundamental Frequency

Analysis frames are centred on the epoch locations (GCIs). For epoch location extraction, we use the software REAPER[1], which has shown high consistency, although simpler methods may be also applied. Basically, it gives the locations of GCIs for voiced speech and equally spaced epochs for unvoiced speech. The spacing (or frame shift) in unvoiced speech is constant and set by the user.

From the extracted epoch locations for voiced speech, the frame shift can be obtained by subtracting two consecutive epoch locations:

$$s_{[i]} = e_{[i]} - e_{[i-1]} \tag{3.1}$$

Where i is the frame index, $e_{[i]}$ is the epoch location of the current analysis frame, $e_{[i-1]}$ is the epoch location of the previous analysis frame, and $s_{[i]}$ is the frame shift for the current analysis frame. Also, REAPER gives the voicing decision for each frame, returning $v = 1$ for voiced and $v = 0$ for unvoiced. Then, the fundamental frequency (f_0) can be derived from the frame shift and the voicing decision as:

$$f_{0[i]} = \begin{cases} \frac{1}{s_{[i]}}, & \text{if } v = 1 \\ 0, & \text{if } v = 0 \end{cases} \tag{3.2}$$

Henceforth, the frame index i will be omitted for simplicity when being irrelevant, i.e., for intra frame operations.

3.3.1.2 Phase Spectrum

As said in Section 3.1, we want to encode the FFT coefficients in such a way that they are consistent, to build statistical models representing speech. One of the most challenging characteristics to model is the phase spectrum, which is even very difficult to analyse.

Usually, the wrapped phase spectrum Φ_w is computed from the complex FFT coefficients X, using the $atan2$ function:

$$\Phi_w = \text{atan2}\left(\text{Re}\{\text{X}\}, \text{Im}\{\text{X}\}\right) \tag{3.3}$$

Even though it is a fairly simple and cheap operation, this phase representation is unsuitable for statistical modelling. Let us take an example where the wrapped phase

[1]Freely available at `https://github.com/google/REAPER` (December, 2016)

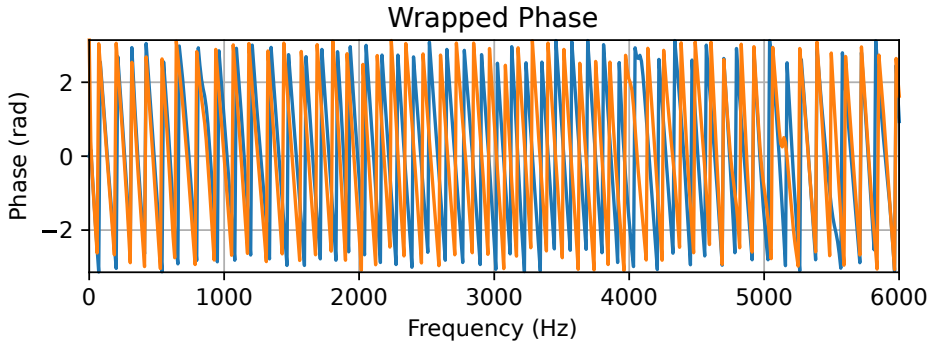

FIGURE 3.4: Wrapped phase spectrums of two consecutive voiced frames.

spectrum of two consecutive frames (voiced and with stable characteristics) is extracted (Figure 3.4).

As this phase representation is expected to be used for statistical modelling, the two curves in Figure 3.4 should look similar, since they represent two similar chunks of the same phoneme. However, as shown in Figure 3.4, they differ greatly. Also, we expect that they may show some recognisable pattern, but instead they exhibit many "jumps" along the frequency axis, due to the wrapping between $-\pi$ and π.

Figure 3.5 shows a zoom into the range of 3kHz and 3.7kHz. In here, the large difference between the two wrapped phase spectrums is very clear. Actually, one can observe that there are several instances where they present completely opposite values in the same frequency location, e.g., at 3.6kHz.

As a consequence, the "jumps" appear to occur randomly, which makes this phase representation unsuitable for statistical modelling. The following subsections will describe the steps required to extract more suitable features that describe phase spectrum of speech.

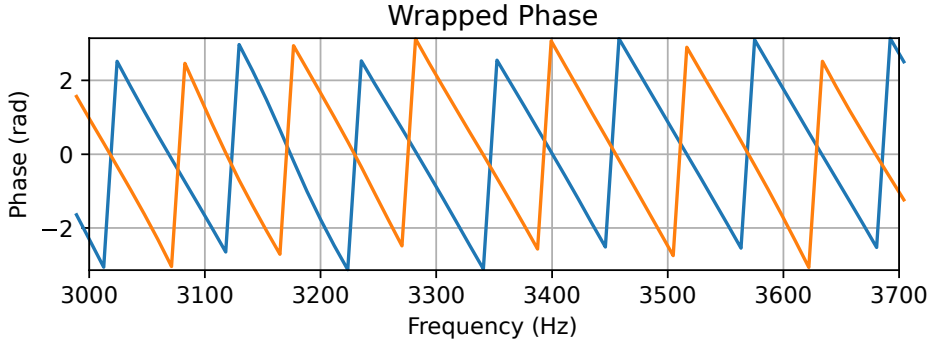

FIGURE 3.5: Close-up of wrapped phase spectrums of two consecutive voiced frames.

3.3.1.3 Windowing

The proposed vocoder performs frame-based analysis. Each analysis frame spans two whole pitch periods, thus including three epochs.

We want to have every frame representing the acoustic characteristics of each epoch and their surroundings. Thus, a window needs to be applied to maximise the amplitude at the target epoch location, while decreasing the amplitude towards the neighbouring epochs. Since it is not guaranteed that the length of two consecutive pitch periods is the same, we need to use a non-symmetrical window. We chose the non-symmetrical Hann window, whose top (maximum height) is placed exactly at the central epoch of the current analysis frame, and its borders are placed at the previous and next epoch locations (frame boundaries), as shown in Figure 3.6. The Hann window ensures perfect amplitude flatness with 50% overlap which results in perfect reconstruction for our system (Heinzel et al., 2002).

By doing so, harmonic structures in the spectrum are almost not present, since the analysis frame is not long enough to include highly periodic events produced by the glottal folds. As a result, its magnitude spectrum remains almost entirely pitch independent, with the exception of the spectral smearing produced by the window and small harmonic structures that are considered negligible.

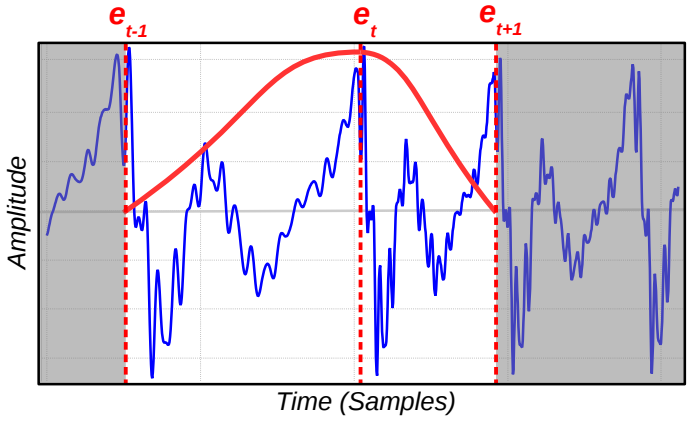

FIGURE 3.6: Non-symmetric window placed between epochs e_{t-1} and e_{t+1}. The maximum of the window is placed right at the central epoch e_t.

3.3.1.4 Delay Compensation

Phase modelling must have inter-frame consistency, such that the phase extracted from different frames across the whole database can be compared, averaged or weighted. In this regard, the time at which the measurement is taken is critical. Within the frame, this point in time is given by the delay of the signal within the analysis frame. In other

words, the delay of the signal relative to the beginning of the analysis frame needs to be normalised. One obvious approach is by computing the group delay and then fixing it to a certain value. However, as stated in Section 3.2, that would be highly error prone due to the heuristics involved.

Our delay compensation process can be thought of as a simple and robust method for group delay normalisation. Firstly, each analysis frame is zero padded to a fixed FFT length (e.g., 4096), which should be at least as long as the longest frame extracted. Secondly, assuming that epochs are located close to points of maximum absolute amplitude within a frame, the signal can be shifted backwards such that its central epoch is placed at the first sample position within the analysis frame (See Figure 3.7). In turn, the portion of the signal that was originally preceding the central epoch location, is folded towards the end of the analysis frame. It behaves like a time-aliasing device, which uses the analysis frame as a circular buffer. This simple operation has several benefits:

- Ensures phase consistency between frames.
- Minimises phase wrapping.
- Maximises the smoothness of phase spectra, which helps for subsequent modelling, dimensionality reduction, or frequency warping, if needed.

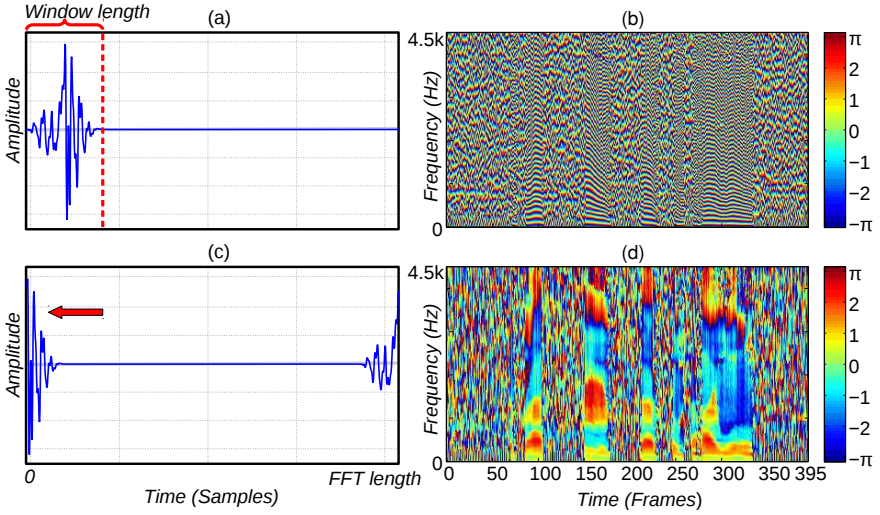

FIGURE 3.7: Delay compensation example and its effects on the phase spectra. Clear phase patterns emerge from spectra after delay compensation. (a) Frame before delay compensation. (b) Wrapped-phase spectra of a utterance before delay compensation. (c) Frame after delay compensation. (d) Wrapped-phase spectra of the same utterance after delay compensation.

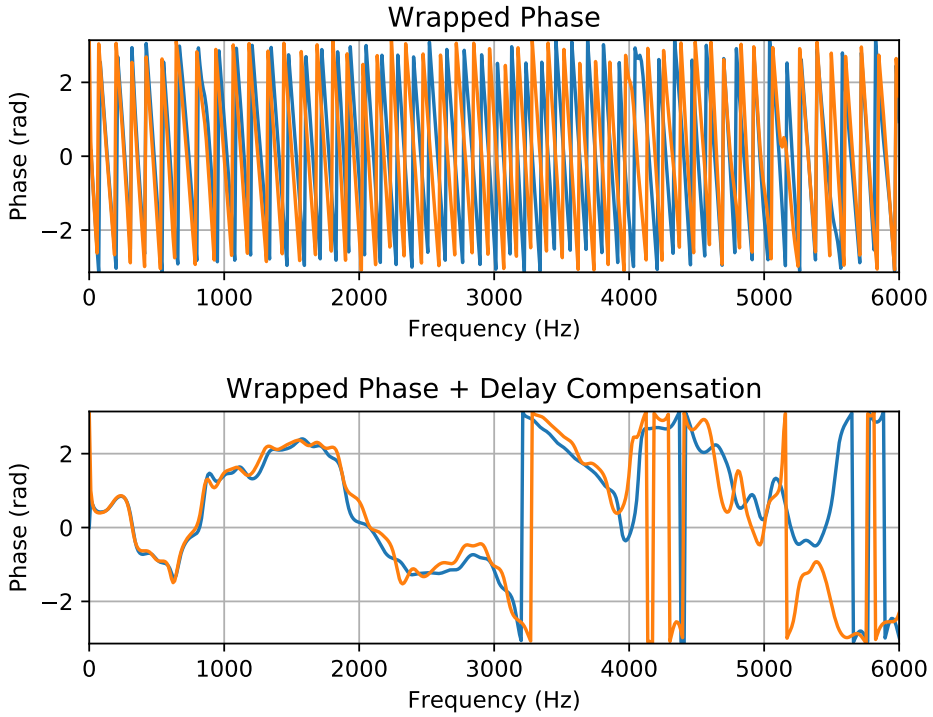

FIGURE 3.8: Phase spectrums of two consecutive frames of voiced speech. Top: Before delay compensation. Bottom: After delay compensation.

These benefits can be seen in Figure 3.8, where the effect of the delay compensation process is obvious: the two phase spectra presented in Figure 3.4 now look smoother, showing possible patterns and also depicting a similar shape, which is what we wanted. Nevertheless, there are still some "jumps" present from around 3.2kHz onwards.

To the best of our knowledge, our proposed delay compensation is a novel approach, which in spite of being quite simple, has never been developed to keep phase consistency across frames.

3.3.1.5 Phase Re-wrapping

Once the delay is compensated within each frame, the complex spectrum (FFT coefficients) X is computed by using an FFT. Then, the wrapped phase spectrum Φ_w is calculated using Equation 3.3.

As mentioned in Section 3.3.1.4, there are still some problems with the resulting phase representation. Hence, it is necessary to carry out an extra process to fix the issues that the delay compensation was not able to correct. Basically, the problem can be illustrated by the difference between the two phase representations in the lower pane in Figure 3.8. For instance, the first "jump" occurs at around 3.2kHz, which makes the two curves take completely opposite values; namely the blue curve close to π and the

orange curve gets a value close to $-\pi$. We expect the two curves should look similar, as stated in Section 3.3.1.2, thus in the example, the ideal phase representation should make the curves have a similar value if they originally pointed close to π and $-\pi$, respectively.

The *cosine* function is used to "re-wrap" the wrapped phase spectrum Φ_{w}. Thus for instance, the phase values close to π and $-\pi$ are mapped close to the same value, -1. However, this presents some unwanted effects, e.g., phase values close to $\frac{\pi}{2}$ and $-\frac{\pi}{2}$ converge to the same value, zero, producing some ambiguity. Hence, for disambiguation, the wrapped phase is also "re-wrapped" by a *sine* function. By doing so, two phase descriptors R and I are built per frame (frame index i is committed for simplicity):

$$\begin{aligned} \mathrm{R} &= \cos(\Phi_{\mathrm{w}}) \\ \mathrm{I} &= \sin(\Phi_{\mathrm{w}}) \end{aligned} \tag{3.4}$$

Interestingly, these are nothing but the normalised real and imaginary parts of the FFT complex spectrum X:

$$\mathrm{R} = \frac{\mathrm{Re}\{\mathrm{X}\}}{\mathrm{M}} \quad , \quad \mathrm{I} = \frac{\mathrm{Im}\{\mathrm{X}\}}{\mathrm{M}} \tag{3.5}$$

Where R and I are the normalised real and imaginary spectra, respectively. X is the FFT complex spectrum, and M is the magnitude spectrum obtained from Equation 3.6.

To the best of our knowledge, our phase re-wrapping is a novel approach not based on any previous developed method for acoustic feature representation. Figure 3.9 shows an example of the re-wrapping procedure where it can be seen how the phase spectrums get even smoother, closer, and consistent, specially over 3kHz.

3.3.1.6 Magnitude Spectrum

The Magnitude spectrum is computed as usual from FFT coefficients:

$$\mathrm{M} = \sqrt{\mathrm{Re}\{\mathrm{X}\}^2 + \mathrm{Im}\{\mathrm{X}\}^2} \tag{3.6}$$

3.3.1.7 Lossless Features

As a result, four feature streams are extracted from each analysis frame:

$$\mathrm{F} = \{f_0, \mathrm{M}, \mathrm{R}, \mathrm{I}\} \tag{3.7}$$

It is worth emphasising that these features are full resolution, and support perfect reconstruction of speech, if used for "copy-synthesis".

