# Restoration of pitch contours

Driessen, J.A.J.

*Award date:*
1995

# Restoration of pitch contours

Jacques Driessen

October 1, 1993

## Abstract

A restoration method to restore unknown samples in discrete-time signals is used to improve the quality of pitch contours. These pitch contours are produced by a pitch determination algorithm based on the summation of subharmonics, that determines the pitch contour on a per-point basis. The restoration is a two-step procedure. In the first step the pitch contour is divided into reliable and unreliable points. The second step uses the restoration method to make estimates for the unreliable points from the reliable points and a model for the pitch contour. An algorithm is presented to determine which points are reliable in the pitch contour and which are not. The restored pitch contours are compared to pitch contours produced by another pitch determination algorithm, also based on the summation of subharmonics, but where the whole pitch contour is determined using dynamic programming techniques. It cannot be conclusively determined whether the restored pitch contours or the pitch contours determined by the pitch determination algorithm using dynamic programming techniques are better.

# Contents

# Chapter 1

# Introduction

In speech perception research, it is often valuable to have a measurement of the pitch as a function of time for a speech utterance. This is called a pitch contour. To make a pitch contour, the speech is recorded and fed to a device that determines the pitch as a function of time. These days, this is usual done by numerical analysis in a computer. This computer is fed with a sampled and digitized version of the speech utterance. The general procedure then is to divide this sampled data in so-called segments. The segment length is chosen to be of the order of the response-time of the human auditory system. For each segment the pitch is determined using a pitch determination algorithm. This then yields the pitch contour.

It is not always possible to measure the pitch correctly for all segments. This can be so because of a number of possibilities. First of all, natural speech always contains unvoiced parts, that have a noisy character, to which no pitch can be attributed. Any pitch measurement will produce meaningless pitch estimates for unvoiced parts. The second possibility is that due to environmental, electronic or measurement noise, for example other voices, ordinary noise, background music etc., it becomes impossible for the pitch determination algorithm to separate the noise from the signal. This then yields unreliable estimates for the pitch or even estimates for the pitch from the noise instead of the pitch from the signal. The third possibility is that the pitch determination, if at all, mimics the human method of determining the pitch, it never does so perfectly, which leads to errors in the pitch determination. Particularly troublesome are the so called octave failures, where the pitch is estimated one octave to high or to low. These are inherent to many of the pitch determination algorithms.

Except for the erroneous pitch estimates, another thing corroborates the determination of pitch contours. These are the octave failures in the so-called creaky voices. These are not to be confused with the octave failures as produced by the pitch determination algorithm. The difference lies in the fact that for creaky voices, the octave failures correspond to an actual shifting of one octave of the pitch, while the other kind of octave failures are errors of the pitch determination algorithm.

Many pitch determination algorithms correctly measure octave failures produced by a creaky voice. If these octave failures are shifted back with a computer, using speech-analysis and resynthesis techniques, it is impossible for an inexperienced user to hear the difference between the "corrected" resynthesized and the original sentence. This is called perceptual equivalence.

Since these octave failures cannot be heard, it is desirable to have a pitch contour that does not contain them anymore. This is then called a "corrected" ore restored pitch contour. In this report a method is presented to make such a "corrected" pitch contour. This is done as follows. First, the pitch contour is made by an algorithm presented in [4]. Secondly, the octave failures as produced by a creaky voice, as well as unvoiced segments, and low-intensity segments are detected, using an algorithm presented in this report. These have to be detected, since the pitch estimates for these segments have to be corrected in some way, either because they are wrong pitch estimates - the latter two cases -, or undesirable - first case -. The low-intensity segments are included because for these segments it is likely that the different noise-sources are of the same order of magnitude as the signal, which may lead to faulty pitch estimates.

The third step is the "correction procedure". For this, restoration methods for unknown samples in discrete-time signals, presented in [9] are used. The restoration methods suppose that the pitch estimates for the segments, for which the pitch estimates are to be corrected, are unknown. These are then restored with a restoration method from [9], using the "correct" (also called known) pitch estimates and a model for the pitch contour. If this model is (partly) estimated from the known pitch estimates the restoration method is called adaptive.

In this report, first the restoration methods are described in Chapter 2. It starts with the general restoration method, followed by two general classes of restoration methods, one based on an autoregressive model for the data, the other based on a band-limited model for the data. In Chapter 3 the pitch determination algorithm is described. Furthermore a short qualitative explanation of another pitch determination algorithm to make a "corrected" pitch contour is given, that will be used to compare the restored pitch contours with. In Chapter 4 the results of the restoration of pitch contours are presented and discussed. It starts with introducing a method to measure the quality of a restored pitch contour by comparing it with the other method of making a "corrected" pitch contour. It then continues with an investigation to determine if the restoration methods can be used to make restorations of a pitch contour. This is followed by a description of the algorithm that determines which pitch estimates are "correct" and which are not. This is completed with a graphical presentation of restored pitch contours and pitch contours produces by the other method of making a “corrected” pitch contour. These are compared and shortly discussed. The report ends with some conclusions in Chapter 5.

# Chapter 2

# Restoration

## 2.1 Introduction

In this chapter a method will be described to restore unknown samples in a discrete-time sequence[1]. This means that there is a sequence of samples

$$s_k,\ k = a, a+1, \ldots, b,$$

from which the samples

$$s_{t(i)},\ a \leq t(1) < t(2) < \ldots < t(m) \leq b,$$

are unknown and have to be restored. Here $m$ is a finite integer, $a$ and $b$ integers that may be chosen to be equal to $-\infty$ and $\infty$, respectively. In the latter case there is an infinite sequence of samples with only a finite number of unknown samples. In the remainder of this chapter it is assumed that the given sequence of samples is (part of) a realization of a stochastic process $\underline{s}_i$, $i = -\infty, \ldots, \infty$, which is stationary up to at least order 2 unless explicitly stated otherwise. The condition that a stochastic process $\underline{s}_i$, $i = -\infty, \ldots, \infty$, is stationary up to order 2 means that

$$\mathcal{E}\left\{\underline{s}_i\right\},\ \mathcal{E}\left\{\underline{s}_i \underline{s}_{i+k}\right\}$$

are independent of the index $i$. The $\mathcal{E}\left\{\right\}$ denotes the expectation value operator. From now on the term stationarity will be used to denote stationarity up to order 2. Stationarity is required in order to be able to define and make use of the autocorrelation function of a stochastic process, which is defined by

$$R(k) = \mathcal{E}\left\{(\underline{s}_i - \mu)(\underline{s}_{i+k} - \mu)\right\}, k = -\infty, \ldots, \infty, \tag{2.1}$$

$$\mu = \mathcal{E}\left\{\underline{s}_i\right\}\ . \tag{2.2}$$

[1] This method was developed by R.N.J. Veldhuis e.a., an extensive survey can be found in [9]

The spectrum of a stationary stochastic process $\underline{s}_i$, $i = -\infty, \ldots, \infty$, denoted by $S(\theta)$, is given by the fourier transform of its autocorrelation function:

$$S(\theta) = \sum_{k=-\infty}^{\infty} R(k)\mathrm{e}^{-\mathrm{i}\theta k}, \; -\pi \leq \theta \leq \pi. \tag{2.3}$$

In Sections 2.2 and 2.3 it will be assumed that this autocorrelation function is known in advance and the general theoretical background of restorations of realizations of stationary stochastic processes will be explained under this assumption. This is first done for the case of finite sequences in Section 2.2 and extended to infinite sequences in Section 2.3. After this general introduction, that should provide a little bit more insight into the theoretical backgrounds, some more practical cases will be studied in the next sections. In Section 2.4 it is assumed that the data sequence can be modeled as an autoregressive process, which in many practical cases gives good restoration results (see for example [9]). Special attention is paid to the case that there are unknown samples on the boundaries of the sequence that need restoration. In Section 2.5 is is assumed that the data sequence is a realization of a band-limited stochastic process. Theoretically this gives a perfect restoration. In practical cases this band-limitedness assumption almost never holds and it was shown in [9] that this method is very sensitive to out-of-band components. Therefore, some adaptations are made that give better restorations in practical cases. Both methods in principle presuppose some knowledge of the autocorrelation function. Under the assumption of band-limitedness only knowledge of the location of the passband is required. This can be related to properties of the autocorrelation function. Under the assumption that the data sequence is a realization of an autoregressive process, the prediction coefficients that describe the autoregressive process need to be known. The prediction coefficients can be computed from the autocorrelation function and vice versa. Since the autocorrelation function can be related to and estimated from the data sequence this leads to an adaptive method. This method first estimates prediction coefficients from the known samples with initial estimates substituted for the unknown samples. Subsequently it estimates the unknown samples from the known samples and the prediction coefficients. This process can then be iterated an arbitrary number of times.

## 2.2 The finite data sequence

Suppose that there is a sequence of samples $s_i$, $i = 1, \ldots, N$ with unknown samples at $t(1), \ldots, t(m)$. Now estimates $\hat{s}_{t(j)}$, $j = 1, \ldots, m$ ($\hat{s}_{t(j)}$ denotes an estimate for $s_{t(j)}$) for the unknown samples are sought that are linear combinations of the known samples

$$\begin{pmatrix} \hat{s}_{t(1)} \\ \vdots \\ \hat{s}_{t(m)} \end{pmatrix} = \begin{pmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,N} \\ h_{2,1} & h_{2,2} & \dots & h_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m,1} & h_{m,2} & \dots & h_{m,N} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}, \tag{2.4}$$

where, defining the index sets $S = \{1, \dots, N\}$, $V = S \backslash \{t(1), \dots, t(m)\}$,

$$v_i = \begin{cases} s_i & \text{if } i \in V \\ 0 & \text{if } i \notin V \end{cases}$$

Coefficients $h_{i,j}$, $i = 1, \dots, m$, $j \in V$ (the coefficients $h_{i,j}$, $i = 1, \dots, m, j \in S \backslash V$, do not influence the estimates) are to be determined. The restoration should be optimized over all possible realizations $s_i$, $i = 1, \dots, \mathbb{N}$, of the stationary stochastic process $\underline{s}_i$, $i = -\infty, \dots, \infty$. One way to do this is to minimize the variance of the statistical restoration error defined by

$$\mathcal{E}\left\{\sum_{i=1}^{m}(\hat{\underline{s}}_{t(i)} - \underline{s}_{t(i)})^2\right\}. \tag{2.5}$$

The $\hat{\underline{s}}_{t(i)}$ are called estimators and follow from (2.4) by replacing $\hat{s}_{t(i)}$ by $\hat{\underline{s}}_{t(i)}$ and $v_i$ by $\begin{cases} \underline{s}_i & \text{if } i \in V, \\ 0 & \text{if } i \notin V \end{cases}$. The particular reason for minimizing (2.5) is that this problem is relatively easy to solve. Moreover, assuming that the samples have a gaussian probability density function, the solution maximizes the log likelihood function

$$L(\mathbf{x}) = \log(p_{\underline{\mathbf{x}}|\underline{\mathbf{v}}}(\mathbf{x}|\mathbf{v})) \tag{2.6}$$

as a function of the unknown samples $\mathbf{x} = [s_{t(1)}, s_{t(2)}, \dots, s_{t(m)}]^{\mathrm{T}}$. Maximizing the log likelihood function is a well known method to obtain estimates in statistics. Writing out (2.4) results in

$$\mathcal{E}\left\{\sum_{i=1}^{m}\left(\left(\sum_{j \in V} h_{i,j}\underline{s}_j - \underline{s}_t(i)\right)\left(\sum_{k \in V} h_{i,k}\underline{s}_k - \underline{s}_t(i)\right)\right)\right\}.$$

