Adaptive Time-segmentation for Packet Loss Channels

Rødbro, Christoffer Asgaard; Jensen, Jesper; Heusdens, Richard

Publication date:
2005

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Rødbro, C. A., Jensen, J., & Heusdens, R. (2005). Adaptive Time-segmentation for Packet Loss Channels. (pp. 21).

Adaptive Time-segmentation for Packet Loss Channels

Reduced Version

Christoffer A. Rødbro[†], Jesper Jensen[‡] and Richard Heusdens[‡]

[†]: Department of Communication Technology, Aalborg University, Denmark

[‡]: Department of Mediamatics, Delft University of Technology, The Netherlands

17th March 2005

Technical Report
Department of Mediamatics
Delft University of Technology

Preface

This document is a reduced version of a technical report describing an adaptive time-segmentation strategy for use in real-time voice communications with frame losses, such as voice over IP (VoIP). The copyrights for parts of this work was handed over to the IEEE in conjunction with the following two papers:

C. A. Rødbro, J. Jensen, and R. Heusdens,
Adaptive Time-segmentation for Speech Coding with Limited Delay,
Proc. IEEE ICASSP, vol. 1, 465–468,
2004.

and

C. A. Rødbro, J. Jensen, and R. Heusdens,
Rate-Distortion Optimal Time-segmentation and Redundancy Selection for VoIP,
Submitted to: IEEE Transactions on Speech and Audio Processing,
June 2004.

The document at hand is the result of removing the copyrighted material from the original technical report and serves as a reference for information that could not be included in the papers due to space limitations. The remaining content is:

1. A frame-independent harmonic sinusoidal coder upon which the the above-mentioned papers are based.

2. A receiving end PLC algorithm employed in the latter of the two papers.

Contents

Chapter 1

Harmonic Sinusoidal Speech Coder

Because of the possibility of packet losses a speech coding algorithm for VoIP should produce self-contained frames. By this we mean that no information from previous (or future) frames should be necessary in order to decode the current. This limitation rules out the use of inter-frame differential parameters. In the following we will describe a harmonic sinusoidal coding scheme that fulfills this requirement.

1.1 Signal Model

In a harmonic sinusoidal model a frame of speech (denoted by the vector $\mathbf{s}$) is represented by a weighted sum of harmonically related sinusoids:

$$\hat{\mathbf{s}} = \sum_{k=1}^{K} A_k \cos\left(k\omega_0 \mathbf{t} + \phi_k\right) \tag{1.1}$$

Here we used MATLAB-style notation, i.e. $\cos(\cdot)$ is applied elementwise and the + denotes ϕ_k being added to each component of the vector $k\omega_0\mathbf{t}$. Moreover,

- ω_0 is the fundamental frequency in radians
- $\mathbf{t}$ is a time index vector, $\mathbf{t} = [0, 1, \ldots, L-1]^T$ where L is the frame length and $(\cdot)^T$ denotes the transpose.
- A_k is the amplitude of the k'th component
- ϕ_k is the initial phase of the k'th component
- K is the number of components (determined by the fundamental frequency)

This model is physiologically founded for strictly voiced speech only, however we will later show how - with some modifications - it can be applied to unvoiced speech as well.

1.2 Fundamental frequency estimation

The fundamental frequency ω_0 is estimated based on the YIN algorithm proposed in [dCK02]. The algorithm has been modified here in order to increase robustness towards halvings and multiples. This is achieved by considering local minima of the YIN cost-function $C(P_0)$, examples of which are shown in Figure 1.1. The fundamental frequency period P_0 is chosen as the time-lag minimizing the cost-function that is a measure of the unvoiced-to-total power ratio [dCK02]. In the left-hand plot, the global minimum is significantly lower than the second lowest local minimum and the estimate is therefore classified as being "confident". On the right-hand side, two local minima attain almost the same value, and thus the estimate is "non-confident". Specifically, an estimate is classified as being "confident" if the ratio between the two lowest local minima exceeds 2.

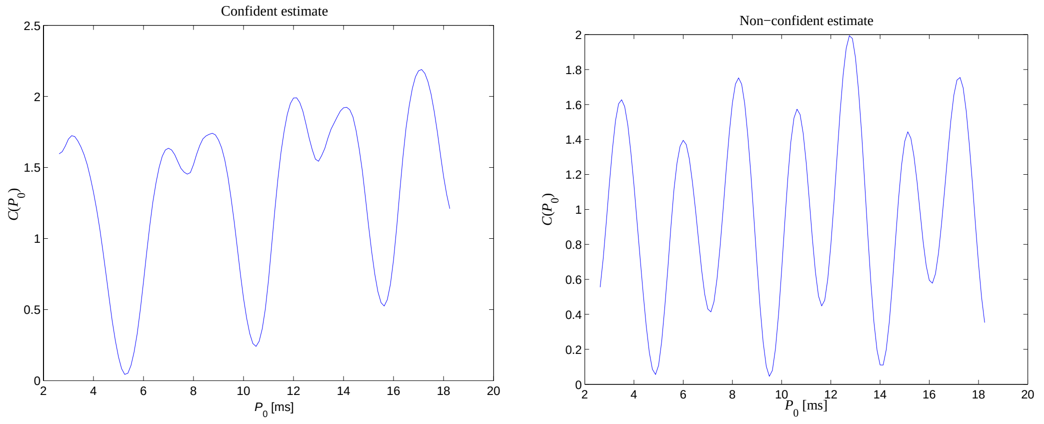

Figure 1.1: The YIN cost-function in a "confident" (left) and a "non-confident" (right) case. Using the right hand side directly would result in a ω_0 halving error (doubling of P_0), since in both cases the true P_0 is around 5 ms corresponding to $f_0 \approx 200Hz$.

Increased robustness is achieved through storing 5 recent "confident" estimates and finding the median of these. Any "non-confident" estimate is then compared to this median value in order to detect if a halving, doubling, or higher multiple error has occurred. If so, the estimate is modified accordingly, e.g. divided by two if a doubling was detected.

Note that when the fundamental frequency is determined so is the number of components, since $K = \lfloor \frac{\pi}{\omega_0} \rfloor$.

1.2.1 Voiced/unvoiced classification

An mentioned earlier the HSM must be modified when applied to unvoiced speech frames. Since the YIN cost function is a measure of the unvoiced-to-total power ratio we found its minimum value (i.e. the value at the fundamental frequency estimate) to be a reasonable voiced/unvoiced classifier. Specifically, if $\min\left(C(P_0)\right) > 0.3$ the unvoiced model is used. When the voiced model is chosen, this measure can also be used for determining a voicing cut-off frequency (sec. 1.3.3).