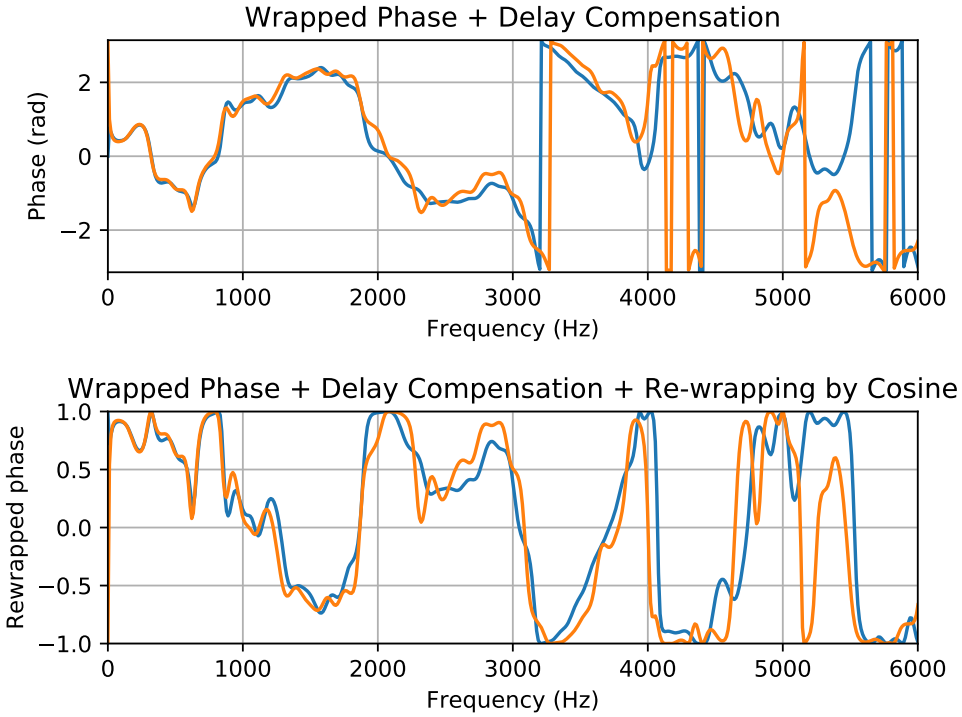

FIGURE 3.9: Re-wrapped phase spectrums (by cosine) of two consecutive frames of voiced speech. Top: Before phase re-wrapping. Bottom: After phase re-wrapping.

If we use the proposed features for statistical modelling (i.e., SPSS), the effect of the magnitude on phase calculations will be removed. This is because of the normalisation of each complex number of the spectrum by its magnitude. This is of great importance especially in least-squares trained models, in which summing and averaging are the key operations (e.g., DNN). Hence, during prediction, phase can be correctly approximated in the output of the model. As a final step, the resulting complex number (actually two real valued streams) is scaled to the correct magnitude, using the M parameter that has been modelled separately.

Even though the lossless features constitute a full representation of speech signals and can be used for perfect reconstruction, they need to be adjusted to work optimally with a standard SPSS architecture. This feature engineering will be explained on Section 3.3.1.8.

3.3.1.8 Feature Engineering for Statistical Parametric Speech Synthesis

In order to work optimally with a standard SPSS architecture, the feature streams need to be:

- ***Low-dimensional:*** The full resolution features streams M, R and I are of length 1 + FFT length / 2, which translates into a very long acoustic feature set. That is inconvenient for the acoustical model. Some techniques for compression were

attempted, always aiming at obtaining a low-dimensional representation in frequency domain.

Mel-frequency warping lowers the resolution of data in high frequencies, progressively, which allows for decreasing the number of features without affecting the speech quality too much (perceptually). It is well known (Volkmann et al., 1937) that humans are less able to distinguish details in higher frequencies making this type of compression suitable for frequency domain speech features.

The first Mel-frequency warping method attempted was based on the Mel-Cepstral Analysis (MCEP) (Tokuda et al., 1995a), which basically warps the frequency axis to the Mel-scale and transforms the data from frequency domain into cepstral domain. All of this is done just in one step, basically. By doing so, and thanks to the Mel-warping, it is possible to lower the dimensionality of the feature stream. Once the data is in ceptral domain, it is again transformed back into frequency domain by means of the Fourier transform (FFT). As a last step, the exp function is used to remove the log applied during the MCEP analysis. As a result, a lower dimensional Mel-scaled spectral representation is obtained in the frequency domain.

Also, a filter bank approach was attempted for the Mel-frequency warping. It essentially comprised several triangular frequency bands in Mel-frequency domain exhibiting a constant band width. The filter bank is represented as a matrix with shape [*number of filters* $\times$ *number of frequency bins*]. The Mel filter bank frequency bins are warped to linear-frequency domain by the inverted Mel-scale. Then, the filter bands are resampled at the locations of the linear-frequency FFT bins. As a result, a warped filter bank is obtained in linear-frequency domain, whose band-widths increase progressively towards high frequencies. Then, the filter bank is applied by multiplying each band with the frequency domain spectral features (all in linear-frequency domain).

According to informal experiments, the first approach, based on the Mel-Cepstral Analysis (MCEP), achieved slightly better results, therefore, it was the approach used to test the system in formal experiments.

- ***Normally distributed:*** We intend to use a typical neural network-based SPSS system, which uses mean-squared error as a loss function. Hence, it assumes some properties of the data to be modelled: that it should be normally distributed, unimodal and symmetric.

- ***Predictable:*** Sometimes, data is just too hard to predict for a mean-squared error based regressor, i.e., when data is generated by a purely stochastic process. Accordingly, some data was removed in certain cases where it was found detrimental for inference.

Thus, the specific modifications applied to the feature streams were:

1. **Magnitude Spectrum (M):** The high resolution magnitude spectrum M is Mel-warped, thus compressed by using the method based on Mel-Cepstral Analysis (MCEP). Additionally, since M is defined in the codomain $[0, +\inf[$, its distribution is highly skewed. Therefore, M is compressed by the log function to approximate to a normal distribution:

$$\mathrm{M}_c = \log\left(\mathrm{W}(\mathrm{M})\right) \tag{3.8}$$

   Where M_c is the new compressed magnitude spectrum feature, and W is the Mel-warping function.

2. **Phase Derived Spectral Features (R, I):**

   The high resolution phase features R and I are also compressed to achieve lower dimensionality using the same Mel-warping process as for the feature M. However, due to the randomness and unpredictability of the phase features in high frequencies, the last few Mel-frequency bands are removed. Also due to the same reason, the phase features are removed for unvoiced frames; in practice, they are zeroised, when $f_0 = 0$. As a result, the new compressed phase features R_c and I_c are defined as:

$$\mathrm{R}_c = \begin{cases} \mathrm{W}(\mathrm{R}), & \text{if } f_0 > 0 \\ 0, & \text{if } f_0 = 0 \end{cases} \tag{3.9}$$

$$\mathrm{I}_c = \begin{cases} \mathrm{W}(\mathrm{I}), & \text{if } f_0 > 0 \\ 0, & \text{if } f_0 = 0 \end{cases} \tag{3.10}$$

3. **Fundamental Frequency (f_0):** The raw fundamental frequency f_0 is also optimised for acoustic modelling. Firstly, smoothing is applied, because it was observed that the extracted frame shifts changed faster than typical f_0 curves. Thus, to avoid the risk of producing some jitter effect in the fundamental frequency, a median filter of 3 frames length is employed. Secondly, the log function is applied over f_0, as typically done in SPSS systems.

   Thirdly, the "zeros" in the f_0 array during unvoiced speech are replaced by linearly-interpolated values between the last voiced f_0 and the next voiced f_0 value, as

described in Yu and Young (2011). As a result, a smooth f_0 trajectory is produced exhibiting a continuous contour, as opposite to the discontinuous original behaviour. However, the voicing decision information is lost, thus an auxiliary acoustic feature is added: the voicing decision v, which is defined by:

$$v = \begin{cases} 1, & \text{if } f_0 > 0 \\ 0, & \text{if } f_0 = 0 \end{cases} \tag{3.11}$$

Hence, the new compressed fundamental frequency feature f_{0c} is defined by:

$$f_{0c} = \begin{cases} \log(\text{medf}(f_0)), & \text{if } v = 1 \\ \text{interp}\left(\log(\text{medf}(f_0))\right), & \text{if } v = 0 \end{cases} \tag{3.12}$$

Where, *interp* is the linear-interpolation function for unvoiced frames and medf is the median filter of 3 taps.

As a result, the new compressed acoustic feature set F_c, ready to be used for acoustic modelling, comprises:

$$\mathrm{F}_c = \{f_{0c}, \mathrm{M}_c, \mathrm{R}_c, \mathrm{I}_c\} \tag{3.13}$$

3.3.2 Synthesis From Natural Speech Features (Lossless Copy Synthesis)

Once the feature set F presented in Equation 3.7 is extracted from natural speech, it is possible to reconstruct the speech signal without loss. The waveform generation is straightforward, and is carried out by the following steps:

1. The complex spectrum X for the frame i is given by:

$$\mathrm{X}_i = \mathrm{M}_i \cdot (\mathrm{R}_i + \mathrm{I}_i j) \tag{3.14}$$

   Where M is the magnitude spectrum, R the normalised real part of the spectrum, I is the normalised imaginary part of the spectrum, and j is the imaginary unit.

2. Then, the time-domain signal x for the frame i is obtained by:

$$x_i = \text{IFFT}(\mathrm{X}_i) \tag{3.15}$$

3. The time domain signal in each frame x_i was delay compensated during feature extraction. Now, the delay compensation needs to be removed. This is done by

shifting the signal forward by half of the frame size (FFTlength). It works as a circular buffer (See Section 3.3.1.4).

4. The location of epochs (GCIs) is computed by:

$$e_{[i]} = \begin{cases} e_{[i-1]} + \frac{1}{f_{0[i]}}, & \text{if } f_{0[i]} > 0 \\ e_{[i-1]} + c, & \text{if } f_{0[i]} = 0 \end{cases} \tag{3.16}$$

   Where $e_{[i]}$ is the epoch location at frame index i, f_0 is the fundamental frequency, and c is a constant frame shift for unvoiced segments defined by the user during feature extraction (e.g., 5ms).

5. Finally, the synthesised waveform w (time domain series) is generated by applying pitch synchronous overlap-add (PSOLA) over the frames $\{x_i\}$ centred at epoch locations e:

$$\mathrm{w} = \underset{e}{\mathrm{PSOLA}}\{x_i\} \tag{3.17}$$

As a result, the synthesised signal w is identical to the original speech signal.

3.3.3 Synthesis From Statistical Inference

Even though lossless features achieve perfect reconstruction, the objective of this work is a vocoder optimised for SPSS. Consequently, a synthesis procedure needs to be defined, which is able to synthesise from the compressed feature set F_c (Equation 3.13).

The synthesis process is made up of three main blocks. The *periodic spectrum generation* block produces complex spectra that represent the periodic excitation of speech production. It entirely relies on the input features (e.g., the ones predicted by a DNN) and the assumption that above a maximum voiced frequency (MVF), speech is only composed by aperiodic components. Hence, this stage produces complex spectra only for voiced segments, at frequencies lower than the MVF.

The *aperiodic spectrum generation* block uses the magnitude spectra and f_0 features, plus the phase extracted from random noise to produce complex spectra for unvoiced segments and the higher frequencies in voiced segments.

The final step, *waveform generation*, takes the complex spectra at the input and generates the waveform in the time domain.

The whole synthesis process is illustrated in Figure 3.10, and a detailed description of these stages is presented in the following subsections. For the sake of generality, it

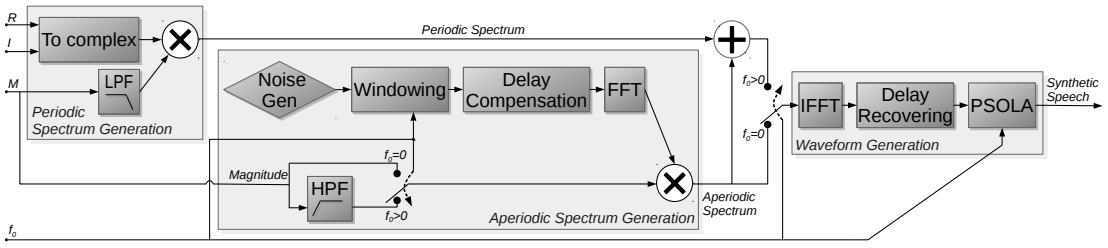

FIGURE 3.10: General diagram of the synthesis process. At the input, there are four feature streams: fundamental frequency (f_0), magnitude spectrum (M), normalised real spectrum (R), and normalised imaginary spectrum (I). At the output, the generated synthesised speech signal.

is assumed that the input features are generated by a statistical model (e.g., DNN), which we name F'_c, where the apostrophe denotes "predicted", rather than extracted from natural speech.

3.3.3.1 Feature Decoding

The inferred features F'_c are decoded using the inverse of the mechanisms described in Section 3.3.1.8. The features are:

- Unwarped from Mel-scale to linear-frequency by using the method presented in Valentini-Botinhao et al. (2015), where appropriate.
- Uncompressed by the exp function when the feature is log-based.
- f_{0c} is obtained by using the voicing decision v to set it to zero in unvoiced speech.

As a result, we have the inferred high resolution feature set F':

$$\mathrm{F}' = \{f'_0, \mathrm{M}', \mathrm{R}', \mathrm{I}'\} \tag{3.18}$$

3.3.3.2 Periodic Spectrum Generation

This process starts by taking the normalised real and imaginary spectra R' and I' features to produce the complex phase spectrum Φ'_{per}:

$$\Phi'_{per} = \frac{(\mathrm{R}' + \mathrm{I}'j)}{\sqrt{\mathrm{R}'^2 + \mathrm{I}'^2}} \tag{3.19}$$

Where j is the imaginary unit. The use of the normalisation term in the denominator is required due to prediction errors: the magnitude of the predicted complex $\mathrm{R}' + \mathrm{I}'j$

may not be 1. If the parameters R' and I' were extracted from natural speech, it would be guaranteed that its magnitude equals to 1.

The predicted magnitude spectrum M' is low-pass filtered at the maximum voiced frequency MVF. The filter is applied as a ramp in frequency domain. To achieve perfect spectral reconstruction, a half Hann window is selected as the ramp curve. Then, the low-pass filtered magnitude spectrum is multiplied by the complex phase Φ_{per}, resulting in the complex spectrum of the periodic components in speech S'_{per}:

$$\mathrm{S}'_{per} = \mathrm{M}' \cdot \Phi'_{per} \tag{3.20}$$

3.3.3.3 Aperiodic Spectrum Generation

Within this SPSS framework, the phase of aperiodic components cannot be recovered from the normalised real R' and imaginary spectra I' features, at least within the mean-squared error-based regressor. That is because the phase features behave stochastically, and the regressor predicts the expected value of a unimodal distribution. In practice, we would have values close to zero at the output, constantly. Hence, the aperiodic phase is derived from artificially generated random noise, as an approximation to its natural turbulent behaviour. The noise is zero mean and uniformly distributed. Its dispersion is irrelevant at this stage, since its energy will be normalised later.

The noise source signal has to span the whole synthetic speech utterance, since it is present in both voiced and unvoiced segments. Once generated, it is framed as done during analysis (Section 3.3.1.3). In the case of unvoiced speech, frames are equally spaced according to a fixed frame rate defined by the user (e.g., 5ms/sec), whilst epoch locations in voiced segments depend on the fundamental frequency (See Equation 3.16).

For the generated noise, windowing is applied differently according to the voicing decision. For unvoiced speech, the frames are windowed using a Hann window, i.e., as done for analysis. However, during voiced speech it was observed that the time-domain amplitude envelope of aperiodic components is pitch synchronous, and most of its energy is concentrated around the epoch locations. To emulate this behaviour, a narrower window is applied to the frames during voiced speech:

$$w_{[l]} = \left(\mathrm{Bartlett}_{[l]}\right)^{\lambda} \qquad \text{, with } \lambda > 1, \text{ and } l = 0, ..., L-1 \tag{3.21}$$

Where Bartlett is the "Bartlett" window function, l is the time index within the frame, L is the length of the window, and λ is a "sharpening" parameter that controls the degree of sharpness of the window. As for voiced speech during analysis (Section 3.3.1.3),

this window is applied in a non-symmetric fashion due to the variable frame shift during voiced speech. As a consequence, the amplitude envelope of the noise is shaped accordingly during reconstruction.

Then, the frames are "delay-compensated" using the same method applied in analysis (See Section 3.3.1.4). This is necessary because the frames containing noise have to sync and match the time-domain characteristics of the periodic spectrum (during voicing). Thus, the complex noise spectrum S_{noise} is computed by an FFT over each noise frame.

A spectral shape vector M'_{shape} is defined for each frame to modify the magnitude of the noise complex spectrum S_{noise}. M'_{shape} corresponds to the predicted magnitude spectrum M' normalised by the average root mean squared (RMS) energy of the magnitude spectra of noise. Then, in voiced speech, the resulting magnitude spectrum is high-pass filtered (HPF) at a cut-off frequency coinciding with the predefined MVF. This high-pass filter is complementary to the one used in the *periodic signal generation* stage, and is applied in the same form:

$$\mathrm{M}'_{shape} = \begin{cases} \mathrm{HPF}\left(\frac{\mathrm{M}'}{RMS}\right), & \text{if } f_0 > 0 \\ \frac{\mathrm{M}'}{RMS}, & \text{if } f_0 = 0 \end{cases} \tag{3.22}$$

Finally, the complex noise spectrum S_{noise} and the magnitude spectral shape spectral M'_{shape} are multiplied to produce the complex spectrum of aperiodic components S_{aper}:

$$\mathrm{S}_{aper} = \mathrm{M}'_{shape} \cdot \mathrm{S}_{noise} \tag{3.23}$$

3.3.3.4 Waveform Generation

At the input of the waveform generation stage, we have either the aperiodic complex spectra for unvoiced speech or a mixture of periodic and aperiodic spectra for the voiced segments:

$$\mathrm{S}' = \begin{cases} \mathrm{S}_{per} + \mathrm{S}_{aper}, & \text{if } f_0 > 0 \\ \mathrm{S}_{aper}, & \text{if } f_0 = 0 \end{cases} \tag{3.24}$$

The complex spectra S' are first transformed into time domain by IFFT. Then, for each frame, the waveform is shifted forward by a half of the FFT length, to revert the time aliasing produced by the *delay compensation* during analysis. As a result, the central epoch of the waveform is placed right in the centre of the frame, and its aliased part recovers its original location preceding the central epoch. Finally, the synthesised waveform is generated by using Pitch Synchronous Overlap and Add (PSOLA) following

the epochs derived from the predicted f_0. This process is analogous to the one described from step 2 to step 5 in Section 3.3.2.