This is a quadratic expression in $\mathcal{E}\left\{\underline{s}_j, \underline{s}_k\right\}$, $j \in V$, $k \in V$, and $h_{i,l}$, $i = 1, \dots, m$, $l = 1, \dots, m$. Since stationarity and knowledge of the autocorrelation function $R(\cdot)$ were assumed this can be rewritten to yield

$$\mathcal{E}\left\{\sum_{i=1}^{m}\left(\sum_{j \in V}\sum_{k \in V} h_{i,j}h_{i,k}R(j-k) - 2\sum_{j \in V}\sum_{k \in V} h_{i,j}R(j - t(i)) + \sigma_s^2\right)\right\}. \tag{2.7}$$

with $\sigma_s^2$ the variance of the stochastic process. Because (2.7) is quadratic in $h_{i,l}$, $i = 1,\ldots,m$, $l = 1,\ldots,N$, minimizing this equation, if possible, is accomplished by setting the derivatives with respect to $h_{i,l}$ to zero. A proof that there exists a solution (that is not necessarily unique) may be found in [9]. It is also shown there, that the solution can be written in the form

$$\mathbf{H} = -\tilde{\mathbf{G}}^{-1}\mathbf{G}, \tag{2.8}$$

with $\mathbf{H}$ an $m \times N$ matrix with elements $h_{i,j}$, $i = 1,\ldots,m$, $j = 1,\ldots,N$, $\mathbf{G}$ an $m \times N$ matrix with elements $g_{i,j}$, $i = 1,\ldots,m$, $j = 1,\ldots,N$, and $\tilde{\mathbf{G}}$ an $m \times n$ matrix with elements $\tilde{g}_{i,j}$, $i = 1,\ldots,m$, $j = 1,\ldots,m$. The elements of $\tilde{\mathbf{G}}$ are related to the elements of $\mathbf{G}$ by the relation $\tilde{g}_{i,j} = g_{i,t(j)}$, $i = 1,\ldots,m$, $j = 1,\ldots,m$. The matrix $\mathbf{G}$ can be determined directly from the autocorrelation matrix and the pattern of unknown samples. How this can be done will be discussed later in this section. Since $\tilde{\mathbf{G}}$ can be calculated from $\mathbf{G}$, finding a solution comes down to finding $\mathbf{G}$. The solution is given by

$$\hat{\mathbf{x}} = \mathbf{H}\mathbf{v} = -\tilde{\mathbf{G}}^{-1}\mathbf{G}\mathbf{v}. \tag{2.9}$$

This involves inversion of $\tilde{\mathbf{G}}$ which is computationally inefficient, since it requires $O(m^4)$ multiplications. Therefore (2.9) is written in the form

$$\tilde{\mathbf{G}}\hat{\mathbf{x}} = -\mathbf{G}\mathbf{v} = -\mathbf{z}. \tag{2.10}$$

Now finding the estimates comes down to solving a set of linear equations, which only costs $O(m^3)$ multiplications, and is therefore computationally much more efficient. Only $\mathbf{G}$ remains to be calculated from the autocorrelation function. For this the autocorrelation matrix $\mathbf{R}$ with elements $r_{i,j} = R(i-j), i, j = 1,\ldots,N$, is needed. In [9], two cases are distinguished. The first case is when the rows of the autocorrelation matrix are linearly independent, i.e. $\mathbf{R}$ has full rank. The relation between $\mathbf{R}$, $\mathbf{G}$ and the pattern of unknown samples is then given by

$$\mathbf{R}\mathbf{G}^{\mathrm{T}} = [\mathbf{i}_{t(1)},\ldots,\mathbf{i}_{t(m)}]. \tag{2.11}$$

Here $\mathbf{i}_{t(j)}$ is the $t(j)$th column of the $N \times N$ identity matrix $\mathbf{I}$. This case is called the *regular case*. The other case is called the *singular* case. It applies when there are only $N - m$ or less linearly independent rows of the autocorrelation matrix. In this case it can be proved [9] that a restoration can be made with zero variance of the statistical restoration error, in other words, a perfect restoration can be made. Then $\mathbf{G}$ can be found from $\mathbf{R}$ by finding a non-trivial solution of

$$\mathbf{R}\mathbf{G}^{\mathrm{T}} = \mathbf{0}, \tag{2.12}$$

with $\mathbf{0}$ the all zero $m \times N$ matrix. Non-trivial means that $\mathbf{G}$ must have rank $m$.

## 2.3 The infinite data sequence

For the infinite data sequence, some slight modifications have to be made. For the regular case (2.11), which can also be denoted as

$$\mathbf{R}\mathbf{g}_j = \mathbf{i}_{t(j)}, j = 1, \ldots, m,$$

with $\mathbf{g}_j$ the $j$th column of $\mathbf{G}^{\mathrm{T}}$, becomes

$$\sum_{-\infty}^{\infty} R(k)(g_j)_{l-k} = \delta_{l-t(j)}. \tag{2.13}$$

For the singular case (2.16), rewritten as

$$\mathbf{R}\mathbf{g}_j = \mathbf{0},$$

becomes

$$\sum_{-\infty}^{\infty} R(k)(g_j)_{l-k} = 0. \tag{2.14}$$

For both cases, the elements of the matrix $\mathbf{G}$ become shifted versions of a sequence $g_k$. The elements of $\mathbf{G}$ are related to the sequence $g_k$ by

$$g_{i,j} = g_{j-t(i)}. \tag{2.15}$$

For the regular case the sequence $g_k$ may be computed from

$$g_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1}{S(\theta)} \mathrm{e}^{\mathrm{i}\theta k} \mathrm{d}\theta. \tag{2.16}$$

With the fourier transform $G(\mathrm{e}^{\mathrm{i}\theta})$ of the sequence $g_k$ defined by

$$G(\mathrm{e}^{\mathrm{i}\theta}) = \sum_{k=-\infty}^{\infty} g_k \mathrm{e}^{\mathrm{i}\theta k}, \tag{2.17}$$

this is equal to

$$G(\mathrm{e}^{\mathrm{i}\theta}) = \frac{1}{S(\theta)}.$$

For the singular case a $G(\mathrm{e}^{\mathrm{i}\theta})$ has to be found such that

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} G(\mathrm{e}^{\mathrm{i}\theta}) S(\theta) \mathrm{e}^{\mathrm{i}\theta k} \mathrm{d}\theta = 0. \tag{2.18}$$

The $g_k$ are then given by the inverse transform of (2.17)

$$g_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} G(\mathrm{e}^{\mathrm{i}\theta}) \mathrm{e}^{-\mathrm{i}\theta k} \mathrm{d}\theta, \tag{2.19}$$

## 2.4 The autoregressive model

### 2.4.1 Autoregressive processes

First, the restoration method will be examined under the assumption that the data sequence $s_k$, $k = -\infty, \ldots, \infty$, is a realization of a stationary stochastic process $\underline{s}_k$, $k = -\infty, \ldots, \infty$, that can be modeled as an autoregressive process of finite order. This means that the following equation holds for the stochastic process

$$\sum_{l=0}^{p} a_l \underline{s}_{k-l} = \underline{e}_k, \; k = -\infty, \ldots, \infty. \tag{2.20}$$

Here $p$ is the (finite) order of the autoregressive process, $a_0, a_1, \ldots, a_p$, $a_0 = 1$, are the prediction coefficients and $\underline{e}_k$, $k = -\infty, \ldots, \infty$ a zero-mean white noise process with excitation noise variance $\sigma_e^2$. A stochastic process $\underline{e}_k$, $k = -\infty, \ldots, \infty$ is a zero-mean white noise process with excitation noise variance $\sigma_e^2$ if $\mathcal{E}\left\{\underline{e}_k\right\} = 0$, $k = -\infty, \ldots, \infty$, and $\mathcal{E}\left\{\underline{e}_k \underline{e}_l\right\} = \delta_{k-l}\sigma_e^2$, $k, l = -\infty, \ldots, \infty$. The spectrum of the autoregressive process is given by

$$\begin{aligned} S(\theta) &= \frac{\sigma_e^2}{\left|\sum\limits_{l=0}^{p} a_l \mathrm{e}^{-\mathrm{i}\theta l}\right|^2} \\ &= \frac{\sigma_e^2}{\sum\limits_{l=-p}^{p} b_l \mathrm{e}^{-\mathrm{i}\theta l}} \end{aligned} \tag{2.21}$$

with

$$b_k = \sum_{l=-\infty}^{\infty} a_l a_{l+k}, \tag{2.22}$$

assuming $a_k = 0$ for $k < 0$ and $a_k = 0$ for $k > p$. An autoregressive process can be regarded as the output of an all-pole filter, as exemplified in Figure 2.4.1. This follows from rewriting (2.20) in the following form

$$\underline{s}_k = \underline{e}_k - \sum_{l=1}^{p} a_l \underline{s}_{k-l} =, \; k = -\infty, \ldots, \infty. \tag{2.23}$$

This all-pole filter then has a transfer function

$$\frac{1}{A(\mathrm{e}^{\mathrm{i}\theta})} = \frac{1}{\sum\limits_{l=0}^{p} a_l \mathrm{e}^{\mathrm{i}\theta l}}, \tag{2.24}$$

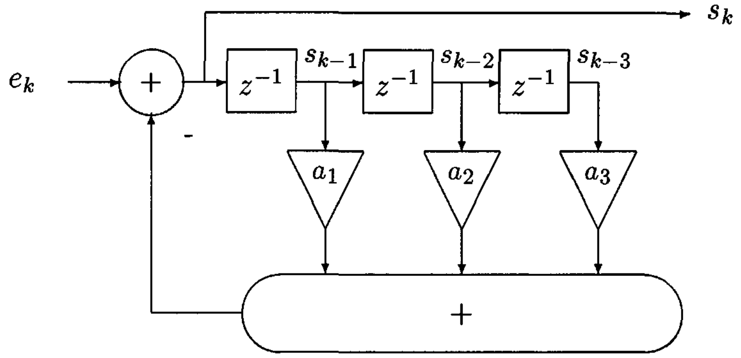

Figure 2.1: Model of an autoregressive process of order 3

and is excited with white noise with variance $\sigma_e^2$. The poles of the filter then are the zeros of

$$A(z) = \sum_{l=0}^{p} a_l z^{-l}, \quad z \in \mathbb{C}. \tag{2.25}$$

The autoregressive filter must be stable, which is equivalent to requiring that the poles of (2.25) are all within the unit circle of the complex plane. First, in Subsection 2.4.2 the restoration method for realizations of autoregressive processes will be examined under the assumption that an infinite data sequence is available. It will be shown that this yields a method that is also valid when only a finite data sequence is available with no unknown samples in the first and the last $p$ samples. It will then be shown that for these cases the solution can also be expressed as the solution of a minimization problem. Until then, it was assumed that the autocorrelation function and thus the autoregressive parameters were completely known. In most practical cases, this is not true. This means that the autoregressive parameters have to be estimated first from the incomplete data sequence. This is discussed in Subsection 2.4.3. This estimation method will also be shown to be related to the minimization problem mentioned before. In Subsection 2.4.4 the results of the previous subsections will be used to make the restoration method adaptive. Then, finally, in Subsection 2.4.5 attention will be paid to the case where there is a finite data sequence available with unknown samples in the first and last $p$ samples. This is of interest since for the signals under consideration these samples are likely to be unknown.

### 2.4.2 Restoration, infinite case

It was already assumed that there would be only a finite number of unknown samples and that their indices are all within the interval

$$p+1 \leq t(1) < t(2) < \ldots < t(m) \leq N-p. \tag{2.26}$$

For reasons that will become clear later, the index of the first unknown sample was chosen to be greater than or equal to $p+1$ ($p$ is the order of prediction) and $N$ was taken so that $N-p$ would be greater than or equal to the index of the last unknown sample. In [9] it was derived that the estimates can then be found by taking

$$g_k = \begin{cases} b_k, & |k| \leq p, \\ 0, & |k| > p, \end{cases} \tag{2.27}$$

with the $b_k$ given by (2.22) and subsequently solving (2.16)

$$\tilde{\mathbf{G}}\hat{\mathbf{x}} = -\mathbf{z}$$

with

$$\tilde{g}_{i,j} = b_{t(j)-t(i)}, \quad i,j = 1,\ldots,m,$$

and

$$z_i = \sum_{k=-p}^{p} b_k v_{k-t(i)}, \quad i = 1,\ldots,m.$$

Since the $g_k$ given in Subsection 2.4.2 constitute a sequence of finite length, it can be seen that, if $p+1$ is smaller than or equal to the index of the first unknown sample and $N-p$ greater than or equal to that of the last unknown sample, only known samples with indices between 1 and $N$ will be used to make a restoration. This means that this method of restoration is also applicable in a situation where a finite segment of an infinite sequence is under consideration, provided that at least the first $p$ samples and the last $p$ samples are known. Now consider minimizing

$$\mathbf{Q}_{p+1}^{N}(\mathbf{a},\mathbf{x}) = \sum_{k=p+1}^{N} \left| \sum_{l=0}^{p} a_l s_{k-l} \right|^2, \tag{2.28}$$

with respect to the unknown samples $\mathbf{x}$. Here $\mathbf{x}$ is the vector of unknown samples

$$\mathbf{x} = \left[s_{t(1)}, s_{t(2)}, \ldots, s_{t(m)}\right]^{\mathrm{T}},$$

and $\mathbf{a}$ the vector of prediction coefficients

$$\mathbf{a} = [a_1, a_2, \ldots, a_p]^{\mathrm{T}}.$$

It can be shown that, under the assumptions stated above, (2.28) can be written as

$$\mathbf{Q}_{p+1}^{N}(\mathbf{a},\mathbf{x}) = \sum_{k=p+1}^{N} (\sum_{l=0}^{p} a_l v_{k-l})^2 + 2\mathbf{x}^{\mathrm{T}}\mathbf{z} + \mathbf{x}^{\mathrm{T}}\tilde{\mathbf{G}}\mathbf{x}. \tag{2.29}$$

Minimizing (2.29) with respect to the unknown samples $\mathbf{x}$ is equal to setting the derivatives with respect to $\mathbf{x}$ to zero. This gives

$$\tilde{\mathbf{G}}\hat{\mathbf{x}} = -\mathbf{z}. \tag{2.30}$$

with the same $\tilde{\mathbf{G}}$ and $\mathbf{z}$ as given for the finite case with the first and last $p$ samples not unknown. Thus minimizing (2.29) will yield the same estimates for $\mathbf{x}$ as the restoration for the infinite case c.q. the finite case with p known samples on either side.