1.3 Estimation of amplitudes and phases

Amplitudes and phases are estimated from the Weighted Least Squares (WLS) principle, see [MAT90] for the original idea and [Jen00] for the matrix-vector formulation used here. The basic idea is to minimize the energy of the weighted error signal:

$$J = \|\mathbf{W}\left(\mathbf{s} - \hat{\mathbf{s}}\right)\|^2 \tag{1.2}$$

Here $\mathbf{W}$ is a diagonal matrix representing the analysis window, e.g. hanning. In order to solve this minimization problem we rewrite (1.1) using Euler's equation:

$$\hat{\mathbf{s}} = \sum_{k=1}^{K} \frac{A_k}{2} \left(e^{jk\omega_0 \mathbf{t} + j\phi_k} + e^{-jk\omega_0 \mathbf{t} - j\phi_k} \right) = \begin{bmatrix} \mathbf{V} & \mathbf{V}^* \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{c}^* \end{bmatrix} \tag{1.3}$$

where

$$\mathbf{V} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ e^{j\omega_0 \mathbf{t}} & e^{j2\omega_0 \mathbf{t}} & \cdots & e^{jK\omega_0 \mathbf{t}} \\ \vdots & \vdots & & \vdots \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \frac{A_1}{2} e^{j\phi_1} \\ \frac{A_2}{2} e^{j\phi_2} \\ \vdots \\ \frac{A_K}{2} e^{j\phi_K} \end{bmatrix} \tag{1.4}$$

We see that minimizing J over $\mathbf{c}$ becomes a linear LS problem. Writing the solution in terms of a pseudo inverse we obtain[1]:

$$\begin{bmatrix} \mathbf{c} \\ \mathbf{c}^* \end{bmatrix} = \left(\mathbf{W} \begin{bmatrix} \mathbf{V} & \mathbf{V}^* \end{bmatrix} \right)^+ \mathbf{W}\mathbf{s} \tag{1.5}$$

One should note that the pseudo-inverse will yield the LS solution as long as the system of equations is over-determined, i.e. when $L > 2K$ since $\begin{bmatrix} \mathbf{V} & \mathbf{V}^* \end{bmatrix} \in \mathbb{C}^{L \times 2K}$. If $L = 2K$ the system is square and we get a unique solution with perfect reconstruction, but if $L < 2K$ the system is under-determined with non-unique solutions. In the latter case, which may occur in short (< 20 ms), low-pitched frames the pseudo-inverse will yield the *minimum 2-norm* perfect reconstruction solution.

Now, amplitudes and phases can readily be found from $\mathbf{c}$ as the absolute values and angles, respectively. Because the basis vectors are harmonic, in practice[2] $\mathbf{V}^T\mathbf{V} \approx \mathbf{0}$, which results in a nearly block-diagonal matrix in the pseudo-inverse of (1.5). Simulations show that we can therefore compute $\mathbf{c}$ by

$$\mathbf{c} \approx (\mathbf{W}\mathbf{V})^+ \mathbf{W}\mathbf{s}, \tag{1.6}$$

which reduces computational complexity from $\mathcal{O}\left(L(2K)^2\right)$ to $\mathcal{O}\left(LK^2\right)$ i.e. by a factor of 4 if a QR algorithm is used for solving the LS problem, see [GL96] for details.

1.3.1 Modified basis functions

Using the LS approach for estimating amplitudes and phases directly can lead to problems in frames that are not perfectly harmonic. The reason for this is that often, due to small inaccuracies in the f_0 estimate, the estimated harmonics do not match the power spectrum peaks of the original signal at high frequencies (see Figure 1.2), and consequently, amplitude values are underestimated. Perceptually, this results in slight "low-pass" char-

[1] It is straight forward to show that the right hand side is indeed structured $\begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}$ as required

[2] To see this, consider the (m,n)'th element of $\mathbf{V}^H\mathbf{V}$ given by $\mathbf{v}_m^H \mathbf{v}_n = \sum_{t=0}^{T-1} \exp(j(n-m)\omega_0 t)$ versus the elements of $\mathbf{V}^T\mathbf{V}$ given by $\mathbf{v}_m^T \mathbf{v}_n = \sum_{t=0}^{T-1} \exp(j(n+m)\omega_0 t)$. Using the identity:

$$\sum_{t=0}^{T-1} \exp(j\omega t) = \begin{cases} T & \text{if } \exp(j\omega) = 1 \\ \frac{e^{j\omega T} - 1}{e^{j\omega} - 1} & \text{otherwise} \end{cases},$$

we see that as long as T is large and ω_n, ω_m are not both close to zero (or π), the elements of $\mathbf{V}^T\mathbf{V}$ will be small as compared elements near to or on the diagonal of $\mathbf{V}^H\mathbf{V}$.

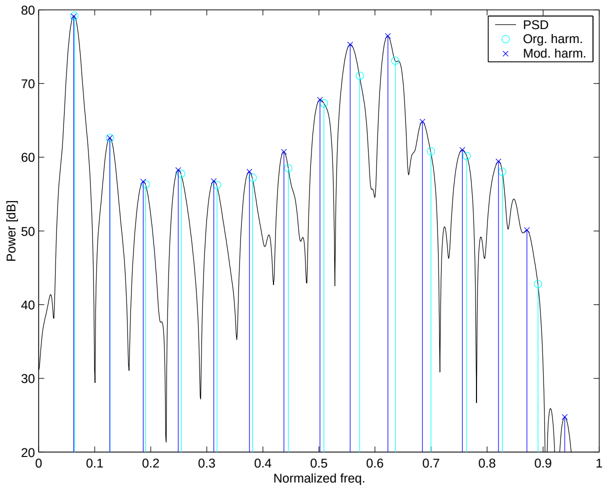

Figure 1.2: Modification of harmonic frequencies to match peaks in the power spectral density (PSD)

acteristics in the reconstructed signal. Similarly to [MQ95] the problem can be relieved by modifying the harmonic frequencies such that they match the nearest peaks in the power spectrum. Denoting the modified k'th harmonic by $\tilde{\omega}_k$ ($\approx k\omega_0$) we use the modified Vandermonde matrix in the LS estimate:

$$\tilde{\mathbf{V}} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ e^{j\tilde{\omega}_1\mathbf{t}} & e^{j\tilde{\omega}_2\mathbf{t}} & \cdots & e^{j\tilde{\omega}_K\mathbf{t}} \\ \vdots & \vdots & & \vdots \end{bmatrix} \tag{1.7}$$

If the modified frequencies $\tilde{\omega}_k$ were to be used for synthesis they would have to be quantized and transmitted thereby adding to the overall bit rate. Instead, we merely estimate amplitudes and phases using $\tilde{\mathbf{V}}$ but synthesize using the original Vandermonde $\mathbf{V}$ since this only requires transmission of the fundamental frequency ω_0.