3.3.4 Context frame features

Originally, the TTS system[2] that we use works with 9 frame positional features as described in Wu et al. (2016), of which 5 are state features (aligned with a 5-states HMM). Because in the proposed method the frame rate is variable, the mapping between frames and states is not surjective. Thus, the original 5 state features as frame positional indicators cannot be used due to their sequential nature. Instead, a numerical feature, the *normalised frame-state position* (p_f), composed by two sub-features is defined:

$$p_f = s_f + 0.1 \times r_f \tag{3.25}$$

Where s_f is an integer in the range $[0, 4]$ that indicates the state where the frame f belongs to. r_f is a real number in the range $[0, 1)$ that gives the relative frame position within the state. As a result, the proposed method works with 4 less positional features than the standard recipe used by the baseline.

[2]The Merlin toolkit (`https://github.com/CSTR-Edinburgh/merlin`)

3.4 Experiments

Two text-to-speech (TTS) voices running at 48kHz sample rate were built using the Merlin toolkit (Wu et al., 2016). A male voice, "Nick", was produced using 2400, 70 and 72 sentences for training, validation, and testing, respectively. Also, a female voice, "Laura", was produced with 4500, 60, and 67 sentences, respectively. The system was of type statistical parametric speech synthesis (SPSS) built with 4 feed-forward layers plus a simplified long-short term memory (SLSTM) layer on top (Wu and King, 2016). Each feed-forward layer contained 1024 units, while the last layer, the recurrent SLSTM, comprised 512 units.

The baseline system is made up by the same network architecture, but using the vocoder STRAIGHT (Kawahara et al., 1999a,b) for feature extraction and synthesis. It extracted high resolution acoustic features: spectral envelope, aperiodicities, and fundamental frequency, per each frame. The extraction ran at a 5ms constant frame rate. Then, the extracted acoustic features were compressed to 60 Mel-cepstral coefficients (MCEPs), 25 band aperiodicities (BAPs), and one log fundamental frequency (lf_0). This configuration has been extensively tested and is part of the standard recipes included in the Merlin toolkit.

The proposed vocoder supports several selectable parameters, which for these experiments were adjusted to:

- FFT-length = 4096
- Aperiodic voiced window sharpening factor $\lambda = 2.5$
- Maximum voiced frequency, MVF=4.5kHz
- Mel-warping factor $\alpha = 0.77$
- Log-magnitude spectrum size (M_c) = 60
- Normalised real spectrum size (R_c) = 45
- Normalised imaginary spectrum size (I_c) = 45

The fundamental frequency features lf_0 (baseline) and f_{0c} (proposed method) are equivalent in terms of meaning and characteristics, and need to be supported by the voicing decision v as an auxiliary acoustic feature (see Section 3.3.1.1).

3.4.1 Evaluation

A subjective evaluation was conducted to measure the performance in terms of naturalness, which allowed for comparing different configurations of the proposed method, and with a state-of-the-art system.

Thirty native English speakers (college students) were recruited to take a MUSHRA-like[3] test. The listeners evaluated the stimuli in sound-proof booths, where they wore headphones Beyerdynamic DT 770, fed by Focusrite iTrack Solo audio interfaces. Each subject evaluated 18 randomly selected utterances from a male and a female datasets, respectively, resulting in 36 MUSHRA screens per subject. The stimuli evaluated in each screen is presented on Table 3.1. The instructions handed to the participants is in Appendix A.

Table 3.1: Stimuli per MUSHRA screen for Experiment

Name	*Description*
Nat	Natural speech (the hidden reference).
Base	The baseline system running at constant frame rate and using STRAIGHT for analysis/synthesis.
PM	The proposed method with "ideal" settings.
PMVNAp	The proposed method without using aperiodic components in voiced speech. During synthesis we bypassed the LPF filter in the *periodic spectrum generation* (See Figure 3.10), and constrained the *aperiodic spectrum generation* to work only for unvoiced speech (VNAp="voiced no-ap").
PMVNApW	The proposed method without using the narrower analysis window presented in Section 3.3.3.3 (VNApW="voiced no-ap-win").

For all the systems under test, the postfilter included in the Merlin toolkit was applied (Koishida et al., 1995). In the case of the male speaker, the postfilter moderately affected unvoiced speech regions, thus high frequencies were slightly boosted there to compensate. At the output, the synthesised signal was high-pass filtered to protect against any spurious spectral component that could appear below the voice frequency range[4].

[3]Code available at `http://dx.doi.org/10.7488/ds/1316`

[4]Freely available demo audio samples at `http://www.felipeespic.com/demo_fft_feats_2017`

3.4.2 Results

One of the subjects was rejected from the analysis due to inconsistent scores: Nat was rated <20% several times. Also, Holm-Bonferroni correction was used because of the large number of tests (18×29 per voice). A summary of the systems average scores for both voices is given in Table 3.2.

TABLE 3.2: Average MUSHRA Score Per System in Evaluation

	System				
Speaker	Nat	Base	PM	PMVNAp	PMVNApW
Male	100	43.6	**51.4**	45.6	49.4
Female	100	32.6	**43.8**	34.6	43.1

Figures 3.11 and 3.12 show the mean, the median, and the dispersion of the scores per system under test, for the male and female voices, respectively.

Significance tests indicate that all the configurations of the proposed method significantly outperformed the baseline for both voices. The highest scores were achieved by the proposed system with "ideal" configuration, PM, which was significantly preferred over all other systems, except for the female voice where PM and PMVNApW were not found to be significantly different. The PMVNApW system was significantly preferred over PMVNAp and the baseline for both voices.

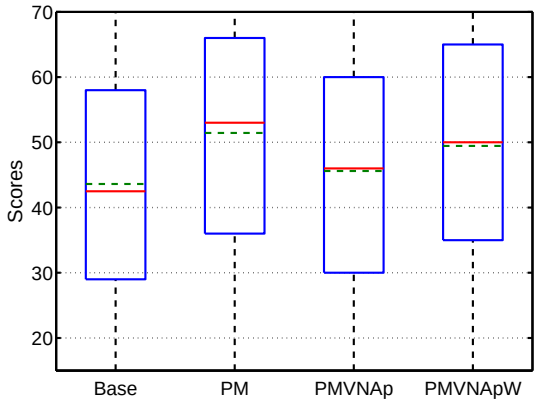

FIGURE 3.11: Absolute scores for the male voice. The green dotted line is the mean, and the continuous red line is the median. Natural speech is omitted (mean score is 100) and the vertical scale is limited to 15-70, for clarity.

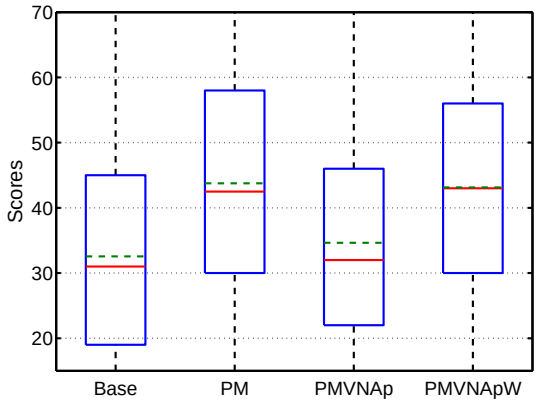

FIGURE 3.12: Absolute scores for the female voice. The green dotted line is the mean, and the continuous red line is the median. Natural speech is omitted (mean score is 100) and the vertical scale is limited to 15-70, for clarity.

3.4.3 Efficiency

In addition to listener preference, an important aspect to evaluate is the efficiency of the proposed system compared to the baseline. As the former works pitch synchronously, its frame rate is variable, while the baseline runs at a constant 5 frames per second. Hence, it is useful to test if on average, its frame rate is higher or lower than the baseline, at least for the available voices. That will give an indication of how efficient the proposed method is. Table 3.3 shows the average durations, number of frames and frame rates for the proposed system and the baseline. All the numbers are averages across utterances. The data used was the test set for the male and female speakers.

For the male speaker, the proposed method highly outperforms the baseline. It decreases the number of frames per second by a relative 31.5%. In the case of the female speaker, both systems perform equally, both running at 200 frames per second. In summary, the proposed method decreases the frame rate to a relative 16.0% when comparing with the baseline.

TABLE 3.3: Frame Rates Per System in Evaluation

	System					
	Base			PM		
Speaker	No. Frms	Dur	Rate	No. Frms	Dur	Rate
Male	424	2.11	**200**	303	2.20	**137**
Female	534	2.67	**200**	544	2.71	**200**
Average	479	2.39	**200**	423	2.46	**168**

3.5 Conclusion

We have proposed a new waveform analysis/synthesis method for SPSS, which encodes FFT magnitude and phase spectra into four feature streams (including f_0). It can be considered as a step forward towards direct waveform modelling for SPSS.

It does not require the estimation of high-level parameters, such as spectral envelope, aperiodicities, harmonics, etc., used by vocoders that attempt to decompose the speech structure, using source-filter separation or harmonics+noise model.

The subjective results have showed that the proposed method clearly outperforms a state-of-the-art SPSS system that uses the STRAIGHT vocoder, for a female and a male voice. It gets rid of the "buzziness" and "phasiness" typical of vocoders, delivering a more natural sound.

It does not require any iterative or estimation process beyond the epoch detection performed during analysis. Indeed, it mainly relies on cheap operations, such as: FFT, OLA, and IFFT, allowing short run time, especially during synthesis.

In addition, the proposed method decreases the frame rate, reaching an impressive reduction of 31.5% for the male speaker, when comparing with typical SPSS. It is clear that in all cases for speakers with lower average pitch than the tested female voice, the proposed method will run at a beneficial slower frame rate.

The proposed method does not use any heuristics or unstable methods for phase modelling typically required for phase unwrapping or group delay computations. We also show the importance of proper modelling of aperiodic components during voiced speech, illustrated by the poorer results obtained by the systems that did not include it, although they still outperformed the baseline.

It was necessary to create a new frame positional feature for input to the neural network, which embeds two sub-features in one numeric value. It simplified the system by reducing the number of frame positional features by 4.

The proposed method, as a reliable representation of FFT spectra, could be used for a series of other applications related to audio signal processing, such as: noise-reduction, automatic speech recognition, voice conversion, etc.

Chapter 4

MagPhase Vocoder: Improvements to the Original Design

After completing the original design of the MagPhase vocoder, several changes were implemented to improve its quality and/or investigate other available options in the same framework. Among these improvements are:

4.1 Optimal Use of Phase-Derived Features

4.1.1 Motivation

In vocoding, it is usually desired to have the least possible number of acoustic features because:

- Neural networks need less data points for training if the number of output features is lower, i.e., avoiding data sparsity. In practice, the network can be trained faster and with a smaller database. On the contrary, if more features need to be predicted, more examples need to be seen by the network making the training more difficult. However, in some cases, some redundancy in the output layer is desired to make the system more robust due to the dependency of the redundant features, but it is preferred to have a smaller output layer in the general case.
- Having a smaller output layer means less computational burden, which is critical for real time applications. Less features to be predicted means less number of operations and memory required during synthesis.

- In case the vocoder is used for transmission, the lower the number of features is, speech is transmitted more efficient.

However, the resulting speech quality maybe degraded by lowering the number of features too much, since they could not be able to represent the perceivable details of speech. Thus, there is a trade-off between efficiency and speech quality that needs to be investigated.

Even though the MagPhase vocoder is capable of extracting lower-dimensional features from the original high resolution features, it still requires a high number compared to another vocoders. Moreover, this number may be increased when using acceleration features (deltas and delta-deltas) typically needed in feed-forward neural network-based SPSS. These are even used with recurrent neural networks-based systems in certain conditions. See Section 1.3.1.3 for more details.

Table 4.1 shows a comparison between the number of features output by the network using each vocoder, respectively. It is noticeable that the number of acoustic features when working with the MagPhase vocoder is much higher than for the other vocoders. This suggests that the number of features generated by MagPhase vocoder is too high and is a clear disadvantage comparing to the other vocoders, with which the network needs to learn considerably fewer features.

We observe that the dimensionality of the feature M matches with some of the features describing the magnitude spectrum from other vocoders, such as MCEP with STRAIGHT and WORLD. This situation repeats for the scalar feature $f0$. Consequently, we conclude that the phase derived features R and I are the ones that make the difference and their high dimensionality needs to be addressed.

The obvious way to decrease the dimensionality of phase-derived features is by using fewer Mel-Frequency bands to represent them. As explained in Section 3.3.1.8, the number of bands of phase derived features follows the same characteristics of the Mel-warping applied to the magnitude spectrum derived feature M_c. That is, it uses the same number of frequency bands and the same α value. However at the end, the highest frequency bands are removed since they are not necessary. The frequency bands of all spectral features M_c, R_c, and I_c coincide in location in the Mel frequency domain. By decreasing the number of bands used in the Mel-compression of phase, this synchronism is lost. Nevertheless, it is expected that it should not be detrimental enough to produce unwanted effects. The parameter α is kept the same. Figure 4.1 shows an example of the feature R recovered from different numbers of Mel-frequency bands. From the figure, it is clear that even if the dimensionality is lowered from high resolution (1+fft_length/2)

to 45 or 20 coefficients, the shape of the curves resembles the high resolution one, at least up to the bin 400 (∼4.7kHz).

After informal subjective evaluations, it was not clear which is the "sweet spot" between frequency resolution and number of features to be inferred by the network. Hence, we proceeded to carry out formal subjective experiments, which are detailed in Section 4.1.2.

TABLE 4.1: Typical number of acoustic features used by the SPSS System

Vocoder	Feature Stream	No. extracted features[2]	Additional voi/unvoi[3]	No. deltas[4]	**Total**
STRAIGHT	MCEP:60 BAP:25 F0:1	86	1	261	**348**
WORLD	MCEP:60 BAP:5 F0:1	66	1	201	**268**
GlottDNN	Energy:1 F0:1 HNR:25 VSS:10 VTS:50	87	1	254	**342**
MagPhase	M:60 R:45 I:45 F0:1	151	1	456	**608**
Ahocoder[1]	MFCC:40 F0:1	40	1	123	**168**

[1] *The values reported for Ahocoder are at 16kHz sample rate, while for the rest of the vocoders is reported running at 48kHz.*

[2] *Number of acoustic features extracted by the vocoder.*

[3] *Additional voiced/unvoiced decision feature needed to support $f0$ modelling in SPSS.*

[4] *Total number of acceleration features, including deltas, delta-deltas, etc.*

4.1.2 Experiments

In this section, we presented a way to decrease the number of features that are needed to be inferred by the network. Specifically, we reduced the number of features for phase on R_c and I_c. Now, we need to find the "sweet spot" between the number of features and the quality of the representation for SPSS.

4.1.2.1 Evaluation

The following experiment evaluates the subjective performance in terms of *naturalness* achieved by different resolutions of the MagPhase features R_c and I_c. Also, a standard vocoder is added to the list of stimuli as a baseline. The type and details of the experiment are similar to the subjective evaluations carried out in previous chapters.

One SPSS male voice "Nick" was built by using the Merlin toolkit (Wu et al., 2016) running at 48kHz sample rate. A 5 layer network was used with 4 feed-forward layers plus a SLSTM layer in top (Wu and King, 2016). Each feed-forward layer contained 1024 units, whilst the SLSTM layer had 512 units. The voice was built by using 2400 sentences for training the model, 70 for validation, and 71 for testing.

In order to make a fair comparison, the same architecture and database were used to build voices with the proposed method MagPhase, and the baseline vocoder STRAIGHT. The dimensionality of the MagPhase features used for the experiment were: $M_c = 60$, $f_{0c} = 1$, while the dimensionality of the features R_c and $I_c = 45$ was varied as described

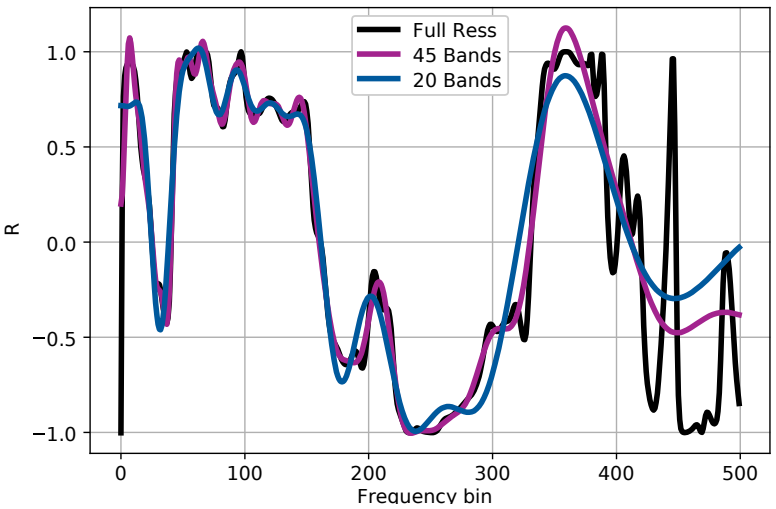

FIGURE 4.1: Example of the R feature recovered from different resolutions. Full Ress: Full resolution. 45 Bands: Compressed using 45 Mel-Frequency bands. 20 Bands: Compressed using 20 Mel-Frequency bands.

in Table 4.2. For the baseline STRAIGHT, the features used were of sizes: MCEP=60, BAP=25, lF0=1.