### 2.4.3 Estimation of the AR-parameters

To make the restoration, the autocorrelation function must be known, which, for autoregressive processes is equivalent to knowing the prediction coefficients. These have to be estimated from the data. However, not all the data are known in advance, because a number of samples is unknown. Therefore an initial estimate for those samples has to be made. Usually this initial estimate is made by setting all the unknown samples to zero. This data segment will be used to make a first estimate of the prediction coefficients. The prediction coefficients are then used to make new estimates for the unknown samples. This leads to an iterative estimation procedure of subsequently estimating the unknown samples and the prediction coefficients. To estimate the prediction coefficients, two methods are used: the *autocorrelation method* and the *autocovariance method.*

**Autocorrelation method** Suppose that the autocorrelation function of the stationary stochastic process $\underline{s}_k$, $-\infty < k < \infty$ is known, how this is estimated from the data will be discussed later. Now, since it is assumed that $\underline{s}_k$ is an autoregressive process of order $p$ the following must apply

$$\underline{s}_k = \underline{e}_k - \sum_{i=1}^{p} a_i \underline{s}_{k-i},$$

therefore the following equation is satisfied

$$R(l) = \mathcal{E}\left\{\underline{s}_k \underline{s}_{k-l}\right\} = -\sum_{i=1}^{p} a_i \mathcal{E}\left\{\underline{s}_{k-i}\underline{s}_{k-l}\right\} + \mathcal{E}\left\{\underline{e}_k \underline{s}_{k-l}\right\}.$$

Since the all-pole filter is causal and $\underline{e}_k$ is white noise, $\mathcal{E}\left\{\underline{e}_a\underline{s}_b\right\} = 0$ if $a > b$, and the above equation reduces to

$$R(l) = -\sum_{i=1}^{p} a_i R(l-i),$$

for $l \geq 0$. This can be rewritten as the so-called Yule-Walker equations

$$\begin{pmatrix} R(0) & R(-1) & \dots & R(-p+1) \\ R(1) & R(0) & \dots & R(-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \dots & R(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = - \begin{pmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{pmatrix}. \tag{2.31}$$

Solving (2.31) then gives estimates $\hat{\mathbf{a}}$ for $\mathbf{a}$. To solve (2.31) an estimate for the autocorrelation function lags is needed. In principle this could be done by taking

$$\hat{R}_{\text{unbiased}}(k) = \frac{1}{N-|k|} \sum_{n=1}^{N-|k|} s_{n+|k|} s_n \tag{2.32}$$

as an estimate where $s_1, \dots s_N$ are the data available. It is called unbiased because

$$\mathcal{E}\left\{\underline{\hat{R}}_{\text{unbiased}}(k)\right\} = R(k).$$

Another estimate is

$$\hat{R}_{\text{biased}}(k) = \frac{1}{N} \sum_{n=1}^{N-|k|} s_{n+|k|} s_n. \tag{2.33}$$

It can be seen directly that the expectation values of the estimates for the autocorrelation lags are biased, since

$$\mathcal{E}\left\{\underline{\hat{R}}_{\text{biased}}(k)\right\} = \frac{N-|k|}{N} R(k).$$

This bias is rather small if $k \ll N$. Since in practical cases, only the autocorrelation lags $R(0)$ through $R(p)$ are used and usually $p \ll N$ and thus $k_{\max} \ll N$ this bias can be neglected. The relation between the two estimates for the autocorrelation function can be expressed as

$$\hat{R}_{\text{biased}}(k) = \frac{N-|k|}{N} \hat{R}_{\text{unbiased}}(k).$$

This means that the biased estimate is the unbiased estimate windowed with a triangular window. In practical applications there are two reasons to favor the biased estimates. First, for practical applications the sum of the variance and the squared bias of the tends to be smaller for the biased estimate than for the unbiased estimate [7]. The second reason is that the unbiased estimate may provide invalid autocorrelation sequences. This can be seen as follows. The estimates for the autocorrelation lags can only be valid if $S(\theta) \geq 0$. This can be shown [8] to be equivalent to the requirement that any $N \times N$ autocorrelation matrix is positive semi-definite (a matrix $\mathbf{M}$ is positive semi-definite if, with $\mathbf{u} \neq \mathbf{0}$, $\mathbf{u}^{\mathrm{H}}\mathbf{M}\mathbf{u} > 0$). This automatically provides estimates for the autoregressive coefficients that form a stable and causal autoregressive filter. It can be proven that the positive semi-definiteness necessarily holds for the biased autocorrelation estimates, in contrast to the unbiased autocorrelation estimates, which may form a non-positive-definite estimate for the autocorrelation matrix. For these two reasons the biased autocorrelation estimates are used preferably, rather than the unbiased estimates. It can be proved that for the unbiased estimate, the autoregressive parameters may equally well be found by minimizing a slight modification of (2.28), namely

$$\mathbf{Q}_{-\infty}^{\infty}(\mathbf{a}, \mathbf{x}) = \sum_{k=-\infty}^{\infty} \left| \sum_{l=0}^{p} a_l s_{k-l} \right|^2, \tag{2.34}$$

with respect to the vector of prediction coefficients $\mathbf{a}$. Here the available data segment is extended with zeros on both sides. Note the similarity between (2.28) and (2.34).

**Autocovariance method** Minimizing (2.28), written in a somewhat different notation

$$\mathbf{Q}^{(f)}(\mathbf{a}^{(f)}, \mathbf{x}) = \sum_{k=p+1}^{N} \left| \sum_{l=0}^{p} a_l^{(f)} s_{k-l} \right|^2, \tag{2.35}$$

with respect to the vector of forward prediction coefficients $\mathbf{a}^{(f)}$ then yields the so-called autocovariance equations

$$\begin{pmatrix} C(-1,-1) & C(-2,-1) & \dots & C(-p,-1) \\ C(-1,-2) & C(-2,-2) & \dots & C(-p,-2) \\ \vdots & \vdots & \ddots & \vdots \\ C(-1,-p) & C(-2,-p) & \dots & C(-p,-p) \end{pmatrix} \begin{pmatrix} a_1^{(f)} \\ a_2^{(f)} \\ \vdots \\ a_p^{(f)} \end{pmatrix} = - \begin{pmatrix} C(0,-1) \\ C(0,-2) \\ \vdots \\ C(0,-p) \end{pmatrix}, \tag{2.36}$$

with

$$C(-k,-l) = \sum_{i=p+1}^{N} s_{i-k}s_{i-l}, \quad -k,-l = 0, \dots, p.$$

Note that reversion of the data sequence will yield different prediction coefficients. These are called the backward prediction coefficients. The backward prediction coefficients follow from minimizing

$$\mathbf{Q}^{(b)}(\mathbf{a}^{(b)}, \mathbf{x}) = \sum_{k=1}^{N-p} \left| \sum_{l=0}^{p} a_l^{(b)} s_{k+l} \right|^2, \tag{2.37}$$

with respect to the vector of backward prediction coefficients $\mathbf{a}^{(b)}$. Both methods are widely used for estimating prediction coefficients for applications in spectral estimation [7]. The autocorrelation method, with biased estimates ensures a stable filter. The autocovariance method does not, but has better performance on short data sequences, since it does not extend the data sequence with zeros, therefore reducing the bias caused by that extension.

### 2.4.4 Adaptive restoration

Until now, it was either assumed that the samples were known and the autoregressive parameters had to be estimated or that the autoregressive parameters were known and the samples had to be estimated. In practical cases, however, neither of these will be known. However, estimating both autoregressive parameters $\mathbf{a}$, as well as estimating the unknown samples $\mathbf{x}$, can be formulated as minimizing a function $\mathbf{Q}(\mathbf{a}, \mathbf{x})$ quadratic in both the unknown samples $\mathbf{x}$ and the autoregressive parameters $\mathbf{a}$. Either $\mathbf{Q}(\mathbf{a}, \mathbf{x})$ was chosen (2.35)

$$\mathbf{Q}^{(f)}(\mathbf{a}^{(f)}, \mathbf{x}) = \sum_{k=p+1}^{N} \left| \sum_{l=0}^{p} a_l^{(f)} s_{k-l} \right|^2 ,$$

or (2.37)

$$\mathbf{Q}^{(b)}(\mathbf{a}^{(b)}, \mathbf{x}) = \sum_{k=1}^{N-p} \left| \sum_{l=0}^{p} a_l^{(b)} s_{k+l} \right|^2 ,$$

where it was required that there were at least $p$ known samples on either side or (2.34)

$$\mathbf{Q}_{-\infty}^{\infty}(\mathbf{a}, \mathbf{x}) = \sum_{k=-\infty}^{\infty} \left| \sum_{l=0}^{p} a_l s_{k-l} \right|^2 ,$$

where the data segment was extended with zeros on either side. Since these expressions are of fourth order when minized simultaneously for $\mathbf{a}$ and $\mathbf{x}$ it is a difficult problem because the solution cannot be found analytically. Therefore the following approach is used: An initial estimate $\hat{\mathbf{x}}^{(1)}$ for the unknown samples (usually an all zero vector) is made. $\mathbf{Q}(\mathbf{a}, \hat{\mathbf{x}}^{(1)})$ is minimized with respect to $\mathbf{a}$. This yields an estimate $\hat{\mathbf{a}}^{(1)}$ for the autoregressive parameters. Now $\mathbf{Q}(\hat{\mathbf{a}}^{(1)}, \mathbf{x})$ is minimized with respect to the unknown samples $\mathbf{x}$ which yields a new estimate $\hat{\mathbf{x}}^{(2)}$ for the unknown samples. This procedure may then be iterated until a satisfactory restoration is obtained. Usually convergence is fast and only a few iterations (e.g. 3) are needed [9]. $\mathbf{Q}(\mathbf{a}, \mathbf{x})$ may be used as an indication of the quality the restoration. It can be proven [9] that an unbiased estimate for $\sigma_e^2$ is given by

$$\hat{\sigma}_e^2 = \frac{1}{N-p-m} \mathbf{Q}^{(f)}(\mathbf{a}^{(f)}, \mathbf{x}), \tag{2.38}$$

with $m$ the number of unknown samples, for the case that there are at least $p$ known samples on either side of the data segment. A similar expression can be formulated for $\mathbf{Q}^{(b)}(\mathbf{a}^{(b)}, \mathbf{x})$. Another estimate for $\sigma_e^2$ would be

$$\hat{\sigma}_e^2 = \frac{1}{N+p-m}\mathbf{Q}_{-\infty}^{\infty}(\mathbf{a}, \mathbf{x}), \tag{2.39}$$

Intuitively, it can be understood that if the variance of the excitation noise becomes smaller, the restoration will be better. Another indicator would be the variance of the restoration error. Since this requires inversion of the $m \times m$ matrix $\tilde{\mathbf{G}}^{-1}$ this is not feasible for practical cases with possibly more than 100 unknown samples. It should be noted that convergence of this method is not necessarily to a global minimum.