The discrepancy between the analysis and synthesis basis functions calls for a modification of the estimated phases. To see this, consider the point in time where each modified frequency component has its maximum (denoting the estimated phases by $\tilde{\phi}_k$):

$$\tilde{\omega}_k t_k + \tilde{\phi}_k = 0 \Leftrightarrow t_k = \frac{-\tilde{\phi}_k}{\tilde{\omega}_k} \tag{1.8}$$

Now, the components to be synthesized should be time-aligned with those estimated by LS. Therefore, we seek to maintain these points when synthesizing with the harmonic frequencies instead:

$$t_k = \frac{-\tilde{\phi}_k}{\tilde{\omega}_k} = \frac{-\hat{\tilde{\phi}}_k}{k\omega_0} \Leftrightarrow \hat{\tilde{\phi}}_k = \tilde{\phi}_k \frac{k\omega_0}{\tilde{\omega}_k} \tag{1.9}$$

where $\hat{\tilde{\phi}}_k$ are the phases to be quantized and transmitted.

1.3.2 Phase reference point

In (1.1) the phase is referenced to the beginning of the window, i.e. ϕ_k is the instantaneous phase at the start of the frame. This poses no problem as long as the amplitudes and phases are estimated by the LS method and the same Vandermonde matrix $\mathbf{V}$ is used for both analysis and synthesis. However, when modifying the analysis frequencies as described above, the analysis- and synthesis sinusoids are only in phase at the phase reference point. Specifically, when referencing to the beginning of the frame, the phase dispersion will be 0 here, but grow to $\frac{L}{2}(k\omega_0 - \hat{\omega}_k)$ at the center of the frame and $L(k\omega_0 - \hat{\omega}_k)$ at the end of the frame. If the phase is instead referenced to the center of the frame, the dispersion will be $-\frac{L}{2}(k\omega_0 - \hat{\omega}_k)$ at the beginning, 0 at the center, and $\frac{L}{2}(k\omega_0 - \hat{\omega}_k)$ at the end, i.e. the maximum dispersion is halved. Also, the phase dispersion is located near the frame boundaries, where the synthesis window is close to zero. The phase reference point is easily modified in the Vandermonde/LS framework simply by changing the time index vector from $\mathbf{t} = [0, 1, \ldots, L-1]^T$ to $\mathbf{t} = [-L/2, -L/2+1, \ldots, 0, \ldots, L/2-1]^T$. The need for referencing the phase to the frame center was also reported in e.g. [MQ95].

1.3.3 Voicing cut-off frequency

In many speech frames the harmonic structure is only present in a part of the spectrum. The observation that the harmonic part is often in the low frequency area has lead to the introduction of a *voicing cut-off frequency* [MVSH78]. The main idea is to estimate a frequency ω_c above which a "noise" model is used instead of the harmonic model. We adopt the same idea albeit in a slightly different manner: no change is made to the model above ω_c, instead *subframe phase randomization* [MC97] is applied to these sinusoidal components when synthesized. This simply means that the components in question are synthesized in short, overlapping frames each with a random phase offset, the effect being a smearing of the reconstructed spectrum.

In [MQ95], ω_c is estimated based on a SNR-like measure due to the observation that the more voiced the frame, the better the harmonic sinusoidal model is able to represent it. We take a similar approach here but base it on the LS estimation. Specifically, note that the (windowed) modeled signal segment will be a projection of the (windowed) original signal onto the (windowed) basis functions:

$$\mathbf{W}\hat{\mathbf{s}} = \mathbf{P}\mathbf{W}\mathbf{s} \tag{1.10}$$

Here $\mathbf{P}$ is the projection matrix onto $\mathbf{W}\begin{bmatrix}\tilde{\mathbf{V}} & \tilde{\mathbf{V}}^*\end{bmatrix}$. Using that the modeling error $\mathbf{e}$ is orthogonal to the modeled segment: $\|\mathbf{W}\mathbf{s}\|^2 = \|\mathbf{W}\hat{\mathbf{s}}\|^2 + \|\mathbf{e}\|^2$,this in turn implies that,

$$\frac{\|\mathbf{W}\hat{\mathbf{s}}\|^2}{\|\mathbf{W}\mathbf{s}\|^2} \leq 1 \tag{1.11}$$

with equality only if[3] $\mathbf{s} \in \mathcal{R}\left(\begin{bmatrix}\tilde{\mathbf{V}} & \tilde{\mathbf{V}}^*\end{bmatrix}\right)$, i.e. if the (modified) harmonic model holds. This provides a "fix-point" for determining the voicing cut-off frequency, i.e. if the ratio is 1 the frame is completely voiced and we choose $\omega_c = \pi$.

Now we will find a similar fix-point for a completely unvoiced frame. Here, the basis functions will correspond to a uniform sampling of the relatively smooth unvoiced spectrum. Therefore, on average, the energy along each of the basis vectors will be $\frac{1}{L}$'th of the energy of the windowed signal. Moreover, since the $2K$ basis vectors in $\mathbf{W}\begin{bmatrix}\tilde{\mathbf{V}} & \tilde{\mathbf{V}}^*\end{bmatrix}$ are nearly orthogonal we get that:

$$\text{Unvoiced}: \quad E\left[\frac{\|\mathbf{W}\hat{\mathbf{s}}\|^2}{\|\mathbf{W}\mathbf{s}\|^2}\right] = \frac{2K}{L} \tag{1.12}$$

[3] $\mathcal{R}(\mathbf{A})$ denotes the column space of $\mathbf{A}$

That is, in unvoiced frames the power in the modeled signal is determined by the number of basis functions used to represent it. Remember however, that the basis functions are modified to fit the spectral peaks which implies that the ratio will actually be higher than indicated. For this reason, we scale the fraction above with a constant $a > 1$. This provides a fix-point so that if $\frac{\|\mathbf{W}\hat{\mathbf{s}}\|^2}{\|\mathbf{W}\mathbf{s}\|^2} \leq a\frac{2K}{L}$ the cut-off frequency is set to $\omega_c = 0$.

Having determined the two extreme points these are linearly interpolated for determining the voicing cut-off frequency as illustrated in Figure 1.3.

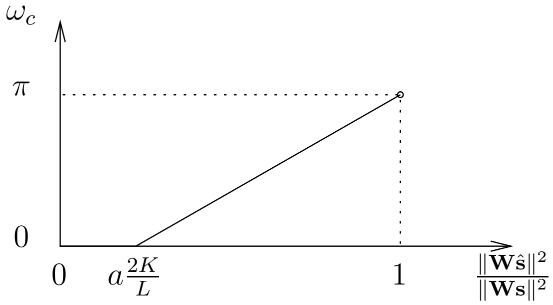

Figure 1.3: Determination of voicing cut-off frequency from the modeled-to-original signal power ratio.