Eighteen university students were hired to evaluate 5 stimuli per screen in a MUSHRA-like test. The listening tests were carried out in sound-proof booths, in which a desktop computer connected to an audio interface Focusrite iTrack Solo and headphones Beyerdynamic DT 770 were provided. They were asked to score the *naturalness* of the configurations on a scale from 0 to 100. Each listener evaluated 30 MUSHRA screens. The order of screens and stimuli was randomised to avoid any psychological factor that may affect the result. Appendix A contains the instructions provided to the listeners during the study. A summary of the stimuli presented per screen is detailed in Table 4.2.

Table 4.2: Stimuli per MUSHRA screen for Experiment - Optimal Use of Phase-Derived Features

Name	*Description*
Nat	Natural speech as the hidden reference.
Base	Using STRAIGHT vocoder as a baseline.
PM45	Original setup. The proposed method (PM), MagPhase vocoder, using 45 coefficient per each phase derived feature R_c and I_c.
PM20	The proposed method (PM), MagPhase vocoder, using 20 coefficient per each phase derived feature R_c and I_c.
PM10	The proposed method (PM), MagPhase vocoder, using 10 coefficient per each phase derived feature R_c and I_c.

4.1.2.2 Results

The Holm-Bonferroni correction was used due to the large number of tests carried out (30×18). The average scores per system were 100% for Nat, 31.8% for the baseline STRAIGHT, and 42.4%, 39.3%, and 34.6% using the MagPhase vocoder with the configurations PM45, PM20, and PM10, respectively. Figure 4.2 shows the scores mean, median, and dispersion per system under test.

4.1.2.3 Discussion

All systems under test were significantly different to each other ($p<0.05$). All the MagPhase configurations MP45, MP20, and MP10 outperformed the baseline, STRAIGHT,

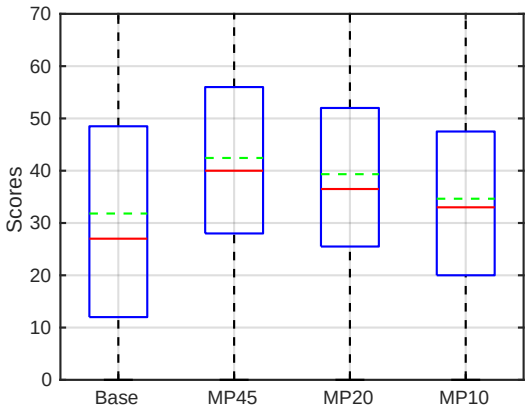

FIGURE 4.2: Absolute scores for the built voice. The green dotted line is the mean, and the continuous red line is the median. Natural speech is omitted (mean score is 100) and the vertical scale is limited to 0-70, for clarity.

showing the consistency in performance of our proposed method. Overall, the highest scored system was MP45, which shows that by using higher dimensional MagPhase phase-derived features, the perceived naturalness is increased. Then, scores decreased gradually with the systems MP20, and MP10 showing correlation between sound quality and dimensionality of phase-derived features. In addition, we can conclude that higher dimensionality does not critically affect the quality of the inference performed by the acoustic model, at least compared to the benefits of having a higher resolution phase descriptors.

4.2 Frame Rate Handling

4.2.1 "Fake" Alignments

One of the peculiarities of the vocoder is that it uses a special type of context frame features, as described in the Section 3.3.4, because of its pitch synchronous process (variable frame rate). As an alternative to this, we developed a method to make the vocoder work in a standard fixed frame rate fashion, as commonly used in e.g. the Merlin toolkit (Wu et al., 2016). It consists in preprocessing the state-aligned labels [1] by modifying ("faking") the durations, such that the SPSS toolkit will process the acoustic features as if they were extracted at a constant frame rate. Assuming that the state-aligned labels are already extracted, the method can be summarised as:

1. Extract MagPhase features from the whole database.
2. Compute the number of variable-rate frames extracted by MagPhase per state, by comparing the locations of the epochs extracted by MagPhase (i.e., REAPER) with the locations of state boundaries.
3. Modify the durations in the state-aligned labels to contain the corresponding number of frames extracted by MagPhase, multiplying them by the constant frame rate. Usually a frame rate of 5ms is used.

Figure 4.3 shows an example of the generation of "fake" state-aligned durations for two states in the phoneme /u/. The SPSS toolkit works at 5ms frame shift, and accepts the state durations in milliseconds. The duration of state 1 is 20ms, which is interpreted by the SPSS toolkit as four frames ($20ms/5ms$). However, the MagPhase vocoder, or more precisely REAPER, has detected only three epochs, at 107ms, 113ms, and 117ms within the state ($100ms-120ms$). Thus, the SPSS system needs to process three frames instead of four. Accordingly, the time boundaries are adjusted to cover the duration of three frames running at 5ms frame shift ($3 \times 5ms$), which is 15ms.

Then, the training of the system can be performed using the "fake" labels as input, so the system will treat the vocoder as a constant frame rate device, even though it internally works in a variable frame rate fashion. That conveniently keeps the internal workflow using 9 context frame features commonly used by the Merlin toolkit.

[1] They are usually computed by a 5-state HMM-based aligner, which as a result gives the durations of each state within each phoneme.

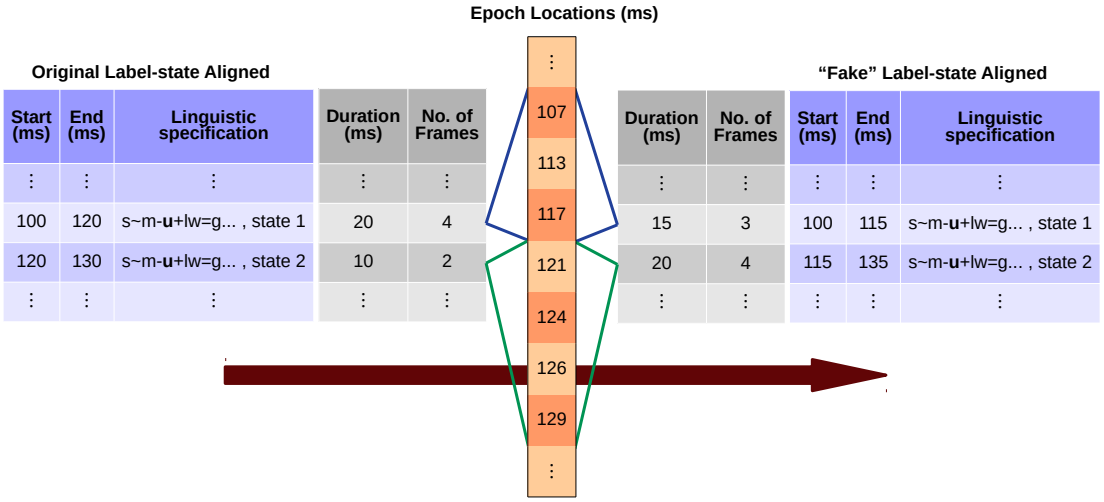

FIGURE 4.3: Example of the generation of "fake" alignments for two states in the phoneme /u/. The SPSS toolkit works at 5ms frame shift.

4.2.2 Resampling to Constant Frame Rate

One step further in making the vocoder work in a more standardised fashion is by producing and processing constant frame rate features directly. For the feature extraction, the extracted variable-rate features are resampled at a constant frame rate. For synthesis, the constant frame rate features are resampled to variable frame rate before processing by the vocoder itself. This conversion is carried out by following the epoch locations given by Equation 3.16. Our aim is to keep the signal processing as simple as possible, and efficient, so only linear and spline interpolators were used.

The implementation of a resampling routine was attempted in the beginning of the development of this vocoder. Unfortunately, it did not at that time show promising results; the interpolation produced highly audible artefacts leading us to abandon the idea. However, motivated by the community and after already completing some of the improvements mentioned in this chapter, we decided to investigate the causes of failure, and find a more suitable method for resampling.

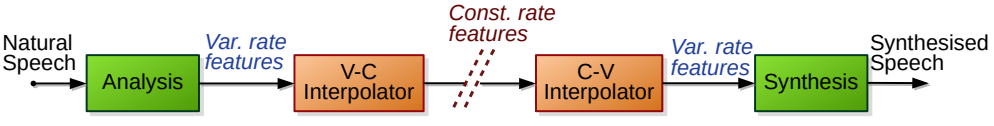

FIGURE 4.4: Conversion to and from constant frame rate in the context of copy-synthesis.

4.2.2.1 High Resolution Features

The high resolution MagPhase features (Section 3.3.1.7) encode the Short-Time Fourier Transform (STFT) in a lossless manner, thus some properties of the STFT need to be taken into account.

The Discrete Fourier Transform (DFT) can be interpreted as a filter-bank, in which each frequency bin is centred at a band-pass filter. Ideally, these filters should look in the frequency domain like perfect non-overlapping rectangles, so that they could be perfectly selective in the frequency range where they are defined. However, in practice that is not possible due to the finite nature of DFT analysis, which inherently imposes a rectangular window to the signal. Even though another more sophisticated window can be applied instead, e.g., Hann or Blackman-Harris, the frequency response of each band-pass filter spreads over the whole spectrum due to the window effect, behaving far from the ideal highly selective filter. This effect is called "Spectral Leakage". See an example in Figure 4.5 showing the spectral leakage generated when using a Hann window. It is noticeable from the figure that the spectral leakage spreads over the whole spectrum.

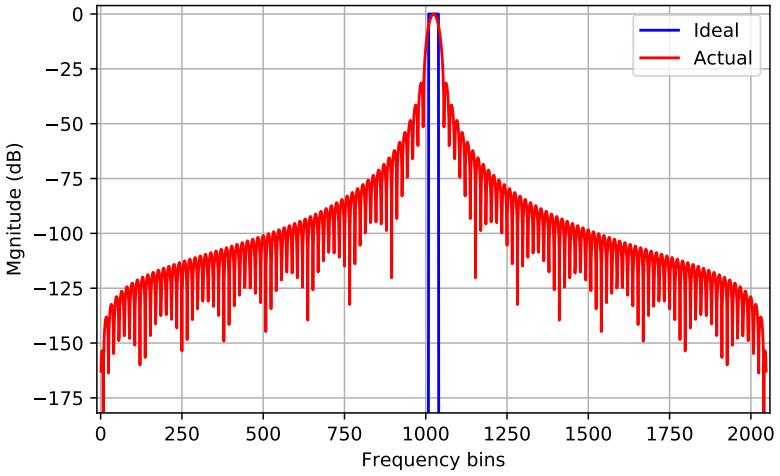

FIGURE 4.5: Example of spectral leakage. Ideal: Ideal frequency response of a band-pass filter part of the DFT filterbank. Actual: Actual frequency response of the filter.

In the case of analysis/synthesis this effect is cancelled out, since both the analysis and synthesis filters exhibit exactly the same shape, leading to an analysis/synthesis process that is completely invertible and lossless. However, if some modification is introduced in the frequency domain before synthesis, it will affect other frequency bins, which may lead to audible artefacts.

For a high resolution spectrum, the difference between neighbouring frequency bins can be substantial, especially in frequency ranges where the spectrum behaves mainly

stochastically, thus any change in a high energy frequency bin may highly affect ("leak" into) other bins originally exhibiting low energy.

In this context, time-domain interpolation (Figure 4.4) leads to the generation of non-existent spectral features by taking the spectra of neighbouring frames as reference. In other words, it can be seen as a midpoint adaptation of the features from one natural speech frame to the next natural speech frame(s), yielding to artificially generated acoustic features. As a result, this modification produces some of the drawbacks mentioned in the last paragraph, which are more pronounced in unvoiced speech and also high frequencies for voiced speech.

Another factor to consider is the high degree of change between the high resolution spectra of two consecutive frames. That forces the interpolator to impose strong modifications to the features extracted from natural speech.

Either upsampling or downsampling is applied depending on the interval between two consecutive epoch locations and the target constant frame rate (e.g., 5ms). In case of upsampling, even though it is safe in terms of aliasing, it creates new data in the same lower band as the original sampled data, being unable to recover rapidly changing variations that may occur just over the point of interpolation.

The most common case in the vocoder is downsampling, that is the variable frame rate is lower, on average, than the target constant frame rate (See Table 3.3). As in any downsampling operation, it risks aliasing. It is usually avoided by using a low-pass filter, in this context called an "anti-aliasing" filter, that removes signal contents over the Nyquist frequency. By doing so, any spurious folded spectral component is filtered out before performing the downsampling. Nevertheless, in the case of this vocoder, anti-aliasing is very difficult to implement since the relation between the target constant frame rate and the original variable frame rate changes dynamically over time, making the implementation of an optimal anti-aliasing filter very difficult. This drawback appears either when resampling during feature extraction (i.e., variable to constant frame rate), or synthesis (i.e., constant to variable frame rate).

4.2.2.2 Low Resolution Features

As explained in Section 3.3.1.8, low dimensionality is achieved by compressing the frequency axis using the Mel-scale, consequently some smoothing is applied towards higher frequencies. This acts as a damping, which reduces the difference between neighbouring frequency bins, reducing the artefacts produced by the spectral leakage typical in high resolution spectral features.

As a collateral effect of the Mel-scale frequency warping, the spectral difference of consecutive frames is decreased considerably, damping the modification required to generate the interpolated spectral data. As a result, audible artefacts are reduced. Hence, taking into account also its effect over the spectral leakage, the Mel-scale compression works as a 2D spectro-temporal smoothing filter, which reduces the amount of modification necessary to produce new interpolated spectral data.

As pointed out in Section 3.3.3.3, for the lower resolution features, the vocoder uses artificially generated noise to replace the natural occurring stochastic components. It ensures high magnitude and phase coherence if the modifications required evolve slowly (smoothly) enough in time and frequency. This differs to the high resolution features, for which the vocoder during synthesis uses the natural features directly.

As a conclusion, for SPSS we decided to perform the resampling directly over the low-dimensional acoustic features, rather than applying it in high resolution domain and then lower the dimensionality. This decision also was supported by informal experiments, which made clear that the selected method worked better.

4.2.3 Experiments

In this chapter, two methods for frame rate handling were proposed with the objective of improving the compatibility between the MagPhase vocoder and standard SPSS toolkits. We proposed the warping of state-aligned durations, which we called "fake" alignments, and also the conversion of the variable frame rate features to constant frame rate by resampling.

4.2.3.1 Evaluation

A subjective evaluation was performed similarly as the previous experiments. Twenty one university students, native speakers, were hired as listeners to assess the degree of *naturalness* achieved by several stimuli generated by different SPSS systems. The same setup as the experiment in Section 4.1.2.1 was used, with only exception of the stimuli presented to the subjects, which is detailed in Table 4.3. For details on the setup, please refer to Section 4.1.2.1. Also, for MagPhase vocoder, we used 45 dimension phase-derived features R_c and I_c.

TABLE 4.3: Stimuli per MUSHRA screen for Experiment - Frame Rate Handling

Name	*Description*
Nat	Natural speech as the hidden reference.
Base	Using STRAIGHT vocoder as a baseline.
VR+MCFF	Original setup. MagPhase vocoder running at variable frame rate (VR), and using the modified context frame features (MCFF), as described in Section 3.3.4.
VR+FA	MagPhase vocoder running at variable frame rate (VR), and using "fake" alignments (FA), as described in Section 4.2.1.
CR	MagPhase vocoder running at constant frame rate (CR) as described in Section 4.2.2.

4.2.3.2 Results

Because of the large number of tests (30×18), Holm-Bonferroni correction was applied. The average scores per system were 100% for Nat, 27.4% for the baseline STRAIGHT, and 43.7%, 42.0%, and 35.9% using the MagPhase vocoder with the configurations VR+MCFF, VR+FA, and CR, respectively. Figure 4.6 shows the scores mean, median, and dispersion per system under test.

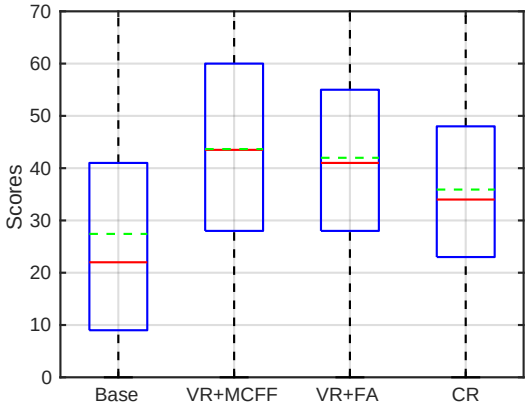

FIGURE 4.6: Absolute scores for the built voice. The green dotted line is the mean, and the continuous red line is the median. Natural speech is omitted (mean score is 100) and the vertical scale is limited to 0-70, for clarity.