### 2.4.5 Estimation of samples on boundaries

When minimization of (2.35) is used to find estimates when there are unknown samples within the first $p$ samples of a data sequence, severe problems arise. This can be seen as follows. Assume that the $A^{(f)}(z)$ from (2.24) has all its zeros within the unit circle. Note that this might not necessarily be the case if the prediction coefficients are estimated using the autocovariance method. For conveniency it will be assumed that there is a data segment available $s_i, 1 \leq i \leq N, N \geq p$. Now the variance of the statistical restoration error will be examined for two cases. The first case is when a burst of $m$ samples on the right with indices $i > N$ has to be restored. The second case is when a burst of $m$ samples on the left with indices $i < 1$ has to be restored. For both cases only the limit for $m$ goes to $\infty$ will be examined. It is relatively easy to prove that in the first case the specific restoration for a sample $s_i, i > N$, will be given by

$$s_i = \sum_{l=1}^{p} a^{(f)})_l s_{i-l}.$$

This means that, with $\hat{\underline{s}}_k = \underline{s}_k, k = N-p+1, \ldots, N$,

$$\hat{\underline{s}}_k = \sum_{l=1}^{p} a_l^{(f)} \hat{\underline{s}}_{i-l}.$$

Since it was assumed that $A(z)$ had all zeros within the unit circle it is clear that

$$\lim_{k \to \infty} \hat{\underline{s}}_k = 0.$$

Therefore

$$\lim_{k \to \infty} \mathcal{E}\left\{(\hat{\underline{s}}_k - \underline{s}_k)^2\right\} = \mathcal{E}\left\{\underline{s}_k^2\right\} = R(0).$$

The statistical restoration error per sample for a burst of $m$ samples therefore too, has in the limit that $m$ goes to $\infty$ the value

$$\lim_{m\to\infty} \mathcal{E}\left\{\frac{1}{m}\sum_{i=1}^{m}(\hat{\underline{s}}_{i+N} - \underline{s}_{i+N})^2\right\} = \mathcal{E}\left\{\underline{s}_k^2\right\} = R(0).$$

Thus for a burst of infinite length the interpolation error approaches the signal variance, the same result that was found in [5] for the case of a burst in between known samples. A similar derivation for the burst on the left side yields

$$\lim_{k\to-\infty} \mathcal{E}\left\{(\hat{\underline{s}}_k - \underline{s}_k)^2\right\} = \infty,$$

for the statistical restoration error for the $k$th sample and

$$\lim_{m\to-\infty} \mathcal{E}\left\{\frac{1}{m}\sum_{i=1}^{m}(\hat{\underline{s}}_{1-i} - \underline{s}_{1-i})^2\right\} = \infty,$$

for the statistical restoration error per sample. Both for the limit that $m$ goes to $\infty$. Finding estimates by minimizing (2.35) while assuming that $A^{(b)}(z)$ has all its zeros within the unit circle in the complex plane yields $R(0)$ for restorations on the left and $\infty$ for restorations on the right, i.e. exactly the opposite of the forward prediction method. The correct method should obviously yield restorations that have statistical restoration errors per sample that go to $R(0)$ in the limit $m$ goes to $\infty$ on both sides. When estimates for samples on both boundaries have to be found, these restoration methods therefore do not work correctly. The actual reason for these errors lies in the fact that, if $\tilde{\mathbf{G}}^{(f)}$ denotes the restoration matrix for forward prediction and $\tilde{\mathbf{G}}^{(b)}$ the restoration matrix for backward prediction have the following property. The row $\tilde{\mathbf{G}}^{(f)}_{1,j}, j = 1,\ldots,m$, becomes much smaller than the other rows of $\tilde{\mathbf{G}}^{(f)}$ in the case of unknown samples on the left side, yielding a sparse matrix with at least one eigenvalue close to zero, typical of the order $a_p^{(f)}$. A similar statement can be made for the row $\tilde{\mathbf{G}}^{(b)}_{m,j}, j = 1,\ldots,m$, of $\tilde{\mathbf{G}}^{(b)}$. Since the variance of the relative restoration error can be expressed as [9]

$$E = \frac{\sigma_e^2 \mathrm{trace}(\tilde{\mathbf{G}}^{-1})}{mR(0)}. \tag{2.40}$$

With $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_m$ the eigenvalues of $\tilde{\mathbf{G}}$ it follows that

$$\mathrm{trace}(\tilde{\mathbf{G}}^{-1}) = \sum_{i=1}^{m}\frac{1}{\lambda_i}, \tag{2.41}$$

which will be dominated by the lowest eigenvalues $\lambda$. Since for the case that samples have to be restored on the boundary, one eigenvalue will be close to zero, it follows that the variance of the statistical restoration error per sample will get rather large. This behavior can be improved by using the following restoration matrix

$$\tilde{\mathbf{G}}^{(b+f)} = \tilde{\mathbf{G}}^{(f)} + \tilde{\mathbf{G}}^{(b)}. \tag{2.42}$$

Before proceeding with evaluating this matrix, first the solution, consistent with solving (2.11) for an autoregressive process of known order $p$ will be examined. This involves inverting the autocorrelation matrix. The autocorrelation matrix is Hermitian $(r_{i,j} = r_{j,i}^*)$ and Toeplitz $(r_{i+1,j+1} = r_{i,j})$. In [7] a method is presented for inversion of a Hermitian Toeplitz matrix. Using this it can be found that the inverse of the $N \times N$ autocorrelation matrix $\mathbf{R}$ is given by

$$\begin{aligned}\mathbf{R}^{-1} = \frac{1}{\rho_R} & \begin{pmatrix} 1 & 0 & \dots & 0 \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{N-1} & \dots & a_1 & 1 \end{pmatrix} \begin{pmatrix} 1 & a_1^* & \dots & a_{N-1}^* \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1^* \\ 0 & \dots & 0 & 1 \end{pmatrix} \\ - \frac{1}{\rho_R} & \begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{N-1}^* & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_1^* & \dots & a_{N-1}^* & 0 \end{pmatrix} \begin{pmatrix} 0 & a_{N-1} & \dots & a_1 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{N-1} \\ 0 & \dots & 0 & 0 \end{pmatrix},\end{aligned} \tag{2.43}$$

with $a_1, \dots, a_{N-1}$ given by the solution of

$$\begin{pmatrix} R(0) & R(-1) & \dots & R(-(N-1)) \\ R(1) & R(0) & \dots & R(-(N-1)+1) \\ \vdots & \vdots & \ddots & \vdots \\ R((N-1)) & R((N-1)-1) & \dots & R(0) \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_{N-1} \end{pmatrix} = - \begin{pmatrix} \rho_R \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{2.44}$$

Note that this is a modified form of the (2.31). If it is assumed that $p < N$, then $a_k = 0, k = p+1, \dots, N-1$, and $\rho_R = \sigma_e^2$. Now, since $\mathbf{R}$ is not only Hermitian, but in fact symmetric, since only real-valued data are used, it follows that $a_i^* = a_i$. This yields for the elements $(\mathbf{R})_{i,j}^{-1}$, $i, j = 1, \dots, N+1$ of $\mathbf{R}^{-1}$

$$(\mathbf{R})_{i,j}^{-1} = \frac{1}{\rho_R} \sum_{l=1}^{N} (a_{i-l}a_{j-l} - a_{N+1-i+l}a_{N+1-j+l}), \quad i, j = 1, \dots, N+1. \tag{2.45}$$

The second term is only used when $i \geq N-(p-1) \wedge j \geq N-(p-1)$. A simple calculation shows that (2.45) can be written as

$$(\mathbf{R})_{i,j}^{-1} = \frac{1}{\rho_R} \sum_{l=1}^{N} a_{l-i}a_{l-j} - a_{N+1-l+i}a_{N+1-l+i}, \quad i, j = 1, \dots, N+1. \tag{2.46}$$

In this form the second term is only used when $i \leq p \wedge j \leq p$. If it is demanded that $p < \frac{N}{2}$ then calculating the lower right $(N-p) \times (N-p)$ submatrix (or equivalently the upper left $(N-p) \times (N-p)$ submatrix) of $\mathbf{R}^{-1}$ can be done by just summing the left part of (2.45) (or (2.46)). It can be proven that for the lower right part this can be done by minimizing with respect to $\mathbf{x}$ (2.35)

$$\mathbf{Q}^{(f)}(\mathbf{a}^{(f)}, \mathbf{x}) = \sum_{k=p+1}^{N} \left| \sum_{l=0}^{p} a_l^{(f)} s_{k-l} \right|^2 ,$$

and for the upper left part by minimizing with respect to $\mathbf{x}$ (2.37)

$$\mathbf{Q}^{(b)}(\mathbf{a}^{(b)}, \mathbf{x}) = \sum_{k=1}^{N-p} \left| \sum_{l=0}^{p} a_l^{(b)} s_{k+l} \right|^2 ,$$

with $\mathbf{a}^{(f)} = \mathbf{a}$ and $\mathbf{a}^{(b)} = \mathbf{a}$ The first expression is the forward prediction and the second the backward prediction. If it is furthermore demanded that $p < \frac{N}{3}$ (this is a stronger requirement than the previous condition $p < \frac{N}{2}$), then the upper right and lower left $p \times p$ matrices are identically zero. This also follows correctly from the solution of the minimization problems.

The problem is thus solved, but still some fast indication of the quality of the restoration is needed. Since there is no minimization problem associated with the correct solution, the only indication would follow from inversion of the restoration matrix. Since this is not feasible, this approach is dropped and the minimization problem associated with the restoration matrix of (2.42) is adopted. The $\tilde{\mathbf{G}}^{(b+f)}$ follows from the solution of minimizing

$$\mathbf{Q}^{(f+b)}(\mathbf{a}^f, \mathbf{a}^b, \mathbf{x}) = \mathbf{Q}^{(f)}(\mathbf{a}^{(f)}, \mathbf{x}) + \mathbf{Q}^{(b)}(\mathbf{a}^{(b)}, \mathbf{x}) =$$
$$\sum_{k=p+1}^{N} \left| \sum_{l=0}^{p} a_l^{(f)} s_{k-l} \right|^2 + \sum_{k=1}^{N-p} \left| \sum_{l=0}^{p} a_l^{(b)} s_{k+l} \right|^2 , \tag{2.47}$$

with respect to $\mathbf{x}$. Here $\mathbf{a}^f = [a_1^f, a_2^f, \ldots, a_p^f]^{\mathrm{T}}$ are the forward prediction coefficients as discussed in Paragraph **autocovariance** of Subsection 2.4.3 and $\mathbf{a}^b = [a_1^b, a_2^b, \ldots, a_p^b]^{\mathrm{T}}$ the backward autocovariance prediction coefficients as discussed there. This follows directly from minimizing this (2.47) with respect to the forward and backward prediction coefficients $\mathbf{a}^f$ and $\mathbf{a}^b$. The estimates for $\mathbf{x}$ now follow from minimizing (2.47) with respect to $\mathbf{x}$. Since (2.42) is the sum of the two sparse matrices, but since these are not sparse in the same regions, this matrix is given by (2.42) is not a sparse matrix. The effect that corrupted the restorations that followed from minimizing (2.35) and (2.37) respectively with respect to $\mathbf{x}$ will therefore be suppressed.

**Comparison** In this paragraph a comparison is made between the restoration method that yields estimates that follow from minimizing (2.47) with respect to $\mathbf{x}$ and the method that follows from solving (2.16). Minimizing (2.47) could be written as assuming that

$$(\mathbf{R_Q})^{-1}_{i,j} = \frac{1}{2\rho_R}\left\{\sum_{l=1}^{N} a^f_{l-i}a^f_{l-j} + \sum_{l=1}^{N} a^b_{i-l}a^b_{j-l}\right\},\ i,j = 1,\ldots,N+1.$$

and if for conveniency it is assumed that $\mathbf{a}^f = \mathbf{a}^b = \mathbf{a}$ this is equal to

$$(\mathbf{R_Q})^{-1}_{i,j} = \frac{1}{2\rho_R}\left\{\sum_{l=1}^{N} a_{l-i}a_{l-j} + a_{i-l}a_{j-l}\right\},\ i,j = 1,\ldots,N+1. \tag{2.48}$$

Now adding (2.45) and (2.46) and dividing by 2 yields

$$(\mathbf{R}_{\text{true}})^{-1}_{i,j} = \frac{1}{2\rho_R}\left\{\sum_{l=1}^{N} a_{l-i}a_{l-j} + a_{i-l}a_{j-l}\right\} -$$

$$\frac{1}{2\rho_R}\left\{\sum_{l=1}^{N} a_{N+1-l+i}a_{N+1-l+i} + a_{N+1-i+l}a_{N+1-j+l}\right\},\ i,j = 1,\ldots,N+1. \tag{2.49}$$

This gives $\mathbf{R}^{-1}$ for an autoregressive process with prediction coefficients $a_k$. It can readily be seen that the second term (after the minus sign) vanishes always, except for the cases that $i \le p \wedge j \le p$ or $i > N-p+1 \wedge j > N-p+1$. This means that only the $p \times p$ upper left and lower right submatrices are different. This demonstrates that for the case that there are at least $p$ known samples on either side, both methods are identical, since then only elements that are not within those submatrices of $\mathbf{R}^{-1}$ are used. Since $\mathbf{R}^{-1}$ is persymmetric only the deviations of the upper left $p \times p$ matrix have to be considered. These are given by

$$(\Delta\mathbf{R})^{-1}_{i,j} = (\mathbf{R}_{\text{true}})^{-1}_{i,j} - (\mathbf{R_Q})^{-1}_{i,j} = \frac{1}{2\rho_R}\left\{\sum_{l=1}^{N} a_{l+i}a_{l+i}\right\},\ i,j = 1,\ldots,p. \tag{2.50}$$

Unfortunately these deviations usually do not become small relative to the other components of $\mathbf{R}^{-1}$. Therefore it can be concluded that for the case that there are unknown samples in the boundaries it cannot be guaranteed that the two restoration methods yield similar restorations.