The measure described here makes no sense in short, low-pitched frames, since then $a\frac{2K}{L} > 1$. In this case we revert to determining the cut-off frequency from the unvoiced-to-total power ratio as found by the YIN algorithm, see [RJ02] for details.

1.4 Unvoiced frames

So far, we have described the parameter estimation in voiced frames. When a frame is classified as being unvoiced there is no reliable fundamental frequency and the HSM has no physiological meaning. It has been argued, however, that the HSM can still be used as long as the frequency spacing ω_0 is keep low enough [MQ95]. The reason for this is easily understood when considering the LS estimation technique: the lower the ω_0 the more columns (basis functions) we get in $\mathbf{V}$ which in turn means that the system of linear equations becomes “less over-determined”. When the number of components reaches half the frame length ($2K = L$) the linear system solved by (1.5) is square and we in principle have perfect reconstruction. From initial listening experiments we found that picking ω_0 such that $K = \frac{L}{4}$ gives a sufficiently dense sampling of the frequency spectrum. Note that since the frequency spacing is uniquely determined from the frame length it does not have to be transmitted.

In principle, once the frequency spacing has been determined, the LS estimation procedure that was used in voiced frames could also be used for estimating amplitudes and phases in unvoiced frames. However, the computational complexity is quite high for long frames (and thus high K). Moreover, an LS fit of amplitudes and phases is overkill since we will actually not need the phases (randomized at synthesis, see sec. 1.7). Instead, we simply determine the amplitudes through sampling of the power spectral density. Finally, to “smear” the reconstructed spectrum, subframe randomization is used in the synthesis in the same way as for the unvoiced components in semi-voiced frames (sec. 1.3.3).

1.5 Quantization of fundamental- and cut-off frequency

The fundamental frequency is quantized in the log-domain, the range set between 55 Hz and 400 Hz for which 8 bits has been found adequate. In [MAT90] only 7 bits very used for a log-domain quantizer but from listening experiments we found improvements for some speakers when using 8 bits.

The voicing cut-off frequency is simply quantized linearly in the range $[0, \pi]$ using 4 bits.

1.6 Amplitude quantization

Speech spectral envelopes are usually represented by an AR model, often estimated by Linear Prediction (LP). When representing a discrete under-sampled spectrum as in this case, however, LP leads to aliasing in the autocorrelation function which makes Discrete All-Pole (DAP) modeling a better choice [EJM91]. Very briefly, this method minimizes the Itakura-Saito distance between the measured amplitudes and the AR model sampled at the same frequencies. The minimization problem is solved iteratively using conventional LP as the initial estimate. In [MPSC98] it is reported that DAP is more accurate than other methods, including the cubic spline approach of [MQ95].

The DAP curve is fitted to the frequency-amplitude pairs $(k\omega_0, \tilde{A}_k)$ where $\tilde{A}_k$ are the amplitudes estimated using the modified frequencies $\tilde{\omega}_k$. The reason for this is that the harmonics $k\omega_0$ define the points at which the DAP curve is resampled at the receiver. This way, the problem with overshoot at high frequncies as reported in [MHC00] is avoided.

The AR model gain factor is readily quantized in the logarithmic domain using 5 bits. Quantization of the AR polynomial coefficients has received a great deal of attention, the prevailing technique being split vector quantization of the LSF parameters. In this report we do not quantize AR model coefficients; instead we refer to [PA93] where transparent quantization of a 10th order AR model is achieved using 24 bits. Since we will need to be able to work with other AR model orders, a linear relationship between the model order P and the number of bits R_{AR} required to obtain transparent quantization is assumed. Specifically, the following relationship is used:

$$R_{AR} = \frac{3}{2}P + 9 \tag{1.13}$$

1.7 Phase quantization

The quantizations of fundamental frequency, power and AR model all follow conventional methods. In case of the phases, however, we develop a new scheme. The reason for this is that in harmonic sinusoidal coding schemes, phases are usually encoded through exploitation of phase prediction, i.e. the phases of each frame is represented partly as a function of the phases and frequencies of the previous frame, see e.g. [AS96], [MAT90]. This approach is not feasible in our case because of the probability of packet losses.

Instead we take another approach that exploits the relationship between the phases within each frame. First, the time instants where each sinusoidal component has its maxima are calculated by:

$$\begin{aligned} t_k &= \arg\max_t \left\{ \cos\left(k\omega_0 t + \phi_k\right) \right\} \\ &= \frac{2\pi q - \phi_k}{k\omega_0}, \quad q \in \mathcal{Z} \end{aligned} \tag{1.14}$$

These points are plotted versus the harmonic frequencies in Figure 1.4 for a frame of voiced speech. In this plot, note that if one + is known at each harmonic the phases can be reconstructed without error. If the speech signal was produced by a zero phase system (as assumed in e.g. [QM89]), a straight horizontal line would fit perfectly a + for each harmonic. For natural speech this is usually not the case, but the +'s are clearly correlated from harmonic to harmonic. We exploit this relationship in the encoding by using a piecewise linear representation that can readily be quantized as will be explained in the following.

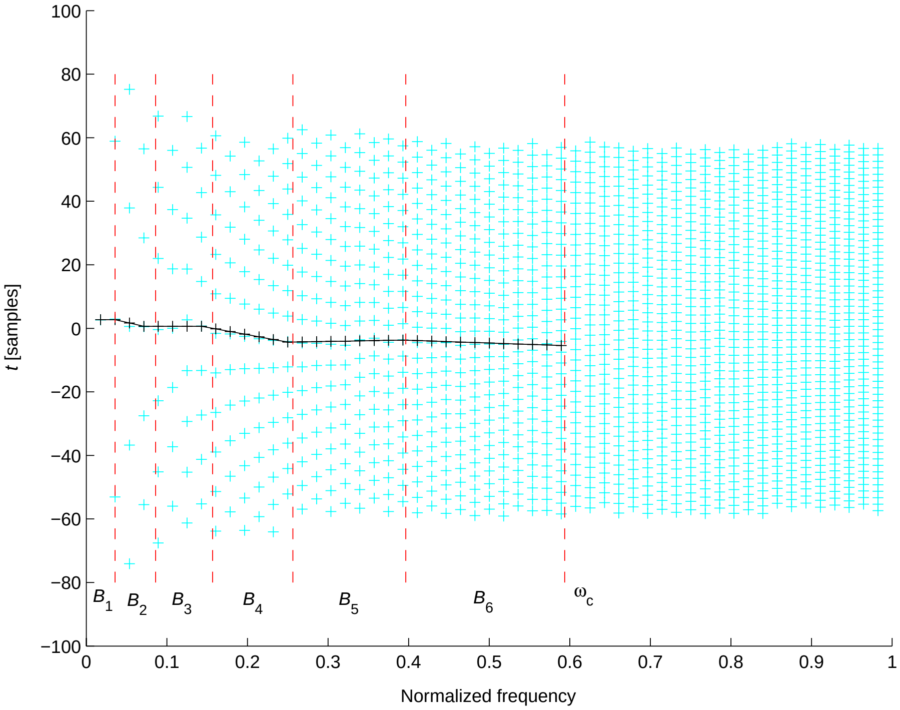

Figure 1.4: Phase quantization principle. The +'s represents the time instants where each sinusoidal component has its maxima. The piecewise linear function represents these points quantized and thus the phases, whereas the vertical lines represents the phase bands.