4.2.3.3 Discussion

The scores of all the systems under test were statistically different ($p<0.05$). Overall, all the variations of the MagPhase vocoder outperformed the baseline, STRAIGHT, which again shows the robustness of the our proposed system, even in different variations. The configuration VR+MCFF was the highest scored system, showing that the original design is the one that achieves the highest quality. VR+FA was scored just below VR+MCFF (42.0% - 43.7%) meaning that by simply "faking" the labels is an efficient technique that can be used in different standard constant frame rate-based systems. Both variable rate configurations of the MagPhase vocoder, VR+MCFF and VR+FA, outperformed its constant frame rate counterpart CR. This coincides with our intuition that by using extra signal processing in vocoding, speech quality is degraded. Even though CR was the worst scored MagPhase variation, it still outperforms the baseline.

4.3 Phase Coherence

The Griffin-Lim algorithm proposed by Griffin and Lim (1984) has been recently used to recover the phase spectrum from the magnitude spectrum (e.g., Takaki et al., 2017; Wang et al., 2017). It relies on the phase coherence between the complex spectrum of adjacent frames analysed by the Short-time Fourier Transform (STFT). The methods is an iterative algorithm, which minimises the mean squared error between the magnitude spectrum of a speech frame with the magnitude spectrum of the same frame overlapped with its contiguous frames.

One of the possible applications of the Griffin-Lim algorithm in our work is the improvement of the already estimated phase spectrum. That is, we can try running the Griffin-Lim algorithm over the estimated complex spectrum derived from MagPhase features. Obviously, with high resolution MagPhase features lossless copy-synthesis is achieved, so it is not worthy applying Griffin-Lim in that case. However, the algorithm is an excellent candidate to improve the quality achieved from using MagPhase features if they were *inferred* by an acoustic model (e.g., neural network).

The implementation consisted in applying the Griffin-Lim algorithm taking the synthesised signal and the magnitude spectrum derived from the feature M'_c as inputs. Then, it iterates reducing the difference between the estimated magnitude spectrum M', and the magnitude spectrum computed from the intermediate synthesised signals. The diagram in Figure 4.7 shows the general workflow of the algorithm, where x^i is the synthesised signal only using MagPhase features, M' is the estimated magnitude spectrum, X^i is the complex spectrum of one frame of x^i, and mod-OLA is a slightly modified version of the Overlap-Add method (details in Griffin and Lim, 1984). The resulting modified signal is taken from the point of x^{i+1} after several iterations (e.g., 50).

Even though the method seemed to be promising, the result was not satisfactory, at least in informal experiments. Actually, the algorithm proved to dramatically worsen the quality already achieved by using only the estimated MagPhase features. Furthermore, in spite of the simplicity of the algorithm, its iterative nature makes it computationally expensive. Consequently, we decided not to perform formal experiments to evaluate the effect of using the Griffin-Lim algorithm over MagPhase features.

4.4 Conclusion

We have implemented different modifications to the original design with different aims, such as: improve synthetic speech quality, improve compatibility with TTS systems, and test different ways of using our proposed method.

In terms of quality, the original design/setup (i.e., PM45, VR+MCFF) remained as the best system according to the results of the experiments presented in this chapter. It means that the highest achievable quality using our proposed system is done by adapting the code of the TTS toolkit to work directly at a variable-frame rate. That coincides with our intention of avoiding unnecessary processes to produce higher quality speech.

In terms of compatibility, systems variable frame-rate + "fake" alignments (VR+FA) and the constant-frame rate (CR) are alternatives highly compatible with standard TTS systems. Even though their performance was not as good as the original system VR+MCFF, the naturalness achieved by these was rated higher than the baseline.

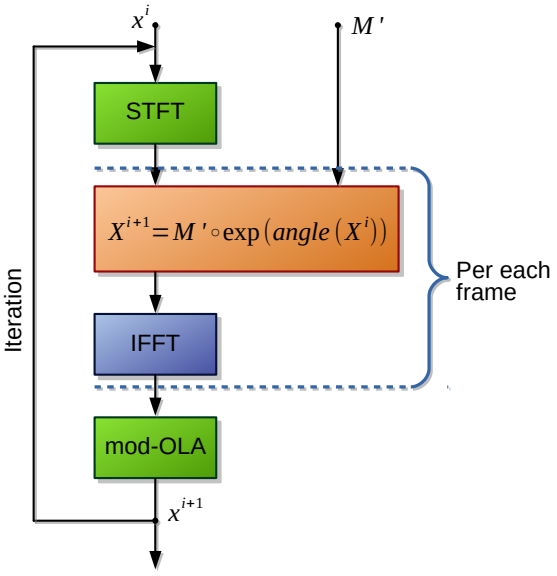

FIGURE 4.7: General diagram of the Griffin-Lim algorithm applied in top of MagPhase features.

Chapter 5

MagPhase Vocoder: Other Applications

So far, the MagPhase vocoder has been designed, described and tested only for Statistical Parametric Speech Synthesis (SPSS). After the implementation of the improvements developed in the previous chapter, we thought that, as a high quality speech processor, it potentially could be used in other tasks. In this chapter, several applications of the vocoder in other tasks are presented and described.

5.1 Unit Concatenation and Join Smoothing for Unit Selection-Based Systems

This section describes a collaborative work with members of CSTR, published in Ronanki et al. (2017); Espic et al. (2018).

Unit selection-based systems are still the most used in production nowadays. Especially, hybrid systems have shown great quality at a moderate computational cost. Hybrid systems combine the stability of SPSS systems with the high quality waveforms produced by raw unit selection. They work by predicting acoustic features as commonly done in SPSS, and then these are used to compute the target cost to select suitable speech units to be finally concatenated.

For the Blizzard Challenges of 2017 and 2018, the CSTR entries (Ronanki et al., 2017; Espic et al., 2018) were hybrid TTS systems, in which the selected speech units are concatenated and post-processed by using the MagPhase vocoder to generate the synthesised waveform. During synthesis, the system comprises:

1. SPSS: Prediction of durations.
2. SPSS: Prediction of acoustic features.
3. Viterbi search for units.
4. Concatenation and smoothing.

To train the acoustic model in the SPSS stage, 60 Mel-cepstral coefficients (MCEPs), 25 band-aperiodicities (BAPs), and one fundamental frequency scalar are extracted from natural speech by using the STRAIGHT vocoder (Kawahara et al., 1999a,b). For each feature stream, deltas and delta-deltas are appended, as well.

A feed-forward neural network was trained to infer acoustic features from linguistic features (regression) using the mean-squared error (MSE) as loss function. During synthesis, the trajectories of the predicted STRAIGHT features are smoothed by using MLPG (See Section 1.3.1.2), and finally post-filtered. Typically, these features would be fed into a vocoder then generating the final synthesised waveform. Instead, in this hybrid synthesiser, these features are used as targets for the unit selection stage.

For the unit selection stage, the halfphone variant from the system described in Valentini-Botinhao et al. (2018) was used. The speech is state-aligned beforehand using a 5 state HMM-based aligner. Then, the first two states are assigned a halfphone and the next three are assigned to another halfphone.

As usual, unit selection-based systems compute the target cost to measure the degree of closeness of candidate units to the target speech characteristics, and the join cost to evaluate how well two units would concatenate. In order to compute the target cost, two STRAIGHT feature streams are used: Mel-Cepstral Coefficints (MCEP), as a representation of the magnitude spectrum, plus the log- fundamental frequency.

MagPhase features are used to calculate the join cost. These are the Mel-warped log magnitude spectrum M_c, log-fundamental frequency f_{0c}, and the phase derived features R_c and I_c, all of them extracted pitch synchronously. It is expected that these two phase-derived feature streams will yield a sequence of speech units with fewer phase discontinuities, thus reducing glitches.

As the SPSS stage is frame-based, and the unit selector operates at a halfphone level, it is necessary to map from frame based features to halfphone timings. Each halfphone is represented by 3 frames extracted from the beginning, middle and end of the halfphone. Their exact locations depend on the the boundaries of the HMM states derived from the forced alignment.

In addition to this data, also stored are the identifiers of speech units: the start and end samples of the time domain signal, symbolic phonetic identity of each unit, and the position of halfphone within the phone, left or right. The weighting of elements for the target and join costs is described in Watts et al. (2018).

Unit search is performed using a Weighted Finite-State Transducer (WFST) that combines the target and join costs in one graph. Both costs are respectively weighted Euclidean distances between target and join representations.

5.1.1 Waveform Generation

This section is the core of my contribution to this research. After the speech units have been selected, they are parameterised by the MagPhase vocoder. In the current implementation, this is performed on-the-fly, but there is nothing to prevent doing it beforehand (off-line).

The frame-based analysis is carried out pitch synchronously and extracts high resolution (lossless) acoustic features that represent the fundamental frequency and the complex spectrum. The join smoothing (glitches reduction), is produced by the concatenation and correction of speech units in the MagPhase features domain.

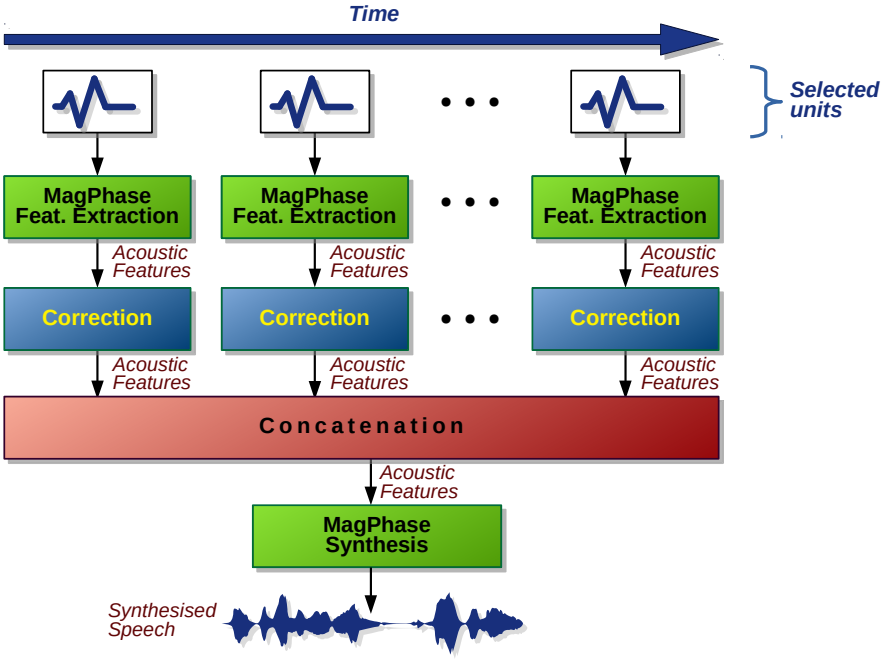

FIGURE 5.1: General diagram of the waveform generation stage using MagPhase features. The correction applied depends on the feature, either f_0 or complex spectrum.

5.1.1.1 Concatenation and correction of f_0 contours

One of the most relevant factors that produces audible glitches in unit selection, are the jumps in the fundamental frequency trajectory, which occur around the locations of the joins of speech units. These are perceived as discontinuities, often not clearly assigned to a jump in pitch. Basically, the discontinuities are generated by highly different neighbouring f_0 values as a result of the concatenation of voiced speech units. An example of this artefact is illustrated by the blue curve in Figure 5.2.

In order to describe the proposed method, let us take two consecutive selected units. The method consists of modifying the f_0 contours of two units to smooth their transition. The fundamental frequency mid point, f_{0m}, between the two consecutive speech units is defined by:

$$f_{0m} = \frac{f_{0p[N_p-1]} + f_{0c[0]}}{2} \tag{5.1}$$

Where p means preceding unit, c current unit, and N is the length of the speech unit in frames. f_0 values per frame are zero indexed in $[\cdot]$ relative to the unit it belongs to. In simplest terms, f_{0m} is the average between the f_0 value of the last frame in the preceding unit p, and the f_0 value of the first frame in the current unit c. Then, the slope of the f_0 contours of both units are adjusted to reach the f_{0m} value just in the join location. The corrected f_0 contours are computed by:

$$\tilde{f}_{0c[n_c]} = f_{0c[n_c]} + (f_{0m} - f_{0c[0]}) \cdot \left(\frac{n_c}{1 - N_c} + 1 \right) \tag{5.2}$$

$$\tilde{f}_{0p[n_p]} = f_{0p[n_p]} + (f_{0m} - f_{0p[N_p-1]}) \cdot \left(\frac{n_p}{N_p - 1} \right) \tag{5.3}$$

Where $\tilde{f}_{0p}$ is the corrected f_0 contour of the previous unit, $\tilde{f}_{0c}$ is the corrected f_0 contour of the current unit, and n is the frame index within the unit. The described process is performed sequentially from the beginning of the utterance, processing the units in pairs (p and c). As a result, the f_0 curve of each unit is modified twice, once to correct for the join with the previous unit, and once to correct for the join with the next unit.

After having all the corrected f_0 contours for all the units, these are appended building a single f_0 contour for the whole utterance. Figure 5.2 shows an example of f_0 fixing, resulting in the corrected f_0 (red). It is noticeable how the method is able to correct the concatenated f_0 contour (blue) by modifying the f_0 slopes around the join locations (black vertical lines).

5.1.1.2 Spectral concatenation and smoothing

The spectral concatenation and smoothing is performed by time-domain overlapping and crossfading the FFT complex spectra of two consecutive units. Some extra frames are extracted from the sources, so the units can be overlapped without affecting their expected durations and locations in the synthesised waveform. Three extra frames on each side of the speech units are extracted from the sources, thus an overlap of seven frames around the joins is produced.

For each frame, the FFT complex spectrum S is derived from the MagPhase features M, R, and I, by:

$$\mathrm{S} = \mathrm{M} \cdot (\mathrm{R} + \mathrm{I}j) \tag{5.4}$$

A time-domain crossfade is linearly applied to mix the FFT complex spectra of two consecutive units, progressively. It is seven frames long, and in case the unit is too short, the crossfade is shortened accordingly.

This operation is performed sequentially over every join of units from the beginning of the utterance. Consequently, the FFT complex spectra of all the selected units are concatenated producing a single complex spectra stream, that describes the whole utterance.

Finally, the signal is synthesised by converting the FFT complex spectra to the time domain, and applying Pitch Synchronous Overlap-Add by the MagPhase vocoder following the epoch locations derived from the corrected f_0 contour, $\tilde{f}_0$, as described in Equation 3.16. Figure 5.3 shows an example how this correction in the complex domain affects the magnitude spectrum along the utterance.

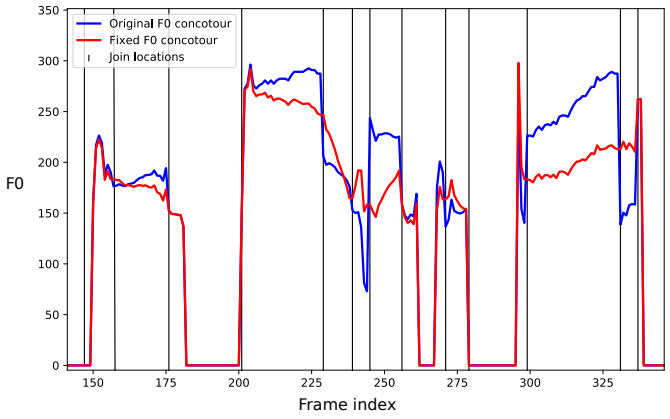

FIGURE 5.2: Example of correction of the F0 contour used as a post-process for an hybrid speech synthesiser.

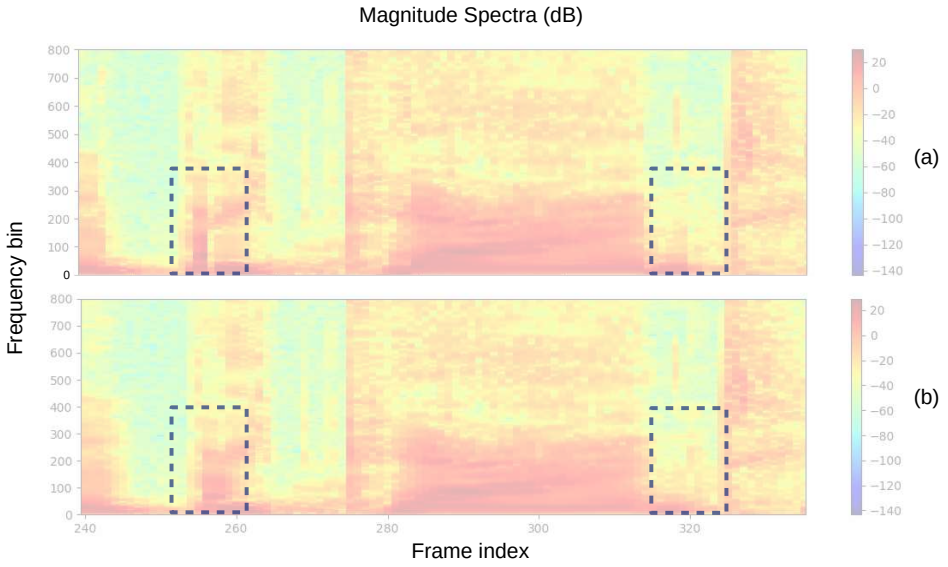

FIGURE 5.3: Example of correction of the complex spectrum used as a post-process for a hybrid speech synthesiser. Magnitude spectrum is used instead for illustration purposes. (a) Original magnitude spectrum of concatenated units. (b) Smoothed magnitude spectrum using the proposed method. The dashed squares indicate areas where the smoothing is highly noticeable.