## 2.5 The band-limited model

In this section, a method will be discussed, in which it is assumed that the data sequence is a realization of a band-limited signal $\underline{s}_k$, $k = -\infty, \ldots, \infty$. Band-limited means that the spectrum of the signal, $S(\theta)$ vanishes on a finite subinterval of the fundamental baseband $[0, \pi]$. It can be proven that for a finite number of unknown samples a perfect restoration can be achieved. This however is the case in only two cases. For both cases, the band-limited assumption should hold quite good, since it can be proven that out-of-band components strongly degrade the results. The two cases are discussed in Subsection 2.5.1 and Subsection 2.5.2 and assume the availability of an infinite segment or a finite, but periodic sequence respectively. Both are in practice too sensitive to out-of-band components. The periodicity requirement is, at least for the signals under consideration, artificial. Therefore the method assuming an infinite data sequence is examined more carefully. It can be shown that using this method when only a finite data sequence is available, introduces serious errors, since the finiteness of the data segment can be interpreted as a form of out-of-band noise. Performance can be improved by using windowing techniques. How this is done and what the advantages and disadvantages are is discussed in Subsection 2.5.3. Then the band-limited method is compared to the method based on an autoregressive model in Subsection 2.5.4. This leads to a windowing technique in Subsection 2.5.5 that is slightly different from the one discussed in Subsection 2.5.3.

### 2.5.1 The infinite case

Suppose that $\mathbf{R}_N$ represents the $N \times N$ autocorrelation matrix of a signal with spectrum $S(\theta)$. The elements of $\mathbf{R}_N$ are $\mathbf{R}_N = (R(k-l))_{k,l}, k,l = 0, \ldots, N-1$. Now, since $S(\theta) = \sum_{k=-\infty}^{\infty} R(k)\mathrm{e}^{-\mathrm{i}k\theta}, 0 \leq \theta \leq \pi$. It follows from Szegö's limit theorem that the eigenvalue distribution of $\mathbf{R}_N$ for $N \to \infty$ approaches the value distribution of $S(\theta)$ on the interval $0 \leq \theta \leq \pi$. It was assumed that the spectrum of the signal vanished on some part of the fundamental interval, say on $[a, b]$, with $0 \leq a < b \leq \pi$. This means that the matrix $\mathbf{R}_N$ will have an infinite number of eigenvalues that go to zero, as $N \to \infty$, which means that the theory for the singular case applies. In [9], Appendix A, it is shown that even for signals that are not band-limited in the sense outlined above, but that have a fairly strong attenuation in some part of the fundamental interval, the autocorrelation matrix may be considered almost singular. Since the theory of the singular case can be applied,

$g_k$, $k = -\infty, \ldots, \infty$ must be found that satisfy (2.18):

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} G(\mathrm{e}^{\mathrm{i}\theta})S(\theta)\mathrm{e}^{\mathrm{i}\theta k}\mathrm{d}\theta = 0,$$

and (2.19)

$$g_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} G(\mathrm{e}^{\mathrm{i}\theta})\mathrm{e}^{\mathrm{i}\theta k}\mathrm{d}\theta.$$

A necessary and sufficient condition is that the fourier transform, $G(\mathrm{e}^{\mathrm{i}\theta})$ of the $g_k$ is zero in the regions of the fundamental interval where the spectrum of the signal $S(\theta)$ is non-zero. One way to do this is to take $G(\mathrm{e}^{\mathrm{i}\theta})$ to be an ideal bandpass filter. For the case that the data segment is a realization of a stochastic process band-limited to $[0, \alpha\pi]$, $\alpha < 1$, this ideal bandpass filter has to be a high-pass filter with passband $(\alpha\pi, \pi]$ given by the expression

$$g_k = \delta_k - \frac{\sin(\alpha k\pi)}{k\pi}, \; k = -\infty, \ldots, \infty. \tag{2.51}$$

This is an infinite sequence. Now (2.10)

$$\tilde{\mathbf{G}}\mathbf{x} = -\mathbf{z}$$

has to be solved with $\tilde{g}_{i,j} = g_{t(j)-t(i)}, i, j = 1, \ldots, m$ and $z_i = \sum_{k=-\infty}^{\infty} g_{k-t(i)v_k}$, $i = 1, \ldots, m$. It can be proven [9] that for the singular case a perfect restoration can be made. if for some reasons, the calculated syndrome contains errors, the restorations will contain errors too. Suppose that the calculated syndrome is $\mathbf{z}'$ as opposed to its true value $\mathbf{z}$ then it can be derived [3] that the following relationship holds for the new (erroneous) estimates vector $\hat{\mathbf{x}}$ and the vector of "true" values $\mathbf{x}$:

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\lambda_{\max}}{\lambda_{\min}} \frac{\|\mathbf{z}' - \mathbf{z}\|}{\|\mathbf{z}\|}, \tag{2.52}$$

in which $\lambda_{\max}$ and $\lambda_{\min}$ denote the maximum and minimum eigenvalues of $\tilde{\mathbf{G}}$. Now for the most critical case, a burst of $m$ samples, it can be derived that

$$\lambda_{\min} \cong \mathrm{e}^{-\alpha m\pi}.$$

Furthermore, $\lambda_{\max} \approx 1$. Thus, for an erroneous syndrome, the restoration errors increase exponentially with increasing bandwidth and increasing burst length. The first error that can occur in the syndrome is out-of-band noise (the effect of out-of-band noise was studied in [9]), this includes the noise due to quantization errors. Finally $g_k$ present an infinite filter and the syndrome $\mathbf{z}$ can therefore never be calculated exactly, in practice some finite approximating sequence will be used, introducing an error in the syndrome. In general, this procedure is just a windowing of the $g_k$. This is studied in Subsection 2.5.3.

### 2.5.2 The finite periodic case

Here it is assumed that the signal has a periodicity of N and is band-limited to $[0, \alpha\pi]$ with $\alpha < \frac{N-m}{N}$. These restrictions mean that it can be solved with the finite method for the singular case. The restoration may be found by minimizing the expression

$$\sum_{i=\lceil \alpha N \rceil}^{N} |\mathcal{F}_{\text{data segment}}|^2(i), \tag{2.53}$$

with respect to the unknown samples $\mathbf{x}$. Here $\mathcal{F}_{\text{data segment}}$ the complex discrete fourier transform of the sequence $s_i$, $i = 1, \ldots, N$. It can be shown that (2.53) is of the form

$$\mathbf{s}^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{P}_{\alpha}^{\mathrm{T}}\mathbf{P}_{\alpha}\mathbf{F}\mathbf{s},$$

with $\mathbf{s}$ the data sequence including the unknown samples, $\mathbf{F}$ the complex fourier transform matrix, and $\mathbf{P}_{\alpha}$ the projection matrix that filters out the high-pass components. This can then be written as

$$\mathbf{v}^{\mathrm{T}}\mathbf{A}_{vv}\mathbf{v} + 2\mathbf{x}^{\mathrm{T}}\mathbf{B}_{xv}\mathbf{v} + \mathbf{x}^{\mathrm{T}}\mathbf{C}_{xx}\mathbf{x}, \tag{2.54}$$

with $\mathbf{A}_{vv}$, $\mathbf{B}_{xv}$ and $\mathbf{C}_{xx}$ submatrices of $\mathbf{F}^{\mathrm{T}}\mathbf{P}_{\alpha}^{\mathrm{T}}\mathbf{P}_{\alpha}\mathbf{F}$. Minimizing (2.54) with respect to $\mathbf{x}$ then directly yields the estimates $\hat{\mathbf{x}}$ for the unknown samples, namely

$$\mathbf{x} = \mathbf{C}_{xx}^{-1}\mathbf{B}_{xv}\mathbf{v}.$$

A slightly different representation however, provides a link with the method discussed in the previous subsection. This is done by using the same $g_k$ as described in that subsection and making a infinite set of equations for the case where the same data series including unknown samples is periodic. Although this set of equations cannot be solved directly, it indicates that the method has the same sensitivity to out-of-band noise. It lacks the other two sources of errors, i.e. errors caused by truncation of the filter and errors caused by only being able to evaluate things for a time-limited part of the data segment.

### 2.5.3 Windowing the $g_k$

In this subsection, some results from [1] are presented. There, the case is examined that the data is low-pass band-limited to the interval $[0, \alpha\pi], \alpha < 1$. It can be straightforwardly be derived from the fact that high-pass filtering this signal yields 0, that the estimates for the unknown samples $x_j$ are given by

$$\sum_{j=-\infty}^{\infty} ((\mathbf{H}^{(\alpha)})^{-1})_{i,j}\hat{x}_j = \sum_{k=-\infty}^{\infty} (\mathbf{H}^{(\alpha)})_{i,k}v_k, \tag{2.55}$$

with $\mathbf{H}^{(\alpha)}$ the perfect high-pass matrix, with pass-band $[\alpha\pi, \pi]$.
With $\mathbf{M}^{(\alpha)}$, the perfect low-pass matrix, such that $\mathbf{H}^{(\alpha)} = \mathbf{I} - \mathbf{M}^{(\alpha)}$, it follows that (2.55) may be written as

$$\sum_{j=-\infty}^{\infty} ((\mathbf{I} - \mathbf{M}^{(\alpha)})^{-1})_{i,j}\hat{x}_j = \sum_{k=-\infty}^{\infty} (\mathbf{I} - \mathbf{M}^{(\alpha)})_{i,k}v_k.$$

Note that $(\mathbf{I} - \mathbf{M}^{(\alpha)})_{i,k}, i \neq k$, may be replaced by $(\mathbf{M}^{(\alpha)})_{i,k}, i \neq k$, yielding the expression used in [1]. The elements of $\mathbf{I} - \mathbf{M}^{(\alpha)}$ can be connected with the theory as presented in this report by making the identification $(\mathbf{I} - \mathbf{M}^{(\alpha)})_{i,j} = g_{i-j}, i, j = -\infty, \ldots, \infty$. This proves that, if the $g_k$ from the theory are taken to be the filter coefficients of an ideal low-pass filter that both theories are equivalent. Independently it was shown in [9] and [1] that the error due to out-of-band components is concentrates in the pass-band c.q. is pulse-shaped and is likely to be "amplified" as shown in Subsection 2.5.1. In [1] it is showed that windowing the $g_k$ can improve the performance of the restoration method in the presence of out-of-band components at the cast of a decreased performance in the absence of noise, in particular that it does not yield good restoration for signals band-limited to $[0, \alpha\pi]$, but only for signals band-limited to $[0, \beta\pi]$, with $\beta < \alpha$. In [1] the sequence $g_k$ is replaced by a windowed version, i.e. by $g_k(\gamma) = g_k W(\sqrt{\gamma}k)$, with $W$ a smooth even window function on $\mathbb{R}$, such that $W(x) \rightarrow 0$ as $x \rightarrow \infty$, and $g_k(\gamma) \rightarrow g_k$ as $\gamma \rightarrow 0$. The $\gamma$ is a number close to zero introduced in [1], that is varied to study the influence of the windowing. In [1] good results were obtained for $W(x) = \mathrm{e}^{-x^2}$, and $\gamma$ in the range $[10^{-2}, 10^{-3}]$. The interpretation is that the $\tilde{\mathbf{G}}$, that is $\mathbf{I} - \mathbf{M}^{(\alpha)}$, that follows from the windowed $g_k$, will tend to have eigenvalues that are less close to zero, which is proved for the case of a burst of unknown samples in [1]. This can be easily seen if $\tilde{\mathbf{G}}$ is written down. The stronger the $g_k$ are windowed the more the $\tilde{\mathbf{G}}$ resembles the matrix $g_0\mathbf{I}$, i.e. $\frac{\lambda_{\max}}{\lambda_{\min}} \rightarrow 1$ as $\gamma \rightarrow \infty$. Furthermore, the calculation of the syndrome is easier, since it converges more strongly as $\gamma \rightarrow \infty$, and truncation will not introduce such a large error. Now $\gamma$ is varied to get a satisfactory tradeoff between better results in the presence of out-of-band noise and the performance of restorations of in-band signals.