1.7.1 Phase bands

The first step in forming the piecewise linear representation is determining the phase bands, i.e. the domain for each linear component of the piecewise linear function. These bands should cover the interval from the fundamental frequency up to the voicing cut-off frequency ω_c since the phases are randomized above. Moreover, because speech signal energy is usually concentrated in the low frequency area, it seems feasible that the bands are more narrow here; this frequency division is also well in line with the "critical band decomposition" typically used in perceptual audio coding [PS00]. Finally, it should be possible to reconstruct the bands at the receiver without additional side information. The requirements are fulfilled in the following way, which has been found to work well in initial simulation studies:

1. The first band contains the first two frequency components, i.e. it covers the interval $B_1 =]0, 2\omega_0]$, so that width$(B_1) = 2\omega_0$.

2. The lengths of the following bands increase exponentially so that width$(B_m) = 2\omega_0\alpha^{m-1}$ for $m = 2 : M$, where M is the predetermined number of bands.

The factor α is determined so that the bands cover the interval $]0, \omega_c]$:

$$2\omega_0 \sum_{m=1}^{M} \alpha^{m-1} = \omega_c \tag{1.15}$$

i.e. we pick α as the positive, real root of this polynomial[4]. An example of this band design is included in Figure 1.4.

The number of bands is determined from the trade-off between a high number of bands and a high number of bits to represent the phases within each band (how this is done will be explained below). Simulations showed that using 4-5 bits in each phase band resulted in a reconstruction of reasonable quality. Therefore, having a total of, say, R_p bits for representing all voiced phases the number of bands is determined by:

$$M = \lfloor \frac{R_p}{4.5} \rfloor \tag{1.16}$$

This leaves between 4 and 5 bits for each band, therefore 5 bits are used in low frequency bands and 4 bits in higher frequency bands.

1.7.2 Estimation and quantization of piecewise linear phase function

We now want to fit a piecewise linear function to the sinusoidal maxima, as illustrated by the black curve in Figure 1.4. The basic approach is to start in B_1, and then work from the left to the right, using the end point in B_m as the starting point in B_{m+1}.

To explain the principle in finding the possible line end points in each band we zoom into two of the phase bands from Figure 1.4, as shown in Figure 1.5. The candidate end points are constructed as follows: one point is at the level of the starting point, as indicated by the horizontal dashed lines. The rest of the points are now placed relative to this, the spacing being a fraction of the period of the last (i.e. highest frequency) sinusoidal component in the band, e.g. in figure 1.5 the spacing is $\frac{1}{4}$ of the period. It turns out to be necessary to span two periods "up" and two "down". This in turn means that the resolution will be $\frac{1}{4}$ periods (for 4 bits per band) or $\frac{1}{8}$ periods (for 5 bits per band).

[4] The uniqueness of this solution follows directly since each term in the polynomial (1.15) is monotonically increasing for $\alpha > 0$ and thus the left-hand side is monotonically increasing for $\alpha > 0$. Moreover, if $\omega_c < 2\omega_0 M$ there will not be "room" for increasing bandwidths and we will get $\alpha < 1$, i.e. decreasing bandwidths. In this case we use equal bandwidths, width$(B_m) = \frac{\omega_c}{M}$, $\forall m = 1 : M$.

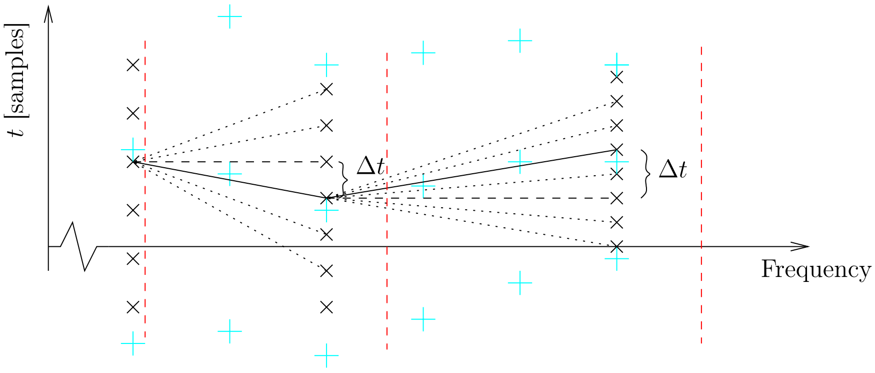

Figure 1.5: Choosing linear components in phase quantization. Light +'s denote maximum points for sinusoidal components and vertical lines represents the phase band limits. ×'s show possible line ending points, dotted lines possible linear functions and solid lines chosen linear functions.

Picking linear components

As indicated in Figure 1.5 each component of the piecewise linear function is chosen as the one best matching the sinusoidal maximum points. Specifically, in each band m there is I_m possible line end points and thus I_m possible linear functions. Each of these can be seen as a vector:

$$\tilde{\mathbf{t}}_{i,m} = \left\{ t_i(\omega_m^{(1)}), t_i(\omega_m^{(2)}), \ldots, t_i(\omega_m^{(G_m)}) \right\}, \quad \text{for } i = 1 : I_m$$

where G_m is the number of components in the m'th band and $\omega_m^{(g)}$ is the g'th harmonic frequency within this band. That is, the vector $\tilde{\mathbf{t}}_{i,m}$ contains locations of the sinusoidal maxima in band m, attained at synthesis is the i'th possible line was used. In Figure 1.5, the dimension of $\tilde{\mathbf{t}}_{i,m}$ will be 2 in the left band and 3 in the right band.