5.1.2 Experiments

The system was subjectively evaluated in the Blizzard Challenges of 2017 and 2018 (Ronanki et al., 2017; Espic et al., 2018). Listeners were hired to evaluate stimuli synthesised by several speech synthesis systems built using expressive audio books data. Mean opinion score (MOS) was used as a measure of "naturalness", "speaker similarity" on synthesised utterances, "overall impression" on whole paragraphs, and word error rate (WER) for "intelligibility". For further details on the evaluation, refer to King et al. (2017, 2018).

5.1.3 Results

Only the results of the Blizzard Challenge 2018 are presented in this chapter, since they are representative of the current state of the system and how it compares to other state-of-the-art systems. Figures 5.4 and 5.5 show the performance of the proposed system L compared to the other participants on synthesised utterances and synthesised paragraphs, respectively. System A is natural speech, and systems B, C, D, E are benchmarks. The remaining 10 systems are submitted by competing teams coming from industry or academy.

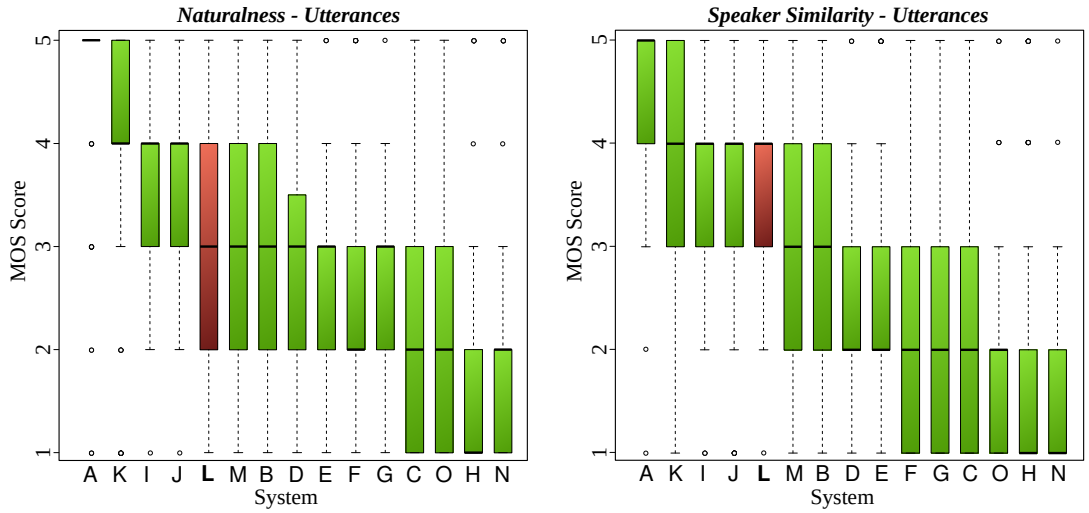

FIGURE 5.4: Results of the Blizzard Challenge 2018: Mean opinion score for naturalness and speaker similarity on synthesised utterances. L is our proposed system (Espic et al., 2018).

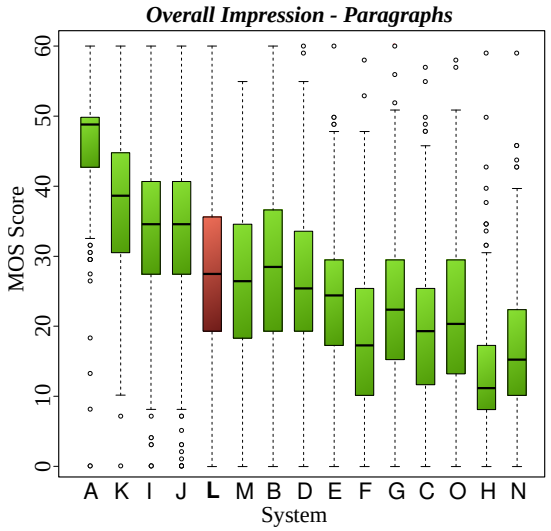

FIGURE 5.5: Results of the Blizzard Challenge 2018: Mean opinion score for the overall impression on synthesised book paragraphs. L is our proposed system (Espic et al., 2018).

In terms of naturalness (Figure 5.4), it is clear that our system showed a decent performance being just outperformed by 3 systems (K, I, and J) over 10 participating submissions. For speaker similarity, only system K was significantly better than ours.

The overall performance obtained by the systems on synthesising audio book paragraphs is shown in Figure 5.5. We observe that similarly to the naturalness scores, now in terms of overall impression, the proposed system was significantly outperformed again only by the 3 systems K, I and J, are the strongest participants in the challenge.

In spite of the good results showed so far, the proposed system did not perform well in

the intelligibility assessment, which is noted by the fact that our submission only outperformed one system, H. It is well known that unit selection systems tend to perform worse than generative systems on intelligibility, due to segmentation and join issues. On the Blizzard Challenge 2018, only 3 out of 10 systems were unit selection (or hybrid), thus it was not strange to see our system among the worst scores in intelligibility. For further details on the results, please refer to Espic et al. (2018).

5.2 Exemplar-based Speech Waveform Generation

This section describes a collaborative work with colleagues from CSTR, published in Watts et al. (2018).

Vocoders are made of two stages: the analyser (feature extraction) and synthesiser (waveform generation). Even though the MagPhase vocoder works as a fully fledged vocoder, it also can be used as a part of a more complex vocoder or waveform generator.

In this section we describe a new method for waveform generation intended to be used as part of an SPSS system. This research was published in Watts et al. (2018), where you can find full details. The proposed method generates the waveform by selecting small speech units, similarly as unit selection systems work. However, rather than choosing larger phonetically determined units, such as halfphones or diphones, the proposed method works with smaller units which are not determined by phonetic annotation. Hence, we can infer potential benefits of the proposed system:

- The unit selection engine is agnostic about the symbolic content of speech.
- The system is able to freely share speech units between different dialects and languages.
- By using smaller units, the system is less sensitive to errors in annotation and low data sizes.

As a unit selection-like system, the proposed method exhibits several differences with previous work on unit selection (e.g., Hirai et al., 2007; Qian et al., 2011; Ling and Zhou, 2018), such as:

- No reliance on phonetic labels.
- No use of dynamic programming. Greedy search is used instead.
- Temporal boundaries of units are defined by speech waveform structure, rather than phonetic alignments.

The proposed method presents some similarities with "neural vocoders" (e.g., van den Oord et al., 2016; Shen et al., 2017), in the sense that both approaches include data-driven waveform generation, and are able to recover missing characteristics from *underspecified* features, e.g. when using MCEP, phase is not present. Also, these systems are able to compensate for imperfectly predicted acoustic features, if matched trained with test data; that is another desirable characteristic of the proposed method.

5.2.1 Proposed System

We use the term "target sequence" to refer to the sequence of acoustic features to be fed into the waveform generator. These features can be either extracted from natural speech or predicted by an acoustic model, as commonly done in SPSS. Thus, the main objective of the proposed system is selecting a sequence of units that minimises the cost between this and the target sequence. As typically used in unit selection systems, the cost function depends on two sub-costs, the *target* and *join* costs, which measure the degree of *fidelity* and *fluency*, respectively.

5.2.1.1 Database preparation

All the audio files are analysed pitch-synchronously by using the MagPhase vocoder. It locates epochs (pitchmarks) by means of REAPER, and then extracts acoustic feature vectors representing the surroundings of each epoch, covering two pitch cycles, as seen in Figure 3.6. Alternatively, for constant-frame rate vocoders, such as WORLD, the acoustic features can be time-domain interpolated to match the epoch locations, thus generating pitch-synchronous acoustic data.

From the acoustic analysis, a *target* and a *join* representation is derived per speech unit, which are used for the target and join cost, respectively. Hence, a combined representation c_i is built for each unit i in the database by concatenating the join representation of the previous unit in the database j_{i-1} with the target representation of the unit t_i.

Also, an extra high-dimensional feature u_i is necessary to be added as a representation for each unit i. It could be any high resolution feature, like time domain signal, or full-resolution spectrum. In practice, we use the high-dimensional MagPhase features (See Section 3.3.1.7), which are retrieved once the units are selected. Then, join smoothing is applied on the fundamental frequency and complex spectrum domains, as explained in Section 5.1.1. Low-dimensional MagPhase features are used for the search.

Although intelligible results are obtained by the concatenation of two pitch cycle segments (surroundings of 1 epoch), better results were obtained by using longer units, covering more pitch cycles. The modification needed to be applied to the system is straightforward: for the *target* representation t_i, the length of a speech unit covers more than two pitch cycles. The *join* representation j_{i-1} keeps the same as originally implemented by just covering two pitch cycles. Typical number of epochs we have used for the new t_i are between 2 and 8. Consequently, the high-dimensional representation u_i also needs to be extended to cover the same number of pitch cycles as t_i.

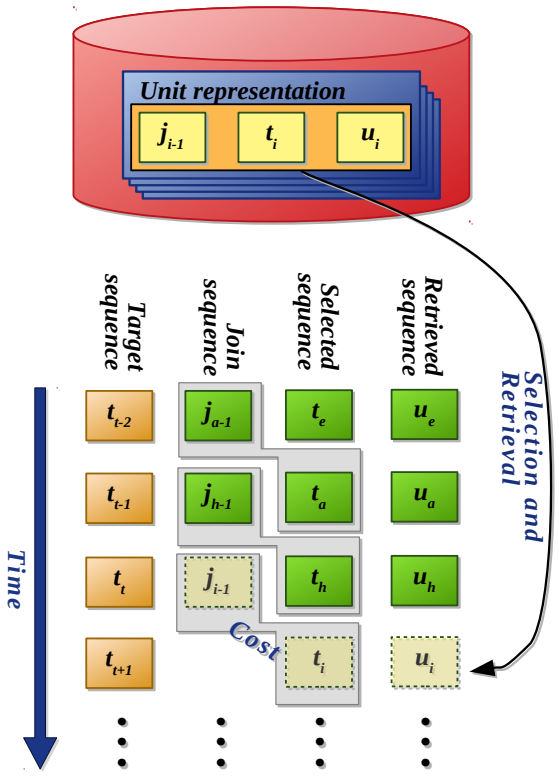

FIGURE 5.6: General diagram of the proposed system.

5.2.1.2 Target and join representations

For the *target* representation, we use only the pitch synchronous low-dimensional MagPhase features f_{0c} and M_c, although features derived from other vocoders could be used instead. For the *join* representation, the full set of low-dimensional MagPhase descriptors f_{0c}, M_c R_c and I_c are added to the feature set. The addition of the phase features in the *join* representation is because we expect that phase information would prevent some phase discontinuities between adjacent frames.

5.2.2 Experiments

5.2.2.1 Design

The experiments have two purposes. Firstly, we want to measure how the waveform generator works in a copy-synthesis task, that is by using features extracted from natural speech and then resynthesising by using the chosen units. We compare the proposed system against two standard vocoders, WORLD and MagPhase. Secondly, we measure the performance of the proposed system by synthesising from features generated by an SPSS system. To do so, we simluate the effect of the degradation applied by the SPSS system by following the technique used in Merritt et al. (2015). That is, features are

degraded by filtering with a sliding 5-frame Hann window. Then, the variance is scaled down to 60% (slight smoothing) or 80% (extreme smoothing). To limit the number of stimuli, we just peformed the mentioned degradation process to only two conditions, the proposed method and the MagPhase vocoder.

The system was built by using 2004 sentences uttered by a native male English speaker, recorded at 48kHz sample rate. In order to tune the system (weighting of feature streams), 19 other sentences were used. As a result, approximately 750k speech units were produced. The length of frames per units was set to 6, and the selected units were extended by 1 frame in both sides to allow spectral smoothing, as explained in Section 5.1.1.

5.2.2.2 Listening tests

We recruited 20 native English speakers, who evaluated 21 MUSHRA-like[1] screens, each. Listeners were asked to rate the quality of the stimuli on a scale from 0 (bad) to 100 (excellent). Natural speech was included as a hidden reference. This worked as a red flag to check for listeners who did not performed the test conscientiously, 3 subjects were rejected, since they rated natural speech less than 100% in more than one quarter of the screens. Details on the stimuli evaluated per screen are shown in Tables 5.1 and 5.2.

[1]Code available at `http://dx.doi.org/10.7488/ds/1316`

Table 5.1: Examplar-based Speech Waveform Generation: Systems under Test

Name	***Feature Extraction***	***Degradation***	***Waveform Generation***
WO-DN	WORLD	No	WORLD
MP-DN	MagPhase	No	MagPhase
EB-DN	MagPhase	No	Examplar-Based (proposed method)
MP-DS	MagPhase	slight	MagPhase
EB-DS	MagPhase	slight	Examplar-Based (proposed method)
MP-DE	MagPhase	extreme	MagPhase
EB-DE	MagPhase	extreme	Examplar-Based (proposed method)
Nat	Natural Speech		

TABLE 5.2: Feature Streams Used per Each Waveform Generator

Waveform Generation	***Feature Streams and Dimensions***
MagPhase	60 M_c + 1 f_{0c} + 45 R_c + 45 I_c
WORLD	60 MCEP + 5 BAP + 1 logF0
Examplar-Based (proposed method)	t_i: 60 M_c + 1 f_{0c} j_{i-1} : 60 M_c + 1 f_{0c} + 45 R_c + 45 I_c u_i : 2049 M + 1 f_0 + 2049 R + 2049 I (*)

* *Only retrieved after selection of units.*

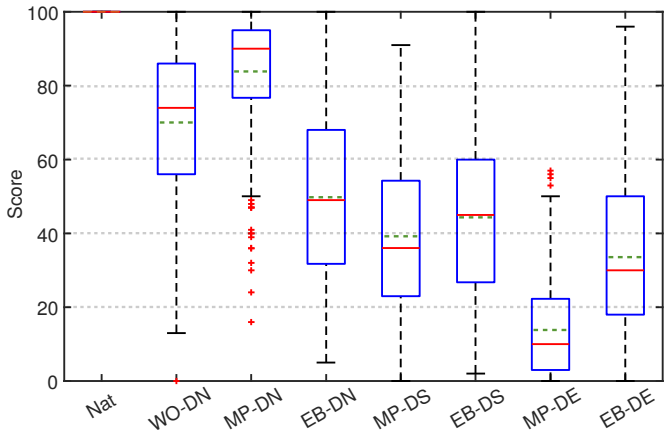

FIGURE 5.7: Absolute scores for the female voice. The green dotted line is the mean, and the continuous red line is the median.

5.2.2.3 Results

All systems performed significantly different to one another ($p < 0.05$). As shown in Figure 5.7, in the copy-synthesis task, the MagPhase vocoder obtained the highest mean score by far, followed by WORLD and the proposed system. However, when parameters are degraded, the MagPhase vocoder quality decreases significantly, while the proposed system is able to reduce the effect of the degradation more. As a result, for both degradations levels (slight and extreme), the proposed system performed the best.

5.2.2.4 Conclusions

A new method of waveform generation was proposed, which is based on the selection of small units. As a difference with typical unit selection methods, it does not require linguistic specification, instead unit boundaries are defined by epoch locations. Unlike vocoders, the proposed method is able to recover quality from degraded acoustic

parameters commonly generated by SPSS systems (statistical model). In terms of experimental results, the proposed method is overperformed by vocoders when working in a copy-synthesis task. However, the proposed method outperforms vocoders when the acoustic features are degraded before synthesis.

5.3 Speech Recognition

So far, the MagPhase vocoder has been used in text-to-speech synthesis (TTS), either in SPSS or unit selection systems. One of the main characteristics that makes MagPhase stand out from most other vocoders is its capability of extracting phase spectrum features, which have proved to be beneficial for improving the quality in TTS.

It is believed that humans distinguish phonemes and voices mostly based on the vocal tract spectral shape (location of formants), while discarding phase information (Jurafsky and Martin, 2009, p. 329-336). Accordingly, magnitude spectrum-derived features, such as Mel-Frequency Cepstral Coefficients (MFCCs) and Log Mel-Filter Bank Outputs (FBANK) are typically used. Some tonal features are used (F0), but only for tonal languages.

Listening tests carried out by Liu et al. (1997) show that phase information is not useful for intelligibility of speech. Moreover, the wrapping of phase spectrum derived from the Fourier Transform is problematic to acoustic modelling (Mowlaee et al., 2016). Accordingly, phase information is usually discarded in Automatic Speech Recognition (ASR). However, recent research highlights that phase may be a useful acoustic property for ASR (Mowlaee et al., 2016).