### 2.5.4 Comparison with the autoregressive model

Given the prediction coefficients $a_0$ through $a_p$ the restoration method for an autoregressive model follows from minimizing (2.34)

$$\mathbf{Q}_{-\infty}^{\infty}(\mathbf{a}, \mathbf{x}) = \sum_{k=-\infty}^{\infty} \left| \sum_{l=0}^{p} a_l s_{k-l} \right|^2,$$

with respect to the vector of unknown samples $\mathbf{x}$. Now, since the data sequence has finite energy, i.e. $\sum_{k=-\infty}^{\infty} |s_k|^2 < \infty$, because only $s_i, i = \dots 1, \dots, N$ is available, then identifying the prediction coefficients with , according to Parseval's inequality

$$\mathbf{Q}_{-\infty}^{\infty}(\mathbf{a},\mathbf{x}) = \sum_{k=-\infty}^{\infty} \left|\sum_{l=0}^{p} a_l s_{k-l}\right|^2 =$$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\sum_{k=0}^{N-1} s_{k+1}\mathrm{e}^{-\mathrm{i}\theta k}\right|^2 |A(\mathrm{e}^{\mathrm{i}\theta})|^2 \mathrm{d}\theta = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\sum_{k=0}^{N-1} s_{k+1}\mathrm{e}^{-\mathrm{i}\theta k}\right|^2 B(\mathrm{e}^{\mathrm{i}\theta})\mathrm{d}\theta, \tag{2.56}$$

with $B(\mathrm{e}^{\mathrm{i}\theta}) = \sum_{l=-p}^{p} b_l \mathrm{e}^{-\mathrm{i}\theta l}$.

Similarly it follows, for the band-limited model, given an infinite data segment, that the restoration method can be written as [2] minimizing

$$\mathbf{Q}_{-\infty}^{\infty}(\mathbf{g},\mathbf{x}) = \sum_{k=-\infty}^{\infty} \left|\sum_{l=-\infty}^{\infty} g_l s_{k-l}\right|^2 =$$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\sum_{k=-\infty}^{\infty} s_{k+1}\mathrm{e}^{-\mathrm{i}\theta k}\right|^2 |G(\mathrm{e}^{\mathrm{i}\theta})|^2 \mathrm{d}\theta = \frac{1}{2\pi}\int_{-\pi}^{\pi} S(\theta) G^2(\mathrm{e}^{\mathrm{i}\theta})\mathrm{d}\theta, \tag{2.57}$$

with respect to the vector of unknown samples $\mathbf{x}$. Here $\mathbf{g}$ is the (infinite) vector of the filter coefficients $g_k$. For an ideal pass-band filter (eigenvalues 1 and 0 only) it follows straightforwardly that $G^2(\mathrm{e}^{\mathrm{i}\theta}) = G(\mathrm{e}^{\mathrm{i}\theta})$. For the band-limited method, it can be proven that in the absence of errors, the restoration method is independent of choosing $G(\mathrm{e}^{\mathrm{i}\theta})$ instead of $G^2(\mathrm{e}^{\mathrm{i}\theta})$. The difference between the method based on an autoregressive filter thus lies in the fact that for the band-limited model it is assumed that $G^2 = G$, while for the autoregressive model $A^2$ is replaced by $B$, with $B = A^2$. It is thus clear that for both methods, a submatrix of a non-negative definite matrix (e.g. $\mathbf{G}$ or $\mathbf{B}$) must be used as the restoration matrix. This means that windowing the $g_k$ must be done such that the fourier transform of the $g_k$ is non-negative definite. The method presented in the following subsection assures that this requirement is met for general windowing functions (for example a simple rectangular window).

### 2.5.5 Squared window

As was noted in the previous subsection, an arbitrary windowing function in general does not yield a sequence $g_k$ with a non-negative definite fourier transform $G$. This can be

[2] This is not a rigorous mathematical proof, since not all steps may be allowed, however it gives a good indication of the line of reasoning.

solved as follows. Instead of using the windowed sequence directly to build the restoration matrix, (2.57) is minimized directly with respect to the unknown samples $\mathbf{x}$:

$$\sum_{k=-\infty}^{\infty} | \sum_{l=-\infty}^{\infty} W(\sqrt{\gamma k}) g_l s_{k-l}|^2. \qquad (2.58)$$

This method is in fact equal to leaving the (windowed) $G^2$ in (2.57) instead of substituting it with (the windowed) $G$.

# Chapter 3

# Pitch determination

## 3.1 Introduction

In this section, pitch determination in speech signals is discussed. In Subsection 3.1.1, the so-called source-filter model for speech, and in particular the linear predictive coding model (LPC) is briefly discussed. Subsequently, in Subsection 3.1.2, the problem of determining the pitch in a sampled audio signal is discussed. The algorithm for the pitch-determination algorithm is discussed there also. Finally another pitch determination algorithm, that will be used for comparison is qualitatively described.

### 3.1.1 Speech model

In speech-analysis, speech is often modeled with a so-called source-filter model. It states that speech can be modeled as a time-varying source that excites a stable time-varying all-pole filter. The source is taken either to be noise, or a periodic source, where the periodicity is called the pitch-period. When this is compared to actual speech, the noise-source may be attributed to the excitation produced by the air being pushed trough a narrow hole in the vocal tract, used to produce consonants like /f/, /p/, /s/ and /t/. This is usually referred to as *unvoiced* speech. The periodic source corresponds to the excitation produced by the vocal chords in vowels and consonants like /j/, /m/, /n/, /w/. This is called *voiced* speech. The filter is then formed by the rest of the vocal tract. This is schematically depicted in Figure 3.1.1 for voiced speech. The source formed by the vocal chords, produces a periodic sound, that for the human voice falls of about 12dB per octave. The periodicity can be recognized by the delta-peaks in the signal. Subsequently, this sound is lead through the acoustical filter formed by the vocal tract. Finally the mouth opening resembles a filter with a filter that falls of circa 6dB per octave as the frequency gets lower. A similar picture can be made for unvoiced speech, where the source is replaced by a (colored) noise source. Often the +6dB/octave effect of the mouth-opening is incorporated in the filter, as well as the -12dB in voiced speech just as the “coloredness” of the noise source in unvoiced

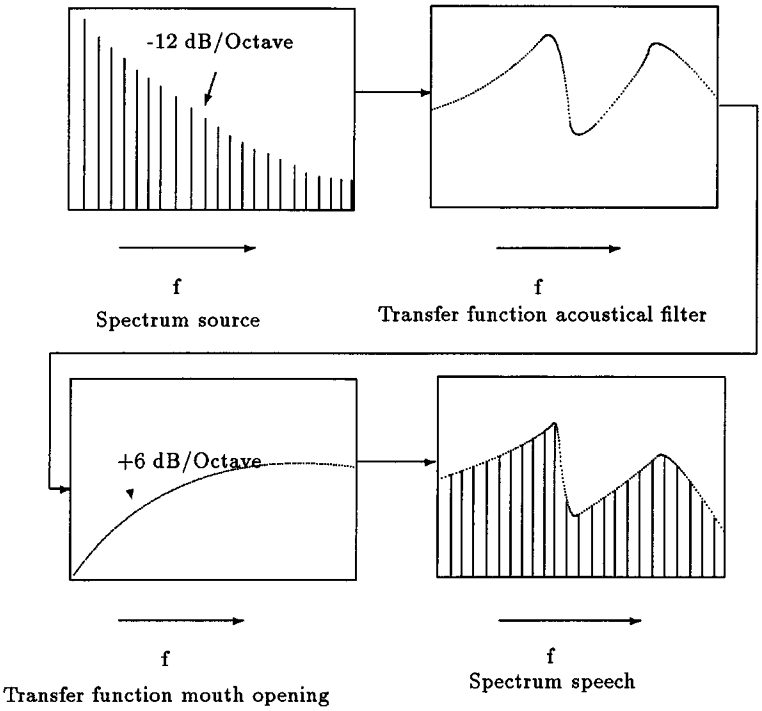
-12 dB/Octave
f
Spectrum source
f
Transfer function acoustical filter
+6 dB/Octave
f
Transfer function mouth opening
f
Spectrum speech

Figure 3.1: source-filter model for voiced speech

speech. This leads to a model that consists of a source that is either white noise or a periodic source with equally strong harmonics that excites an all-pole filter. The filter is often modeled as an autoregressive filter of about order 10. This is called LPC (linear predictive coding) analysis. These can be determined with spectral estimation techniques for determining the autoregressive parameters as were presented in the previous section. Now only the voiced-unvoiced decision and, for the voiced case, the periodicity of the vocal chords remain to be determined. Several algorithms exist [11]. The determination of the periodicity or pitch of the voiced speech is the main interest in this report. In the next subsection, a pitch determination algorithm as presented by Dik Hermes in [4] is explained.

### 3.1.2 Pitch determination algorithm

In this subsection, first some theoretical background of the problem of pitch determination will be explained, followed by a short description of the algorithm.

In nature, many sounds, for example the sound produced by a string, do not consist of a single pure tone, but of a fundamental frequency and higher harmonics, i.e. two, three times the fundamental frequency etc. Not only the fundamental frequency, but also the

Figure 3.2: Illustration of the various stages of the SHS pitch determination algorithm. For simplicity, only subharmonics up to rank 5 are taken into account. The waveform of the signal is shown in (a). In (b), its amplitude spectrum is shown on a linear frequency abcissa. Five main spectral components can be distinguished. In (c) the same amplitude spectrum after peak enhancement $A(f)$ is shown on a logarithmic frequency abcissa, $A(s)$, with $s = \log_2 f$. The spectral window $W(s)$, representing the auditory sensitivity, is presented in (d). The amplitude spectrum after multiplication with this window, $P(s)$, is displayed in (e). The subharmonic summation is shown on the right-hand side of the figure. The harmonically shifted (compressed on linear abcissa) spectra, $h_n P(s + \log_2 n)$, of rank $n = 1, 2, 3, 4, 5$ are shown in (f)-(j). The sum of them, $H(s)$, gives the subharmonic sum spectrum, shown in (k). The maximum of the subharmonic sum spectrum (see arrow) gives the estimate for the pitch.

harmonics contribute to the pitch perception. Even in the absence of the fundamental frequency, i.e. only harmonics are present, the perceived pitch matches the fundamental frequency that corresponds to the harmonics. The concept of subharmonics gives an explanation of this phenomenon. Here it is assumed that there exists an array of frequency-sensitive elements, that are excited not only by the frequency they are most sensitive to, but also by higher frequencies that are twice, three times the most sensitive frequency etc. Higher harmonics contribute less to the formation of the pitch. Viewed in a slightly different way, this statement is equal to assuming that each pure frequency activates not only the frequency sensitive element that corresponds to that frequency, but also the element with half that frequency, one third etc. These are called subharmonics. This is a model for the pitch determination as it is done in the human mind.

The pitch determination algorithm about to be discussed tries to mimic this method. In

Figure 3.1.2 the algorithm is depicted. It will be described stepwise in the remainder of this subsection. The speech segment is sampled at 10 khz and subsequently divided into 40-ms segments. Subsequently to each segment the following procedure is applied:

1. *Low-pass filtering:* This is done by using a running average filter on the the signal. The running average filter $W$ has filtering coefficients $w_k, k = -\infty, \ldots, \infty$, with $w_k = \frac{1}{4}$, for $0 \leq k \leq 3$ and $w_k = 0$ else. Then only each 4th sample is taken, i.e. 75 percent of the samples is thrown away. The signal that is left is thus band-limited to 1250 Hz. This filtering procedure appears not to introduce any errors that seriously affect the pitch determination algorithm [4]. Furthermore, this low-pass filtering implicitly assumes that frequencies above 1250 Hz are not necessary to make a reliable pitch-estimate. For speech signals this is generally the case [4].