We now arrange the *measured* sinusoidal maxima in a similar vector $\mathbf{t}_m$. Note that since each harmonic has multiple maxima it will be necessary to determine which of these to use. A discussion of this follows below, but for now we assume that the decision is made already. Then the best linear function is chosen as the one minimizing the weighted distance measure:

$$D_i = \|\mathbf{W}_m \left(\tilde{\mathbf{t}}_{i,m} - \mathbf{t}_m\right)\|_2, \tag{1.17}$$

where $\mathbf{W}_m$ is a diagonal weighting matrix:

$$\mathbf{W}_m = \operatorname{diag}\left\{ A_m^{(1)}\omega_m^{(1)}, A_m^{(2)}\omega_m^{(2)}, \ldots, A_m^{(G_m)}\omega_m^{(G_m)} \right\}$$

Here, $A_m^{(g)}$ is the amplitude of the g'th sinusoid within the band. Weighting by the amplitudes ensures that high-energy harmonics are modeled most precisely, whereas weighting by the frequencies normalizes the distance between $\tilde{\mathbf{t}}_{i,m}$ and $\mathbf{t}_m$ with respect to the sinusoidal period. The latter is necessary since otherwise the contribution to the distance measure at higher frequencies would be negligible, see Figure 1.4.

Choosing "best" sinusoidal maximum points

In the description above we did not specify which of the sinusoidal maxima in the vertical time-axis (+'s in plots) the fit is actually matched to. Since we only have to model one +

for each harmonic, a straight-forward way of doing this is to calculate (1.17) for the +'s ($\mathbf{t}_m$'s) nearest to each possible line segment $\tilde{\mathbf{t}}_{i,m}$. This approach can lead to problems, though, as illustrated in Figure 1.6. Here, the algorithm will take either the solid or the dashed path through the +'s (ignore the dotted line for now). Which one is actually taken depends on which of the two +'s at ω_k is closest to a possible line ending point. However, the chosen line ending point may be a poor starting point for the line in the next band. Supported by listening experiments we therefore found it reasonable always to follow the "smoothest" path through the +'s, found by going from the left to the right, picking each new + as the one closest to the current. Another argument is that when the harmonics are highly aligned (linear phases) the smooth path will approximate a horizontal line, whereas other paths will approximate hyperbolas, see e.g. Figure 1.4.

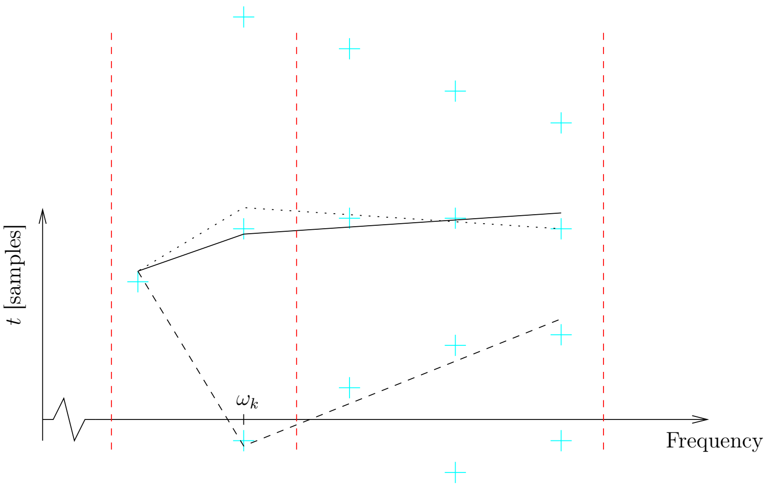

Figure 1.6: Different paths through the sinusoidal maximum points. The solid and dashed lines show two alternative paths, but by finding the smoothest track of +'s, the solid line will always be chosen. The dotted line is a better representation that is not found due to greediness.

Greediness

While using the smooth path as described above increases speech quality, it does not solve the inherent problem of *greediness* in the proposed method: we try to solve a sequence of dependent subproblems (selecting a linear function for each band) by choosing the optimal solution for each. As a result there may be a better representation of the harmonic maxima than found by the proposed method, such as the one sketched in Figure 1.6. A possible work-around is to apply a delayed decision technique such as the M-algorithm [GG92] or modified methods as those applied to Matching Pursuit in [CR01].

Starting point - onset time

The starting point in the first band is encoded separately as an "onset time" using 8 bits. Since the minimum allowable fundamental frequency is 55 Hz this onset time can occur at most $\frac{8\text{kHz}}{55\text{Hz}} \approx 145$ samples after the frame start. This results in a resolution of approximately half a sample in the onset quantization.

Unfortunately, for some signals the first harmonic has very low energy (this can also be caused by high pass filtering), resulting in an unreliable onset estimate. Therefore we instead use the second harmonic for the onset estimation. Note however that there will be two possible onset times for this approach (in Figure 1.4 above sample 0 or below sample 40). Which of these to use is determined by carrying out the entire phase quantization for both and then picking the one that results in the best piecewise linear match.

1.8 Summary

In this chapter we described a speech coder based on harmonic sinusoidal modeling. The coder has a “voiced” and an “unvoiced” coding mode and is flexible in that it can work on any input frame lengths. The main difference between the two coding modes is that a pitch estimate is used as a fundamental frequency in voiced frames, whereas the frequency spacing in unvoiced frames is determined exclusively from the frame length. Moreover, phases are randomized in unvoiced frames and need thus not to be transmitted. Amplitudes are represented using an AR model estimated through DAP. A new coding scheme that does not rely on interframe information were developed for the phases since this is an important requirement for speech coding in VoIP.

Chapter 2

Receiving end PLC algorithm

In the following we will describe a PLC algorithm based on that of [RCAJ03] but modified to fit the framework at hand.

As in most PLC algorithms the frame loss is sought compensated for through the contents of neighbor packets. In the case of a harmonic sinusoidal coder two possibilities seem feasible:

1. Interpolation of the sinusoids over the lost interval.

2. Stretching sinusoids from neighboring packets into the lost interval.

A straight-forward extension is to use a combination of the two possibilities: if two components on each side of the loss seem similar these should be linked through interpolation whereas dissimilar could be stretched from each side ending up as overlap-add, see Figure 2.1. The figure also indicates that interpolation is carried out by replacing the lost interval with a windowed set of new parameters.

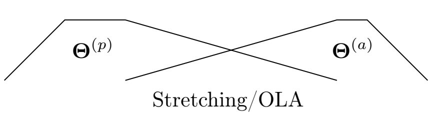

Figure 2.1: Top: shaded frames represent lost frames. Middle: lost frames replaced by interpolated components. Bottom: neighbor packets stretched into lost interval. Note that actual window slopes are hanning but shown linear to ease the illustration. $\mathbf{\Theta}^{(p)}$ and $\mathbf{\Theta}^{(a)}$ denote the sets of sinusoidal parameters in the frames previous to and after the loss, respectively, whereas $\mathbf{\Theta}^{(i)}$ is the set of interpolated parameters.