Hence, we want to evaluate and compare the use of MagPhase features in the task of ASR, in which usually only magnitude spectra-derived features are used, such as MFCCs, etc. We believe that the phase-derived features can make a difference, since they would give essential clues to the recognition systems.

Bacon (2018) recently used MagPhase features for ASR, with special focus on the MagPhase phase-derived speech features. An ASR system is device that transcribes text from speech. To do so, it usually extracts acoustic features from speech, then it evaluates these against an acoustic model, and finally computes the most probable sequence of words taking into account also a language model and lexicon. Most of these components are usually merged into a Weighted Finite State Transducer (WFST). For further details, please refer to Mohri et al. (2002).

5.3.1 Experiments

This section is a summary of the experiments carried out by Bacon (2018) on the application of MagPhase features in ASR.

5.3.1.1 Design

The Kaldi toolkit (Povey et al., 2011) was used for the experiments. A neural network-based recipe[2] for the LDC Wall Street Journal (WSJ) was used as the basic script to run the experiments.

Three types of acoustic features were used for the experiments: 40 FBANKs (baseline) extracted using the WSJ recipe in Kaldi, 40 MagPhase M_c, and MagPhase 10 R_c, which are evaluated in different setups. The MagPhase feature I_c is not used, since its inclusion in speech synthesis is for disambiguation and feature completeness, which should be not required in ASR.

Acoustic features are preprocessed by linear discriminant analysis (LDA) before going at the input of a feed-forward neural network. The LDA processed features are fed into the first block of 3 layers of the network. Then, 3 normal feed-forward layers were added. Each hidden layer was made up of 1,536 units.

The LDC Wall Street Journal (WSJ) corpus was used for the experiments. It contains excerpts read by multiple speakers and recorded at 16kHz sample rate. 28,928, 333, 289 utterances were used for training, development, and test, respectively.

Typically in ASR, the Word Error Rate (WER) is used as an objective measure of performance:

$$\text{WER} = \frac{S + I + D}{N} \tag{5.5}$$

Where S, I, and D are the number of substitutions, insertions, and deletions, respectively. N is the total number of words in the expected transcription.

Two types of ASR setups were tested, one *simple* in which one network was used to evaluate the systems under test, and *Combined*, which by using a "voting" system, several different systems can be combined to improve the results (Goel et al., 2000).

5.3.1.2 Results

For the *Simple* systems, we observe in Table 5.3 that the best performance achieved in the test set is achieved by the baseline (FBANK) (11.28%), although the system Base+R_c is very close (11.55%). The latter outperformed all other systems on the development set (7.43%).

The worst results were obtained by the R_c feature by its own. Even though it just describes phase information, it is able to retrieve *some* phonetic information.

[2] `https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/local/nnet2/run_5b_gpu.sh`

TABLE 5.3: Word Error Rates per Feature Type

System	***Dev***	***Test***	***Type***
Base	7.48	**11.28**	Simple
M_c	7.99	13.99	
R_c	30.5	60.02	
Base+R_c	**7.43**	11.55	
M_c+R_c	7.92	13.96	
(Base) ∘ (R_c)	8.20	13.20	Combined
(Base) ∘ (R_c) ∘ (Base+R_c)	7.50	11.19	
(Base) ∘ (R_c) ∘ (Base+R_c) ∘ (M_c+R_c)	7.28	11.67	
(Base) ∘ (R_c) ∘ (Base+R_c) ∘ (M_c+R_c) ∘ (M_c)	7.32	12.45	
(Base+R_c) ∘ (Base)	7.46	11.19	
(Base+R_c) ∘ (Base) ∘ (M_c+R_c)	**7.27**	**11.14**	
(Base+R_c) ∘ (Base) ∘ (M_c+R_c) ∘ (M_c)	7.46	11.64	

+: Feature concatenation. **∘:** System combination.

Also, the magnitude representation FBANK (baseline) outperforms its MagPhase counterpart feature M_c by about 0.5%.

For the *Combined* systems, Table 5.3 shows that the combined system (Base+R_c) ∘ (Base) ∘ (M_c+R_c) outperforms all other systems on test and development sets.

5.3.2 Discussion

The inclusion of phase-derived features, in this case the R_c MagPhase descriptor tends to improve the accuracy of the system. However, this reduction in WER was slight and not consistent across the experiments.

It is also worth noting that the phase-derived feature R_c by its own is able to make the system recognise some phonetic characteristics, even though it only carries phase information and only during voiced speech.

It is expected that the baseline feature FBANK outperforms the MagPhase feature M_c, since the former is extracted from a much longer frame length, and therefore the resulting spectrum is more stable than over two pitch periods as with M_c.

5.4 Conclusion

The MagPhase vocoder has shown its capabilities not only on statistical parametric speech synthesis, but also in unit selection and hybrid systems. Not only that, it also has been used in automatic speech recognition with promising results.

It works as a good concatenation and join smoothing system for unit selection-based speech synthesis, which was an important component of the CSTR entries to the Blizzard Challenge of 2017 and 2018. The submitted systems performed adequately being positioned among the best systems in the challenges.

It can be used as an essential component of other vocoder/waveform generation systems showing outstanding results.

Finally, The MagPhase feature extractor was studied in the ASR framework. Experiments were carried out showing not conclusive, but promising results for MagPhase phase-derived features.

Chapter 6

Conclusions

6.1 Summary

Several paradigms of text-to-speech systems have been proposed so far. Some of them are used in production, thanks to their small footprint, controllability, robustness, and the high quality speech that they are capable to generate. One of the most interesting paradigms in TTS is Statistical Paramteric Speech Synthesis (SPSS), which achieves high speech quality and high scores of intelligibility. One of the critical blocks in SPSS is the vocoder, which is the device that performs feature extraction, and also the final waveform generation. It has been pointed out as a high source of artefacts in speech quality, which are commonly termed "buzziness" or "phasiness". As a pure signal processing-based device, its accuracy relies not only in the method itself, but on the accuracy of the implementation. No doubt, the improvements of such devices and therefore the final synthesised speech is a fairly challenging task.

In Chapter 1, we provide a summary of current TTS methods, to contextualise the current state of the speech synthesis task. Then, we list and describe current state-of-the-art vocoders emphasising details on their feature extraction methods, acoustic feature characteristics and how they compare to each other. Consequently, we think that the problems in vocoders are:

- In spite of the recent developments on vocoding, there is still room for improvement.
- Vocoders apply extreme decomposition to the structures of speech.
- Dependency between stochastic and deterministic processes of speech production has not been modelled.

- Many processes of speech production are not well understood, and so are approached by simplistic inaccurate models.
- Harmonic-based models sound better than methods that perform source-filter separation.
- The most complex methods perform worse than the ones that use simple and robust excitation modelling.

Taking into account these observations, we proposed ways to address them.

In Chapter 2 we focused our research on the waveform generation block with the main goals: simplify signal processing, reduce estimation steps, and avoid unnecessary decomposition of speech. Usually, during waveform generation, the acoustic features at the input are used as parameters for the raw signal synthesis. E.g., The fundamental frequency gives the rate at which artificial pulses are generated. Instead, in our proposed method, a waveform generator based on signal reshaping, the input features are *targets* for natural speech. Hence, we have proposed a new paradigm for waveform generation, which is opposite to typical methods: it does not intend to decompose waveforms, instead it reshapes natural speech using filtering and pitch manipulations.

Ideally, we could have an infinitely large speech database, from which synthesise any excerpt with no signal modification. Nevertheless, as this scenario is not possible, we would like to use natural speech without modifications to the extent possible. We think that by avoiding unnecessary modifications, we can preserve the natural quality of speech. In terms of signal processing, we opted for the most simple, but robust signal processing techniques: time-domain convolution for filtering, and resampling for F0 modification.

The proposed waveform generator, "Signal Reshaping", was subjectively tested in a SPSS task against a strong baseline, STRAIGHT. The proposed method outperformed the baseline, showing that by simplifying signal processing and using natural speech as the source of generation, the resulting speech quality is improved. Perceptually, we noticed a reduction in "buzziness". Furthermore, an interesting result is that the proposed method does not require any aperiodicity parameter, e.g., when doing copy-synthesis, the synthesised speech keeps the characteristics of the speaker and the original utterance, producing an almost identical copy of the signal. Thus, with the development of the "Signal Reshaping" waveform generator, we started a completely new class of waveform generators that does not fit into harmonic-based models or source-filter separation, and achieves superior sound quality in SPSS.

In spite of the good results of our proposed method, some noticeable artefacts are still produced, which are perceived as "phasiness". This is because the "reshaping" filter is applied on top of the natural vocal tract filter, therefore the signal is double filtered making the tails of the impulse response sufficiently long to be perceived as unnatural.

Some of the findings from the research on signal reshaping are:

- Removing unnecessary decomposition in speech modelling is beneficial to achieve higher quality.
- Even though natural speech was used for waveform generation, speech quality was not dramatically improved.
- Because the waveform generator relies on conventional speech features derived from STRAIGHT, its perceived sound quality resembles STRAIGHT.
- Conventional speech features are suboptimal.

These suggest that to improve speech quality in SPSS, it is necessary to propose not only a new waveform generator, but a whole vocoder involving feature extraction to extract optimal acoustic features, and a new waveform generator to achieve higher speech quality (Chapter 3). To a certain extent, vocoders are defined by the acoustic features that they can extract or use to synthesise from, thus the search for those features is critical.

The goals for the new vocoder were:

- Avoid estimation steps to the greatest extent possible.
- Extract features that are consistent so they can be used for statistical modelling (e.g., they can be safely averaged).
- Get rid of "phasiness" and "buzziness" of typical vocoders.
- Use a real-valued deep learning architecture typically employed in SPSS.

In order to accomplish these goals and by following the trend of working closer to the signal itself, we decided to use the Fourier Transform as a perfect lossless representation of the waveform. Basically, the MagPhase vocoder is a sensible way of encoding the complex FFT coefficients derived from the Short-Time Fourier Transform (STFT). With these robust and consistent encoded features, it is possible to achieve high quality synthesised speech in a SPSS system. Subjective tests were carried out and the

results showed that the proposed method remarkably outperformed the state-of-the-art vocoder, STRAIGHT, for a female and a male voice. It gets rid off the "buzziness" and "phasiness" typical of vocoders, delivering a more natural sound. Moreover, as it works in a pitch-synchronous fashion, it works at a lower frame rate than the conventional 5ms, on average (reaching an impressive reduction of 31.5%). Another advantage of the vocoder is its simple design, which makes it perfect for real time applications. The results also make clear that the inclusion of phase-derived features in SPSS is beneficial and contributes to the naturalness of synthesised speech.

After presenting the basic design, we described several modifications to the MagPhase vocoder in Chapter 4. We realised that the number of features used by our method was noticeable higher than for other vocoders. That might be a cause of degradation in the training/inference performed by the acoustic model. So, we decreased the number of coefficients describing the phase-derived features.

As the MagPhase vocoder works at a variable frame rate, some methods need to be used to make it work with a standard SPSS toolkit, such as Merlin. For the original design, we needed to modify Merlin's code, so it could accept variable frame rate vocoders. However, it was a complex implementation and hard to extend to other SPSS toolkits (e.g., HTS). Hence, we implemented other methods for the variable frame rate handling, which are described in Chapter 4.

The proposed vocoder, as a reliable representation of FFT spectra, can be used for other applications in speech processing (Chapter 5). It was used in a hybrid speech synthesiser to concatenate and smooth the joins between speech units. Also, it was used as a part of an exemplar-based waveform generator. The MagPhase vocoder was essential for the functioning of both systems. Finally, an investigation was presented showing the benefits of using MagPhase speech features in automatic speech recognition, showing promising results.

6.2 Future Work

From the work on Waveform Generation based on Signal Reshaping, we learned how to manipulate natural speech signals whilst reducing artefacts. Join smoothing in concatenation-based systems is one of the potential applications of the methods implemented and learned during this research. E.g., this can be applied to the systems described in Sections 5.1 and 5.2.

The MagPhase vocoder has proved to be a very effective speech processing device in several contexts. Still, there is enough room to investigate and improve the delivered speech quality. Among these are:

- *Voiced/unvoiced decision removal:* We think that hard decisions affect the resulting speech quality. Voicing decisions generate artefacts especially when synthesising partially voiced phonemes (e.g., /Z/) and in the switch between voiced and unvoiced speech. Ideally, the voiced/unvoiced decision should be replaced by a soft voicing descriptor.

- *Maximum voiced frequency removal:* The maximum voiced frequency (MVF) is another hard decision factor involved in the feature extraction and synthesis. For the current implementation, its value is fixed by the user. We think that further research can be carried out to remove this value and use a more sensible descriptor.

- *Speech Styles:* So far, the MagPhase vocoder has been formally tested on *normal* speaking style, but we would like to study how the vocoder behaves with other speaking styles and voice characteristics, such as: creaky voice, lombard, whispering, etc. For instance, we think that some major changes to the vocoder need to be done for creaky voice, especially in the phase-derived feature extraction.

- *Use of lossless features in SPSS:* Due to the high dimensionality of the lossless MagPhase features, we have not used them directly into SPSS. It would be interesting to investigate the speech quality that they may achieve. To make the synthesiser work properly, we expect that it would be necessary to change the loss function commonly used in SPSS (RMSE).

- *Other ways to compress phase features:* Mel-scale frequency warping is used to reduce the dimensionality of spectral features, and showing satisfactory results. However, we have not studied the effect of the Mel-scale frequency warping of the phase-derived features in the waveform domain. Perhaps there are more optimal ways to lower the dimensionality of phase features, in the waveform domain.

6.3 Final Remarks

In spite of the rise of the so-called "neural vocoders", classic signal processing still remains a feasible paradigm for a series of applications in speech processing. Even though we did not cover the neural vocoders in this work, we do believe that pure signal processing vocoders can be very efficient and deliver high quality synthetic speech. Furthermore, they provide controllability, which usually is not fully supported by neural vocoders.

Even though vocoders are critical for the speech quality achieved by SPSS systems, we believe that the acoustic model (i.e., neural network) is a very important factor in the degradation of speech quality; it seems to impose an upper bound on quality. We think that along with improving vocoders, acoustic modelling should be improved simultaneously to work jointly.

In this work we published several peer-reviewed articles (See Section 1.1). Also, we implemented fast and efficient software, which is freely available and open source, written in MATLAB and Python from scratch. To the best of our knowledge, this software has been used in several educational institutions and companies.

Appendix A

Instructions given to the Listeners During Subjective Tests

Listener Instructions

You will listen to a number of speech samples and rate their:

NATURALNESS

on a scale from 0 (bad) to 100 (excellent).

The test interface is pictured below. You press any of the Play buttons to listen to a recording and use the corresponding slider to indicate its quality.

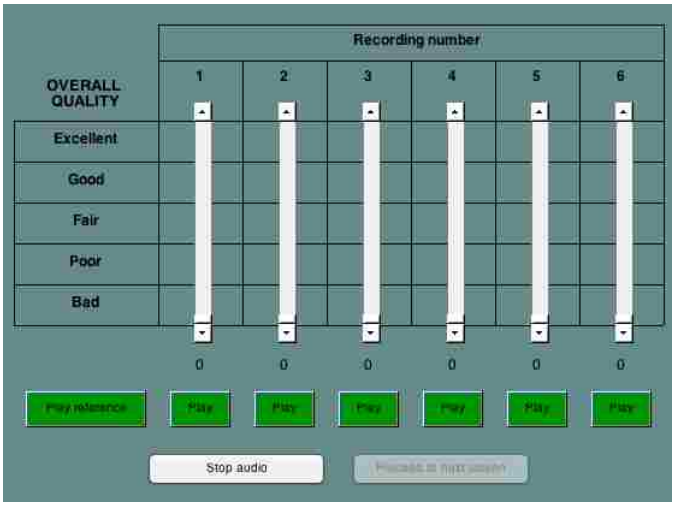

FIGURE A.1: Test interface example.

As a reference point, the Play reference button plays a high quality recording of completely natural speech (sometimes of a different sentence). The same natural speech

recording is also associated with a randomly chosen Play button; it should always be rated at 100.

You can listen to the different recordings in any order as many times as you like. Use this to revise your ratings until you are satisfied. Be careful, as the differences between recordings can be quite subtle.

Once you are confident that your ratings are appropriate, press the button to proceed to the next screen.

There are 30 screens as a total. Before evaluating each of the the voices you will be given one training screen to familiarise yourself with the task and the samples.

Please take this task seriously.

Bibliography

Airaksinen, M., Juvela, L., Bollepalli, B., Yamagishi, J., and Alku, P. (2018). A comparison between straight, glottal, and sinusoidal vocoding in statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1658–1670.

Arik, S. Ö., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Raiman, J., Sengupta, S., and Shoeybi, M. (2017). Deep voice: Real-time neural text-to-speech. *CoRR*, abs/1702.07825.

Babacan, O., Drugman, T., Raitio, T., Erro, D., and Dutoit, T. (2014). Parametric representation for singing voice synthesis: A comparative evaluation. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 2564–2568, Florence, Italy.

Bacon, H. (2018). Using phase-based features in addition to magnitude-based features for reducing wer in asr. Master's dissertation, The University of Edinburgh.