2. *Windowing:* The resulting sequence is then multiplied with a Hamming window to diminish distortions in the frequency spectrum caused by truncation at the segment boundaries.

3. *FFT:* First the sequence is padded with zeros to gain 256 points. Subsequently a FFT is performed, yielding the amplitude spectrum. The frequency resolution then is 9.77 Hz on a linear abcissa.

4. *Peak-enhancement:* All points further than 2 points (equal to 19.53 Hz) away from a peak (local maximum) in the amplitude spectrum are set to zero. This can be interpreted as a method to decrease the influence of anharmonic components to the pitch formation [4]. If compared to Figure 3.1.1 it might be seen as a method to filter out the peaked structure of the source that is lost due to the time-windowing. According to [4] this doesn't influence the magnitude or position of the peaks. This spectrum is then smoothed using a 3 point Hanning window, i.e. if $U_n$ represents the points of the unsmoothed peak-enhanced frequency spectrum, then the smoothed spectrum points $A_n$ follow from $A_n = \frac{1}{4}U_{n-1} + \frac{1}{2}U_n + \frac{1}{4}U_{n+1}$.

5. *Logarithmic scale:* Now the spectrum is calculated for a logarithmic scale. It is calculated at 48 points per octave using a cubic-spline interpolation method. This was found to be enough to prevent undersampling of the peaks at higher frequencies [4].

6. *Auditory system:* This logarithmic spectrum $A(f)$, - for convenience $f$, the frequency is taken to be a continuous variable - is multiplied with a raised arc-tangent function,

$W(f)$. The raised arc-tangent function is the amplitude characteristic of a filter that is supposed to represent the sensitivity of the auditory system for frequencies below 1250 Hz. The result is called $P(f)$, i.e. $P(f) = W(f) \cdot A(f)$.

7. *Summation of subharmonics:* To represent the contribution of subharmonics to the pitch perception this spectrum $P(f)$ is shifted along the logarithmic frequency abcissa, multiplied by a weight factor $h_n$ and added. This is clearly seen in Figure 3.1.2. In a formula, with $s = \log_2 f$, this can be represented as

$$H(s) = \sum_{n=1}^{N} h_n P(s + \log_2 n), \tag{3.1}$$

where $n$ numbers the subharmonics. The number $N$, set to 15, is the number of subharmonics that are taken into account. This number, taking into account the cut-off frequency of 1250 Hz corresponds to a fundamental frequency as low as 80 Hz. Voices with higher pitches could do with a lower number [4]. The factor $h_n$ was chosen $0.84^{n-1}$ to represent that higher harmonics contribute less to pitch than lower harmonics do. The spectrum $H(f)$ is called the subharmonic sum spectrum.

8. *Pitch estimation:* The estimation of the pitch is the value $f = 2^s$ for which $H(f)$ is maximum.

It can be seen that this algorithm only considers the peaks in the subharmonic sum spectrum, which correspond to the virtual pitches in [10]. It doesn't incorporate the information in the spectrum $P(f)$, whose peaks are called spectral pitches and are physically present in the original signal. This distinction is of importance in Section 4.4. It is shown in [4] that this algorithm is in qualitative agreement with the principles that are formulated and implemented in [10], from which he derives the numerical weight attributed to the virtual pitch. These principles will repeated here as well as the qualitative agreement of the SHS algorithm with them, as discussed in [4]. The first principle is that there are more spectral components (read harmonics) that contribute to the virtual pitch. This is represented by the summation in this algorithm (SHS). Secondly, the weight of these contribution of each spectral component to the virtual pitch should increase with the weight of a spectral pitch that matches the spectral component. This is done by multiplying the spectrum with a function that represents the sensitivity of the auditory system. Third, the weight of each spectral component should decrease with the harmonic number, which is realized by the decreasing values of $h_n$. The last principle, that if a spectral component has a frequency that deviates from the harmonic structure of the other components, that its weight should decrease. This is roughly done by the peak-enhancement procedure. Quantitatively, however, SHS is a little bit simpler, where the choice of the parameters is concerned, as opposed to [10], where the parameters are adjusted to quantitative psycho-physical experiments. SHS proved to be a good and reliable pitch estimation algorithm, as was shown in [4], for natural speech. Furthermore SHS was tested on telephone speech, i.e. speech high-passed filter with a cut-off frequency of 300 Hz. This high-pass filtering implies that the fundamental frequency and the first harmonic(s) are no longer present in the signal, and there will certainly be no spectral pitch that matches the virtual pitch. Even in this cases SHS proved a reliable algorithm.

Along with this pitch determination algorithm, a voiced-unvoiced decision algorithm was used, whose exact nature isn't important. For every pitch estimate it yields a number between -1 and 1 that represents the probability that the pitch-estimate was for a voiced segment (higher values) or for an unvoiced segment (lower values). In [4] it was chosen to set the threshold to 0.52. Now for every segment of the speech segment a pitch and a voiced unvoiced decision is made. The graphs from the pitch as a function of time are called pitch contours. Usually they consist only of the pitch estimates that where judged voiced. In this report, however, the pitch contour will often be graphed as a whole, i.e. every pitch estimate is displayed.

### 3.1.3 Enhanced pitch determination algorithm

As was outlined in the general introduction, restorations of pitch contours have to be made. As there was no time left to do perceptual experiments, a comparison with another pitch determination algorithm was the only way to compare the restored pitch contours with a more or less "good" pitch contour. This is also needed to make any judgements about the quality of the restored pitch contours, without doing any perceptual experiments. The other pitch determination algorithm, which will only be discussed qualitatively, is generally a modified version of SHS. In the first part, it is identical to SHS, including the summation of subharmonics, i.e. the spectrum $H(f)$ is determined. Instead of choosing the highest peak, the following is done. For each segment in the pitch contours, a number of high peaks in $H(f)$, along with there amplitude in $H(f)$ is stored. Then it is assumed, that if the amplitude is higher, the peak is more likely to be a good peak. Furthermore penalties are introduced that increase with bigger jumps across different segments. Using backtracking techniques, an optimal path is searched for the whole pitch contour, i.e. that with the least jumps and the highest average amplitude (there is a weighing introduced between the penalties for jumps and the amplitude). This then yields a "smooth" contour, that can be compared with the restored contour. This pitch determination algorithm is generally referred to as PDT.

# Chapter 4

# Results

## 4.1 Introduction

In this section, the results will be presented of the attempts to make restorations of pitch contours. First of all, in Section 4.2 a method of judging the quality of restorations is introduced that will be used throughout this chapter. Results of these restorations will be shown for pitch contours of several sentences. The sentences were divided into those that yield "good" pitch contours, using the SHS pitch determination, and those that yield "bad" contours. This division was made by an experienced researcher, who has been involved in speech-perception research for many years. The sentences will be referred to by their codes. The "good" contours were expected to be relatively easy to restore and are listed in Table 4.1. The general structure of this chapter is as follows. First it has to be investigated, using the "good" contours, if it is possible to make satisfactory restorations for these good contours. This is done in Section 4.3. For these "good" contours it is expected that restorations of pitch contours that were determined using the SHS pitch determination algorithm are almost equal to the pitch contours determined using the PDT algorithm. The pitch contours produced by PDT are taken as a reference. To make restorations a division must be made between known and unknown points. As a first approximation the voiced segments were taken to be the known points and and the unvoiced segments were taken to be the unknown points. The number of unknown points could be increased by setting a higher threshold value for the voiced-unvoiced decision, i.e. a number of voiced segments are taken to be unknown and have to be restored. It proved that the method based on an autoregressive model is a method that yields fairly good and reliable restorations. The method using a band-limited model, although it sometimes gave good restorations, proved not reliable enough to yield a restoration method for pitch contours. No further results are presented therefore for the bandlimited method, although in Subsection 4.3.3, a short example is given to demonstrate why it cannot be used to make restorations of pitch contours. Subsequently, the attention is focused on the restoration of the "bad" pitch

Table 4.1: Good contours

| Code | Gender | Sentence |
|---|---|---|
| T16 | Female | Op een dag kwam een vreemdeling het dorp binnenwandelen. |
| T53 | Male | Weet je wie de sleutel gevonden heeft. |
| T59 | Male | U luistert naar de sprekende chip, ontwikkeld door IPO, Natlab en Elcoma |

Table 4.2: Bad contours

| Code | Gender | Sentence |
|---|---|---|
| T6 | Male | Ik was achttien toen er gebeld werd. |
| T14 | Male | Kom nou, je wil toch niet op een hert schieten met hagel. |
| T19 | Female | De laatste weken is er enige vooruitgang merkbaar. |
| T36 | Male | John says he can't come. |

contours, listed in Table 4.2. The main problem proved to be the algorithm that has to decide between the known and the unknown pitch estimates, i.e. those points that are used as the input for the restoration algorithm and those that have to be estimated using the restoration algorithm. The voiced-unvoiced decision that was used to make this division for the "good" contours was not sufficient for the "bad" contours. It was therefore not tried to make an algorithm out of this method. Therefore an algorithm was developed that used different criteria to discriminate between known and unknown points. In Section 4.4, that algorithm is presented. Finally, in Section 4.5, the restorations of all pitch contours made by the SHS pitch determination algorithm are shown. To make the restorations, the restoration method from Chapter 2 and the algorithm presented in Section 4.4 are used. The restored pitch contours are compared to the pitch contours yielded by the PDT pitch determination algorithm.

## 4.2 Quality determination

In this section, a method will be introduced that is supposed to be a measure for the quality of a restored pitch contour. First of all it should be noted, that only a perceptual determination algorithm are compared to those resynthesized with a restored pitch contour can measure the actual quality of a restoration. It can then be decided whether the performance of the restored pitch contours are superior or inferior to the other pitch determination algorithm. Unfortunately, this was not possible because no time was left to do these experiments. Another method is to compare the restored pitch contours to pitch contours restored by an experienced researcher. The latter are almost always pitch contours that consist of straight lines, the so-called close-copy stylization. A close-copy stylization is a pitch contour consisting of the minimal amount of straight lines that, if used to make a resynthesization of the sentence cannot be discriminated from the original. Since the restorations do not consist of a set of lines, close-copy stylizations cannot be fairly compared to restored pitch contours. Therefore it was decided to compare the restored pitch contours numerically to those produced by the PDT pitch determination algorithm. This numerical comparison had to output some number(s) that should indicate the quality of the restoration. The numerical method had to satisfy the following criteria

1. Only those pitch estimates that were restored should contribute to the quality judgement.
2. Only voiced pitch estimates should contribute to the quality judgement.
3. The quality should degrade with increasing difference in pitch.
4. The differences in pitch should be measured using a perceptual scale that is a measure for the ability to hear the difference between two frequencies.
5. It should be possible to compare the quality of two different restored pitch contours in some way.

The first criterion can be easily satisfied. The second criterion is satisfied by using the voiced-unvoiced decision of the PDT pitch determination algorithm. To satisfy the third and the fourth criterion, for a single restored voiced pitch estimate, with index $i$, the following measure is introduced:

$$\mathsf{E}_i = (\mathrm{ERB}(P_{i,\mathrm{restored}}) - \mathrm{ERB}(P_{i,\mathrm{PDT}}))^2, \qquad (4.1)$$

where $P_{i,\mathrm{restored}}$ is the pitch as determined by the restoration method in Hz, $P_{i,\mathrm{PDT}}$ the pitch as determined by the PDT pitch determination algorithm. The ERB() function transforms the linear frequency abcissa in Hz to the so-called ERB scale [2]. The transformation from Hz to ERB can be approximated by [2]

$$[\mathrm{ERB}] = 21.4 \cdot \log_{10}(4.37\mathrm{E}^{-3}[\mathrm{Hz}] + 1). \qquad (4.2)$$

Graphs of (4.2) are given with a linear frequency axis in Figure A.8 and a logarithmic frequency axis in Figure A.9, both in Appendix A. For low frequencies the transformation is almost a linear, while for high frequencies it behaves as a logarithm. The unit of the ERB-scale is the ERB. In the ERB scale, the difference of two pitches is a measure for the ability to hear the difference in pitch. Therefore, to satisfy the third criterion, the difference from the pitch of the restored contour and the pitch as estimated by the PDT pitch determination algorithm is taken in the ERB scale. Because the measure should increase with pitch-difference, the square from this is taken, to satisfy the fourth criterion. The error for a single restored voiced pitch estimate is thus measured in ERB$^2$. To satisfy all criteria, including the last, finally two measures of quality are introduced. The first measure is given by

$$\mathsf{E} = \frac{1}{h} \sum_{i \in Y} \mathsf{E}_i, \tag{4.3}$$

where $Y$ is the set of voiced restored pitch estimates in the contour, and $h$ is the number of voiced restored pitch estimates in the contour. To get an idea of how the errors are spread, the following quantity is introduced:

$$\sigma(\mathsf{E}) = \sqrt{\frac{1}{h-1} \sum_{i \in Y} (\mathsf{E}_i - \mathsf{E})^2}, \tag{4.4}$$

with $Y$ and $h$ as in (4.3).