2.1 Component classification

The first step in the PLC algorithm is to classify which components to interpolate and which to overlap-add. Because the sinusoidal model is harmonic the frequency linking problem (see e.g. [MQ86]) is trivial since the k'th harmonic in one frame is simply linked to the k'th in another. The k'th harmonic is classified for interpolation over the missing interval if the following three conditions are met (using the (p) and (a) superscripts to denote frames previous to and after the loss as above):

1. The k'th harmonics are below the voicing cut-off frequencies both before and after the loss,
$$k\omega_0^{(p)} < \omega_c^{(p)} \text{ and } k\omega_0^{(a)} < \omega_c^{(a)}$$
2. The absolute frequency difference does not exceed 100 Hz,
$$\left| k\omega_0^{(p)} - k\omega_0^{(a)} \right| < \frac{2\pi \cdot 100 \text{ Hz}}{8 \text{ kHz}}$$
3. The ratio between the amplitudes of the components is below 5,
$$\max \left\{ \frac{A_k^{(a)}}{A_k^{(p)}}, \frac{A_k^{(p)}}{A_k^{(a)}} \right\} < 5$$

These heuristic rules are similar to those used in [RCAJ03], and tuned through informal listening. The first rule simply ensures that unvoiced components are overlap-added whereas the second and third prevent interpolation of dissimilar components.

In practice, one additional rule is necessary: if no packet after the losses is present interpolation is not possible, and thus OLA has to be used. How this is exactly done will be explained below.

2.2 Overlap-added components

Overlap-adding is simply carried out by extrapolating the sinusoidal components into the missing interval and applying the corresponding window slope. Subframe randomization is applied to components above the voicing cut-off frequency ω_c as usual.

For very long packet losses the stretching/OLA strategy indicated in Figure 2.1 is not possible because the packet proceeding the losses will not yet be available in the jitter buffer. In this case OLA is carried out as sketched in Figure 2.2. The basic principle is that play-out of the packet proceeding the losses is started as soon as it arrives.

2.3 Interpolated components

When working on a sinusoidal model, interpolating over missing frames bears strong resemblances to time-scale modification operations, see e.g. [QM86]. Such applications usually utilize a cubic phase model since this allows for perfect matching of phases and frequencies at both frame boundaries. However, in our experience such an approach leads to problems in PLC applications, because the cubic phase model tends to break down the harmonic signal structure. Instead we use a quadratic phase model, which preserves the harmonic structure at the cost of a phase mismatch at frame boundaries. The quadratic

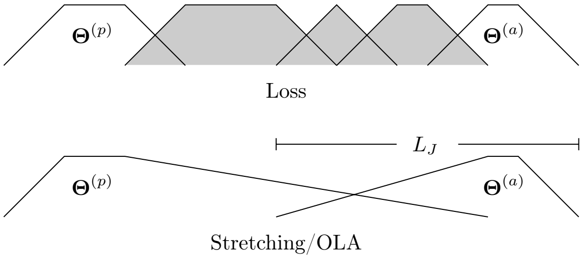

Figure 2.2: OLA in case of long packet losses. L_J denotes the jitterbuffer length.

model arises when linearly interpolating frequencies (and amplitudes) over the missing interval (remembering that phases are referenced to the center of each frame):

$$s_k^{(i)}(t) = A_k^{(i)}(t)\cos\left(\theta_k^{(i)}(t)\right), \quad t = -\frac{L^{(i)}}{2}, \ldots, \frac{L^{(i)}}{2} - 1 \tag{2.1}$$

where

$$\begin{aligned} A_k^{(i)}(t) &= \frac{A_k^{(p)} + A_k^{(a)}}{2} + \frac{A_k^{(a)} - A_k^{(p)}}{L^{(i)}} t \\ \theta_k^{(i)}(t) &= k\alpha t^2 + k\omega_0^{(i)} t + \phi_k^{(i)} \\ \omega_0^{(i)} &= \frac{\omega_0^{(p)} + \omega_0^{(a)}}{2} \\ \alpha &= \frac{\omega_0^{(a)} - \omega_0^{(p)}}{2L^{(i)}} \end{aligned}$$

By inspection we see that this model ensures that the amplitudes and instantaneous frequencies at the beginning and end of the frame equal the parameters from the neighboring frames, i.e.

$$\begin{aligned} A_k^{(i)}\left(-\frac{L^{(i)}}{2}\right) &= A_k^{(p)} \\ A_k^{(i)}\left(\frac{L^{(i)}}{2}\right) &= A_k^{(a)} \\ \frac{d}{dt}\theta_k^{(i)}\left(-\frac{L^{(i)}}{2}\right) &= \frac{d}{dt}\left[k\alpha t^2 + k\bar{\omega}_0 t + \phi_k^{(i)}\right]_{t=-\frac{L^{(i)}}{2}} = k\omega_0^{(p)} \\ \frac{d}{dt}\theta_k^{(i)}\left(\frac{L^{(i)}}{2}\right) &= \frac{d}{dt}\left[k\alpha t^2 + k\bar{\omega}_0 t + \phi_k^{(i)}\right]_{t=\frac{L^{(i)}}{2}} = k\omega_0^{(a)} \end{aligned}$$

The only parameter yet to be determined is the phase of each component ϕ_k. This is done by minimizing the total phase mismatch at the center of the OLA regions denoted by $M^{(p)}$ and $M^{(a)}$ in Figure 2.1. The cost function to minimize is[1]

$$J(\phi_k^{(i)}) = \min_{P^{(p)},P^{(a)}\in\mathcal{Z}} \left\{ \left(\theta_k^{(i)}(\tilde{M^{(p)}}) - \theta_k^{(p)}(\tilde{M^{(p)}}) - 2\pi P^{(p)}\right)^2 + \left(\theta_k^{(i)}(\tilde{M^{(a)}}) - \theta_k^{(a)}(\tilde{M^{(a)}}) - 2\pi P^{(a)}\right)^2 \right\} \tag{2.2}$$