Banos, E., Erro, D., Bonafonte, A., and Moreno, A. (2008). Flexible harmonic/stochastic modeling for HMM-based speech synthesis. In *In V Jornadas en Tecnologias del Habla*, pages 145–148, Bilbao, Spain.

Barney, A., De Stefano, A., and Henrich, N. (2007). The effect of glottal opening on the acoustic response of the vocal tract. *Acta Acustica united with Acustica*, 93(6):1046–1056.

Blum, J. R. (1954). Multidimensional stochastic approximation methods. *Ann. Math. Statist.*, 25(4):737–744.

Boersma, P. (1993). Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In *IFA Proceedings 17*, pages 97–110.

Bollepalli, B., Urbain, J., Raitio, T., Gustafson, J., and Cakmak, H. (2014). A comparative evaluation of vocoding techniques for HMM-based laughter synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 255–259, Florence, Italy.

Csapó, T. G. and Németh, G. (2013). A novel irregular voice model for hmm-based speech synthesis. In *8th ISCA Workshop on Speech Synthesis*, pages 249–254, Barcelona, Spain.

Degottex, G. and Erro, D. (2014). A uniform phase representation for the harmonic model in speech synthesis applications. *EURASIP, Journal on Audio, Speech, and Music Processing - Special Issue: Models of Speech - In Search of Better Representations*, 2014(1):38.

Degottex, G., Lanchantin, P., and Gales, M. (2018). A log domain pulse model for parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):57–70.

Degottex, G. and Stylianou, Y. (2013). Analysis and synthesis of speech using an adaptive full-band harmonic model. *IEEE Transactions on Acoustics, Speech and Language Processing*, 21(10):2085–2095.

Drugman, T., Bozkurt, B., and Dutoit, T. (2012). A comparative study of glottal source estimation techniques. *Computer Speech & Language*, 26(1):20 – 34.

Drugman, T. and Dutoit, T. (2012). The deterministic plus stochastic model of the residual signal and its applications. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(3):968–981.

Drugman, T., Moinet, A., Dutoit, T., and Wilfart, G. (2009). Using a pitch-synchronous residual codebook for hybrid HMM/frame selection speech synthesis. In *Proc. ICASSP*, pages 3793–3796, Taipei, Taiwan.

Drugman, T. and Raitio, T. (2014). Excitation modeling for HMM-based speech synthesis: Breaking down the impact of periodic and aperiodic components. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 260–264, Florence, Italy.

Erro, D., Sainz, I., Navas, E., and Hernaez, I. (2014). Harmonics plus noise model based vocoder for statistical parametric speech synthesis. *Selected Topics in Signal Processing, IEEE Journal of*, 8(2):184–194.

Erro, D., Sainz, I., Navas, E., and Herndez, I. (2011). HNM-based MFCC+F0 extractor applied to statistical speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 4728–4731.

Espic, F., Govender, A., Ribeiro, M. S., Valentini-Botinhao, C., and Watts, O. (2018). The CSTR entry to the Blizzard Challenge 2018. In *Proceedings Blizzard Challenge Workshop*, Hyderabad, India.

Goel, V., Kumar, S., and Byrne, W. (2000). Segmental minimum Bayes-risk ASR voting strategies. In *Proc. of the International Conference on Spoken Language Processing*, volume 3, pages 139–142 (4 pages), Beijing, China.

Gopinath, R. A. (1998). Maximum likelihood modeling with gaussian distributions for classification. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 2, pages 661–664 vol.2.

Griffin, D. and Lim, J. (1984). Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243.

Hanna, N., Smith, J., and Wolfe, J. (2012). Low frequency response of the vocal tract: acoustic and mechanical resonances and their losses. In *Proc. Acoustics*.

Heinzel, G., Rdiger, A., and Schilling, R. (2002). Spectrum and spectral density estimation by the discrete fourier transform (dft), including a comprehensive list of window functions and some new flat-top windows. Technical report, Max Plank Institut.

Hemptinne, C. (2006). *Integration of the Harmonic plus Noise Model (HNM) into the Hidden Markov Model-Based Speech Synthesis System (HTS)*. MSc dissertation - IDIAP Research Institute, Martigny, Switzerland.

Henter, G. E., Merritt, T., Shannon, M., Mayo, C., and King, S. (2014). Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech. In *Proc. Interspeech*, volume 15, pages 1504–1508.

Hirai, T., Yamagishi, J., and Tenpaku, S. (2007). Utilization of an HMM-based feature generation module in 5 ms segment concatenative speech synthesis. In *Proc. 6th ISCA Workshop on Speech Synthesis (SSW-6)*.

Hu, Q., Richmond, K., Yamagishi, J., and Latorre, J. (2013). An experimental comparison of multiple vocoder types. In *8th ISCA Workshop on Speech Synthesis*, pages 155–160, Barcelona, Spain.

Hu, Q., Stylianou, Y., Maia, R., Richmond, K., Yamagishi, J., and Latorre, J. (2014). An investigation of the application of dynamic sinusoidal models to statistical parametric speech synthesis. In *Proc. Interspeech*, pages 780–784, Singapore.

Hu, Q., Yamagishi, J., Richmond, K., Subramanian, K., and Stylianou, Y. (2016). Initial investigation of speech synthesis based on complex-valued neural networks. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5630–5634.

Hunt, A. and Black, A. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 1, pages 373–376 vol. 1, Atlanta, Georgia, USA.

Imai, Sumita, and Furuichi (1983). Mel log spectrum approximation (mlsa) filter for speech synthesis. *Electronics and Communications in Japan (Part I: Communications)*, 66(2):10–18.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., van den Oord, A., Dieleman, S., and Kavukcuoglu, K. (2018). Efficient neural audio synthesis. *CoRR*, abs/1802.08435.

Karaali, O., Corrigan, G., and Gerson, I. (1996). Speech synthesis with neural networks. *World Congress on Neural Networks*.

Kawahara, H., Estill, J., and Fujimura, O. (2001). Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT. In *MAVEBA*, pages 59–64. ISCA.

Kawahara, H., Katayose, H., de Cheveigné, A., and Patterson, R. D. (1999a). Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of F0 and periodicity. In *Sixth European Conference on Speech Communication and Technology, EUROSPEECH 1999, Budapest, Hungary, September 5-9, 1999*.

Kawahara, H., Masuda-Katsuse, I., and de Cheveign, A. (1999b). Restructuring speech representations using a pitch-adaptive timefrequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, 27(34):187 – 207.

King, S., Crumlish, J., Martin, A., and Wihlborg, L. (2018). The Blizzard Challenge 2018. In *Proceedings Blizzard Challenge Workshop*, Hyderabad, India.

King, S. and Karaiskos, V. (2012). The Blizzard Challenge 2012. In *Proceedings Blizzard Workshop 2012*, Portland, OR, USA.

King, S., Wihlborg, L., and Guo, W. (2017). The Blizzard Challenge 2017. In *Proceedings Blizzard Challenge Workshop*, Stockholm, Sweden.

Klimkov, V., Moinet, A., Nadolski, A., and Drugman, T. (2018). Parameter generation algorithms for text-to-speech synthesis with recurrent neural networks. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 626–631.

Koishida, K., Tokuda, K., Kobayashi, T., and Imai, S. (1995). Celp coding based on mel-cepstral analysis. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 33–36 vol.1.

Laroche, J., Stylianou, Y., and Moulines, E. (1993). HNS: Speech modification based on a harmonic+noise model. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, pages 550–553 vol.2.

Levinson, S. (1986). Continuously variable duration hidden markov models for speech analysis. In *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pages 1241–1244.

Ling, Z.-H. and Zhou, Z.-P. (2018). Unit selection speech synthesis using frame-sized speech segments and neural network based acoustic models. *Journal of Signal Processing Systems*, 90(7):1053–1062.

Liu, L., He, J., and Palm, G. (1997). Effects of phase on the perception of intervocalic stop consonants. *Speech Communication*, 22(4):403 – 417.

Maia, R., Akamine, M., and Gales, M. J. (2013). Complex cepstrum for statistical parametric speech synthesis. *Speech Communication*, 55(5):606 – 618.

Maia, R., Toda, T., Zen, H., Nankaku, Y., and Tokuda, K. (2007). An excitation model for HMM-based speech synthesis based on residual modeling. In *Proc. SSW*, pages 131–136, Bonn , Germany.

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A. C., and Bengio, Y. (2016). Samplernn: An unconditional end-to-end neural audio generation model. *CoRR*, abs/1612.07837.

Merritt, T., Latorre, J., and King, S. (2015). Attributing modelling errors in HMM synthesis by stepping gradually from natural to modelled speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4220–4224, Brisbane.

Merritt, T., Raitio, T., and King, S. (2014). Investigating source and filter contributions, and their interaction, to statistical parametric speech synthesis. In *Proc. Interspeech*, pages 1509–1513, Singapore.

Mohri, M., Pereira, F., and Riley, M. (2002). Weighted finite-state transducers in speech recognition. *Comput. Speech Lang.*, 16(1):69–88.

Mowlaee, P., Saeidi, R., and Stylianou, Y. (2016). Advances in phase-aware signal processing in speech communication. *Speech Communication*, 81:1 – 29. Phase-Aware Signal Processing in Speech Communication.

Murthy, H. A. and Gadde, V. (2003). The modified group delay function and its application to phoneme recognition. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 1, pages I–68–71 vol.1.

Pantazis, Y., Rosec, O., and Stylianou, Y. (2010). Iterative estimation of sinusoidal signal parameters. *IEEE Signal Processing Letters*, 17(5):461–464.

Ping, W., Peng, K., and Chen, J. (2019). Clarinet: Parallel wave generation in end-to-end text-to-speech. In *International Conference on Learning Representations*.

Pollet, V. and Breen, A. (2008). Synthesis by generation and concatenation of multiform segments. In *Proc. Interspeech*, pages 1825–1828, Brisbane, Australia.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Prenger, R., Valle, R., and Catanzaro, B. (2019). Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621.

Qian, Y. and Soong, F. (2012). A unified trajectory tiling approach to high quality tts and cross-lingual voice transformation. In *Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on*, pages 165–169, Kowloon Tong, China.

Qian, Y., Xu, J., and Soong, F. K. (2011). A frame mapping based hmm approach to cross-lingual voice transformation. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5120–5123.

Qian, Y., Yan, Z.-J., Wu, Y.-J., Soong, F. K., Zhang, G., and Wang, L. (2010). An HMM trajectory tiling (HTT) approach to high quality TTS - Microsoft Entry to Blizzard Challenge 2010. In *Blizzard Challenge 2010*, Kansai Science City, Japan. ISCA, ISCA.

Raitio, T., Lu, H., Kane, J., Suni, A., Vainio, M., King, S., and Alku, P. (2014a). Voice source modelling using deep neural networks for statistical parametric speech synthesis. In *22nd European Signal Processing Conference (EUSIPCO)*, Lisbon, Portugal.

Raitio, T., Suni, A., Juvela, L., Vainio, M., and Alku, P. (2014b). Deep neural network based trainable voice source model for synthesis of speech with varying vocal effort. In *Proc. of Interspeech*, pages 1969–1973, Singapore.

Raitio, T., Suni, A., Pulakka, H., Vainio, M., and Alku, P. (2008). Hmm-based finnish text-to-speech system utilizing glottal inverse filtering. In *INTERSPEECH*, pages 1881–1884. ISCA.

Raitio, T., Suni, A., Yamagishi, J., Pulakka, H., Nurminen, J., Vainio, M., and Alku, P. (2011). HMM-based speech synthesis utilizing glottal inverse filtering. *IEEE Trans. on Audio, Speech and Language Processing*, 19(1):153–165.

Rodet, X. (1997). Musical Sound Signal Analysis/Synthesis. In *Proceedings of the IEEE Time-Frequency and Time-Scale Workshop (TFTS'97*, pages 1–1, Coventry, United Kingdom. cote interne IRCAM: Rodet97e.

Ronanki, S., Ribeiro, S., Espic, F., and Watts, O. (2017). The CSTR entry to the Blizzard Challenge 2017. In *Proceedings Blizzard Challenge Workshop*, Stockholm, Sweden.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R. J., Saurous, R. A., Agiomyrgiannakis, Y., and Wu, Y. (2017). Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. *CoRR*, abs/1712.05884.

Stylianou, Y. (2001). Applying the harmonic plus noise model in concatenative speech synthesis. *Speech and Audio Processing, IEEE Transactions on*, 9(1):21–29.

Stylianou, Y., Laroche, J., and Moulines, E. (1995). High-quality speech modification based on a harmonic + noise model. In *Fourth European Conference on Speech Communication and Technology, EUROSPEECH 1995, Madrid, Spain, September 18-21, 1995*.

Suni, A., Raitio, T., Vainio, M., and Alku, P. (2010). The GlottHMM speech synthesis entry for Blizzard Challenge 2010. In *Blizzard Challenge 2010 Workshop*, Kyoto, Japan.

Takaki, S., Kameoka, H., and Yamagishi, J. (2017). Direct modeling of frequency spectra and waveform generation based on phase recovery for dnn-based speech synthesis. In *Proc. Interspeech 2017*, pages 1128–1132.

Tamamori, A., Hayashi, T., Kobayashi, K., Takeda, K., and Toda, T. (2017). Speaker-dependent wavenet vocoder. In *Proc. Interspeech 2017*, pages 1118–1122.

Taylor, P. (2009). *Text-to-Speech Synthesis*. Cambridge University Press, Cambridge, U.K.

Titze, I. R. (2008). Nonlinear source-filter coupling in phonation: Theory. *Journal of the Acoustical Society of America*, 123(5):2733 – 2749.

Toda, T., Black, A. W., and Tokuda, K. (2008). Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model. *Speech Communication*, 50(3):215 – 227.

Toda, T. and Tokuda, K. (2005). Speech parameter generation algorithm considering global variance for HMM-based speech synthesis. In *Proceedings of Eurospeech, Interspeech 2005*, pages 2801 – 2804, Lisbon, Portugal.

Tokuda, K., Kobayashi, T., and Imai, S. (1995a). Adaptive cepstral analysis of speech. *IEEE Trans. on Speech and Audio Processing*, SA-3(6):481–489.

Tokuda, K., Kobayashi, T., and Imai, S. (1995b). Speech parameter generation from HMM using dynamic features. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 660–663 vol.1.

Tokuda, K. and Zen, H. (2016). Directly modeling voiced and unvoiced components in speech waveforms by neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5640–5644.

Valentini-Botinhao, C., Watts, O., Espic, F., and King, S. (2018). Examplar-based speech waveform generation for text-to-speech. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. Accepted.

Valentini-Botinhao, C., Wu, Z., and King, S. (2015). Towards minimum perceptual error training for DNN-based speech synthesis. In *Proc. Interspeech*, Dresden, Germany.

Valin, J. and Skoglund, J. (2019). Lpcnet: Improving neural speech synthesis through linear prediction. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5891–5895.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499.

van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L. C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., and Hassabis, D. (2017). Parallel wavenet: Fast high-fidelity speech synthesis. *CoRR*, abs/1711.10433.

Volkmann, J., Stevens, S. S., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):208–208.

Wang, X., Takaki, S., and Yamagishi, J. (2017). An autoregressive recurrent mixture density network for parametric speech synthesis. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4895–4899.

Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q. V., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. (2017). Tacotron: A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135.

Watts, O., Valentini-Botinhao, C., Espic, F., and King, S. (2018). Exemplar-based speech waveform generation. In *Interspeech*.

Wu, Y.-J. and Wang, R.-H. (2006). Minimum generation error training for hmm-based speech synthesis. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I.

Wu, Z. and King, S. (2016). Investigating gated recurrent networks for speech synthesis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5140–5144.

Wu, Z., Watts, O., and King, S. (2016). Merlin: An open source neural network speech synthesis system. In *Proc. 9th ISCA Speech Synthesis Workshop (SSW9)*, Sunnyvale, CA, USA.

Yamagishi, J., Nose, T., Zen, H., Ling, Z., Toda, T., Tokuda, K., King, S., and Renals, S. (2009). Robust speaker-adaptive hmm-based text-to-speech synthesis. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1208–1230.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proc. Eurospeech*, pages 2347–2350, Budapest, Hungary.

Yoshimura, T., Tokuda, K., Masukom, T., Kobayashi, T., and Kitamura, T. (2001). Mixed excitation for HMM-based speech synthesis. In *Proc. Eurospeech*, pages 2263–2267, Aalborg, Denmark.

Young, S. J., Odell, J. J., and Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 307–312, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yu, K. and Young, S. (2011). Continuous f0 modeling for hmm based statistical parametric speech synthesis. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1071–1079.

Zen, H., Nose, T., Yamagishi, J., Sako, S., Masuko, T., Black, A. W., and Tokuda, K. (2007). The HMM-based speech synthesis system (HTS) version 2.0. In *SSW6-2007*, pages 294–299.

Zen, H. and Sak, H. (2015). Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4470–4474.

Zen, H., Senior, A., and Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. In *Proc. ICASSP*, pages 7962–7966, Vancouver, BC, Canada.

Zen, H., Toda, T., and Tokuda, K. (2008). The nitech-naist hmm-based speech synthesis system for the blizzard challenge 2006. *IEICE Transactions on Information and Systems*, E91.D(6):1764–1773.

Zen, H., Tokuda, K., and Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039 – 1064.