Here $\theta_k^{(p)}(\tilde{M^{(p)}})$ and $\theta_k^{(a)}(\tilde{M^{(a)}})$ are the instantaneous phases of the k'th component in the neighbor frames, directly found from the (constant) frequencies and center phases here:

$$\begin{aligned} \theta_k^{(p)}(\tilde{M^{(p)}}) &= \phi_k^{(p)} + k\omega_0^{(p)}\frac{L^{(p)} - L_{OL}}{2} \\ \theta_k^{(a)}(\tilde{M^{(a)}}) &= \phi_k^{(a)} - k\omega_0^{(a)}\frac{L^{(a)} - L_{OL}}{2} \end{aligned}$$

where $L^{(p)}$ is the length of the of the frame (p) previous to the loss and L_{OL} is the length of the analysis frame overlap region so that $\tilde{M^{(p)}} = \frac{L^{(p)}-L_{OL}}{2}$ is the distance from the center of frame (p) to $M^{(p)}$. Similarly we have that

$$\begin{aligned} \theta_k^{(i)}(\tilde{M^{(p)}}) &= \phi_k^{(i)} + k\omega_0^{(i)}\frac{L^{(i)} - L_{OL}}{2} + k\alpha\left(\frac{L^{(i)} - L_{OL}}{2}\right)^2 = \phi_k^{(i)} + \Delta\theta_k^{(i)}(\tilde{M^{(p)}}) \\ \theta_k^{(i)}(\tilde{M^{(a)}}) &= \phi_k^{(i)} - k\omega_0^{(i)}\frac{L^{(i)} - L_{OL}}{2} + k\alpha\left(\frac{L^{(i)} - L_{OL}}{2}\right)^2 = \phi_k^{(i)} + \Delta\theta_k^{(i)}(\tilde{M^{(a)}}) \end{aligned}$$

where $\Delta\theta_k^{(i)}(\tilde{M^{(p)}})$ and $\Delta\theta_k^{(i)}(\tilde{M^{(a)}})$ are introduced for notational reasons.

In (2.2) the terms $2\pi P^{(p)}$ and $2\pi P^{(a)}$ are included in order to obtain the phase difference modulo 2π and can be found by:

$$\begin{aligned} P^{(p)} &= \text{round}\left(\frac{\theta_k^{(i)}(\tilde{M^{(p)}}) - \theta_k^{(p)}(\tilde{M^{(p)}})}{2\pi}\right) \\ &= \text{round}\left(\frac{\Delta\theta_k^{(i)}(\tilde{M^{(p)}}) - \theta_k^{(p)}(\tilde{M^{(p)}}) + \phi_k^{(i)}}{2\pi}\right) \\ &= \text{round}\left(\frac{\Delta\theta_k^{(i)}(\tilde{M^{(p)}}) - \theta_k^{(p)}(\tilde{M^{(p)}})}{2\pi}\right) + q^{(p)}, \quad q^{(p)} \in \{0, 1\} \end{aligned} \tag{2.3}$$

where in the last line we used that $\phi_k^{(i)} \in [0, 2\pi]$. A very similar expression is obtained for $P^{(a)}$ resulting in a total of 4 possible combinations of $P^{(p)}$ and $P^{(a)}$. Now we simply minimize (2.2) for each of the 4 possibilities and pick the one yielding the least minimum. The minimization itself is straight-forward since (2.2) is reduced to a quadratic function in $\phi_k^{(i)}$.

[1] $\tilde{M^{(p)}}$ indicate that the time index is not the absolute $M^{(p)}$ but instead the center of the OLA region measured relative to the center of the index frame.

List of Figures

Bibliography

[AS96] S. Ahmadi and A. S. Spanias. New Techniques for Sinusoidal Coding of Speech at 2400 bps. In *Proc. Asilomar*, pages 770–774, 1996.

[CR01] S. F. Cotter and B. D. Rao. Application of Tree-based Searches to Matching Pursuit. In *Proc. ICASSP*, pages 3933–3936, 2001.

[dCK02] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. In *Journal of ASA*, volume 111(4), April 2002.

[EJM91] A. El-Jaroudi and J. Makhoul. Discrete All-Pole Modeling. In *IEEE Trans. on Signal Processing*, volume 39, pages 411–423, 1991.

[GG92] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.

[GL96] G. H. Golub and C. F. Van Loan. *Matrix Computations*, chapter 5. The Johns Hopkins University Press, 3rd edition, 1996.

[Jen00] J. Jensen. *Sinusoidal Models for Speech Signal Processing.* PhD thesis, CPK, Institute of Electronic Systems, Aalborg University, 2000.

[MAT90] J. S. Marques, L. B. Almeida, and J. M. Tribolet. Harmonic Coding at 4.8kb/s. In *Proc. IEEE ICASSP*, pages 17–20, December 1990.

[MC97] M. W. Macon and M. A. Clements. Sinusoidal Modeling and Modification of Unvoiced Speech. In *IEEE Transactions on Speech and Audio Processing*, volume 5, pages 557–560, November 1997.

[MHC00] D. J. Molyneux, M. S. Ho, and B. M. G. Cheetham. Robust Application of Discrete All-Pole Modeling to Sinusoidal Transform Coding. In *Proc. IEEE ICASPP*, volume 3, pages 1455–1458, 2000.

[MPSC98] D.J. Molyneux, C.I. Parris, X.Q. Sun, and B.M.G. Cheetham. Comparison of Spectral Estimation Techniques for Low Bit-Rate Coding. In *Proc. International Conference on Spoken Language Processing*, dec. 1998.

[MQ86] R. J. McAulay and T. F. Quatieri. Speech Analysis/Synthesis Based on a Sinusoidal Representation. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-34(4):744–754, aug 1986.

[MQ95] R. J. McAulay and T. F. Quatieri. *Sinusoidal Coding*, chapter 4. Elsevier Science B.V., 1995. From *Speech Coding and Synthesis*, Edited by W.B Kleijn and K.K. Paliwal.

[MVSH78] J. Makhoul, R. Viswanathan, R. Schwartz, and A. W. F. Huggins. A Mixed-source Model for Speech Compression and Synthesis. In *Proc. IEEE ICASSP*, volume 3, pages 163–166, 1978.

[PA93] K. K. Paliwal and B. S. Atal. Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame. In *IEEE Trans. on Speech and Audio Processing*, volume 1, pages 3–14, 1993.

[PS00] T. Painter and A. S. Spanias. Perceptual Coding of Digital Audio. In *Proc. IEEE*, volume 88(4), pages 451–513, April 2000.

[QM86] T. F. Quatieri and R. J. McAulay. Speech transformations based on a sinusoidal representation. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-34(6):1449–1464, Dec. 1986.

[QM89] T. F. Quatieri and R. J. McAulay. Phase Coherence in Speech Reconstruction for Enhancement and Coding Applications. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 207–210, 1989.

[RCAJ03] C. A. Rødbro, M. G. Christensen, S. V. Andersen, and S. H. Jensen. Compressed Domain Packet Loss Concealment of Sinusoidally Coded Speech. In *Proc. IEEE ICASSP*, pages 104–107, 2003.

[RJ02] C. A. Rødbro and S. H. Jensen. Time-scaling of Sinusoids for Intelligent Jitter Buffer in Packet Based Telephony. In *IEEE Proc. Workshop on Speech Coding*, pages 71–73, 2002.