The other measure of quality, $\mathsf{E}_{\max}$, is given by the description that $\mathsf{E}_{\max} = \mathsf{E}_j$, with $j$ fixed and $j \in Y$, such that there is no $i \in Y$, such that $\mathsf{E}_i > \mathsf{E}_j$. Although the author realizes that these are not the only possible measures for quality and that no conclusion can be drawn from absolute numbers that result from these quality measures. It is expected, however, that it gives at least an indication as to how good a restored contour is as compared to a restored contour, which was restored using a different restoration method. It should be noted that these measures can only be used to say that a restoration of a pitch contour is bad (i.e. high value which corresponds to a large distance). It is not fit to compare two pitch contours that are not bad (i.e. both have rather small values, i.e. lie closely together).

## 4.3 Feasibility

### 4.3.1 Introduction

Only the pitch contours from Table 4.1 were used to examine the feasibility of the restorations. The results are presented in the form of graphs and tables. Graphs are only presented for the sentence T16. The graphs for T53 and T59 show similar features. First some pitch contours are shown.

**SHS pitch contour:** The pitch contour for T16, determined using the SHS pitch determination algorithm is shown in Figure 4.1. This incorporates the pitch estimates for the unvoiced segments. Most of the "jumpy parts" of this figure can be attributed to pitch estimates for the unvoiced segments.

**Voiced segments of SHS pitch contour:** In Figure 4.2, only the pitch estimates for the voiced (as judged by the SHS algorithm) segments of Figure 4.1 are shown.

**Correct voiced segments of SHS pitch contour:** Later on in this section, trial restorations are made from the pitch contour of T16. Therefore a discrimination between known and unknown points has to made. A first approximation is to use the voiced pitch estimates as known points and the unvoiced parts as unknown points. It can be seen in Figure 4.2, that this still leaves some known points in the pitch contour which, if judged by a human would have been classified as unknown. Unfortunately, leaving this points in the list of known points affects the restoration results in a negative way, i.e. the restoration results become poorer.

To be able to classify these points as unknown, the following approach was followed. As was outlined in the previous chapter, the voiced-unvoiced decision was based on a number that varied between -1 (probably unvoiced) and 1 (probably voiced). The voiced-unvoiced decision then depended on whether that value was below the threshold value of 0.52 (unvoiced) or above it (voiced). Now the threshold value was increased, until the pitch contour did not contain any pitch estimates for voiced segments, that were wrong (as judged by the author). This then results in the pitch contour as shown in Figure 4.3. The benefits from this are twofold.

- First of all, there are no points that are classified known unjustified, that could make the restoration results worse.

- Secondly, some voiced segments are classified as unknown and have to be interpolated. This has the advantage that one can judge the interpolation results for voiced segments, using the measures of quality introduced in Section 4.2. This is of importance, since interpolation of unvoiced segments do not contribute to a better pitch contour since they are perceptually not relevant.

It should be noted that the voiced pitch estimates in Figure 4.2 that lie around 50 Hz are produced by some audible 50 Hz noise-source. This is thus not a failure of the SHS pitch determination algorithm, which produces a correct pitch estimate for these segments. Yet, they would have been classified as unknown by an experienced human and therefore have to be removed before restorations can be made. This was accomplished by the procedure of adjusting the threshold for the voiced-unvoiced decision.

**PDT pitch contour:** For comparison, a pitch contour using the PDT pitch determination algorithm was made. This is shown in Figure 4.4. As can be seen, this yields a smoother pitch contour, which corresponds more closely to what a human listener would expect. On the end of the pitch contour a sudden downward movement of the pitch contour can be seen. This is an artifact, caused by the low-frequency 50 Hz noise source mentioned before. In practice this does not bother, since this part is judged unvoiced, as can be seen from Figure 4.5, where the pitch estimates for the voiced segments of the pitch contour from Figure 4.4 are shown.

In Subsection 4.3.2 the feasibility of restoring the pitch contours as produced by the SHS pitch determination algorithm using an autoregressive model for the pitch contour will be studied. Then the feasibility of restoring the pitch contours using a band-limited model will be shortly discussed in Subsection 4.3.3. Finally in Subsection 4.3.4 it will be explained what model will be used to make the restorations.

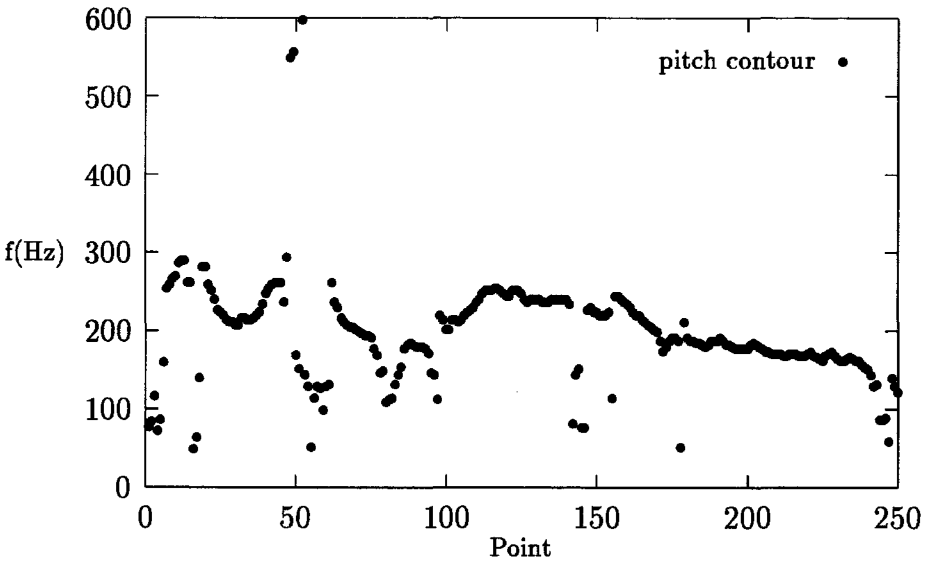

Figure 4.1: Graph of the pitch contour of the sentence *T16* as determined by the *SHS* pitch determination algorithm.

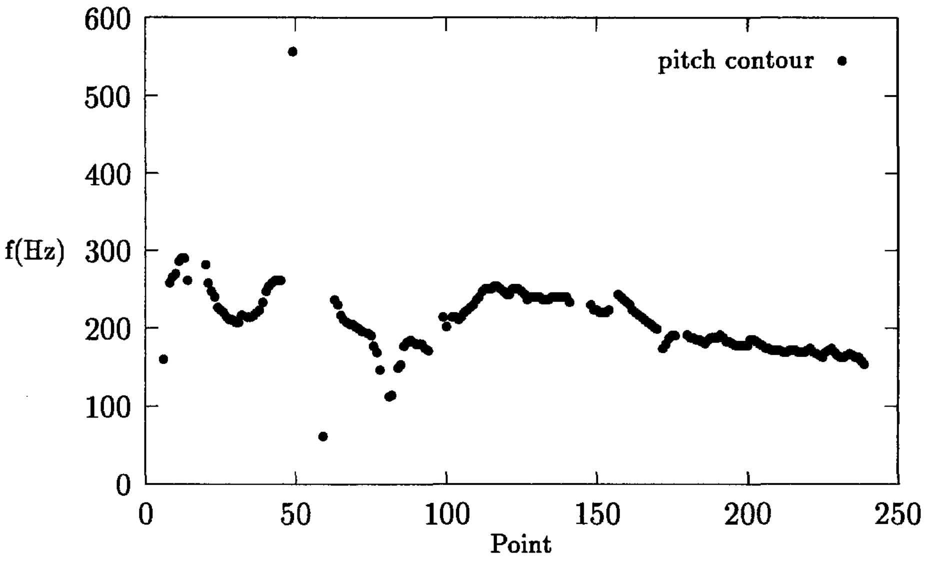

Figure 4.2: Graph of the pitch contour of the sentence *T16* as determined by the *SHS* pitch determination algorithm, only the pitch estimates that were qualified as *voiced* are shown. These constitute 75 percent of the total pitch contour.

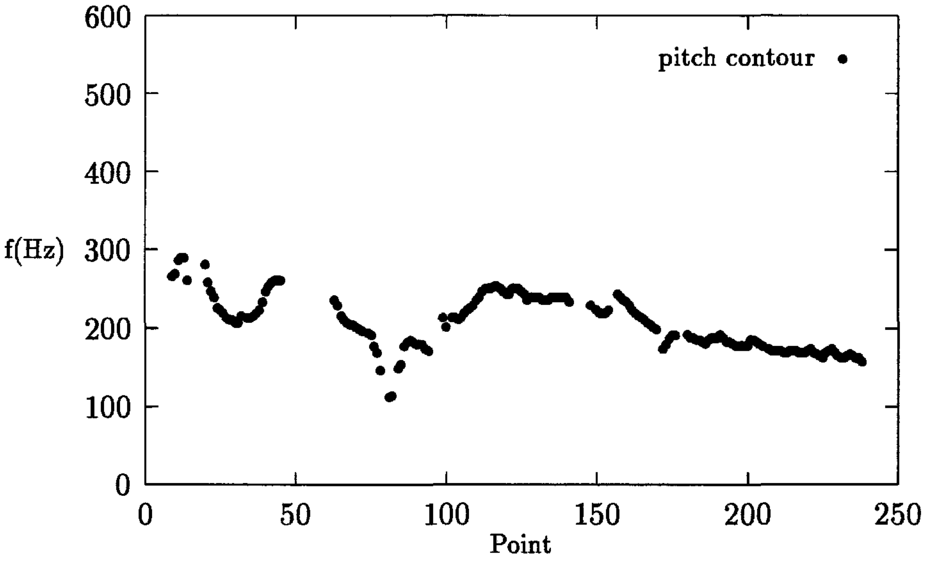

Figure 4.3: Graph of the pitch contour of the sentence *T16* as determined by the *SHS* pitch determination algorithm. The voiced-unvoiced decision was set to a ***higher threshold*** until a satisfactory pitch contour was yielded. Now 72 percent of the total pitch contour remains.

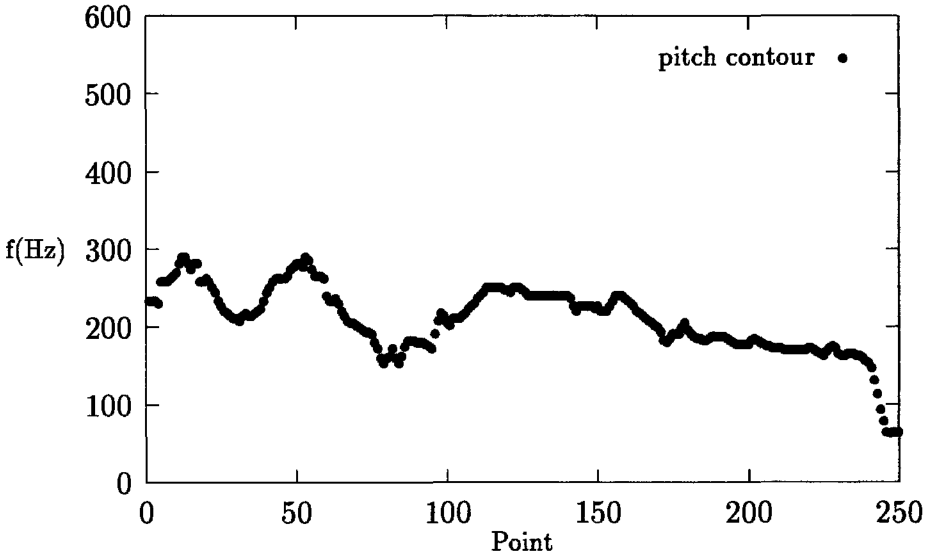

Figure 4.4: Graph of the pitch contour of the sentence *T16* as determined by the *PDT* pitch determination algorithm.

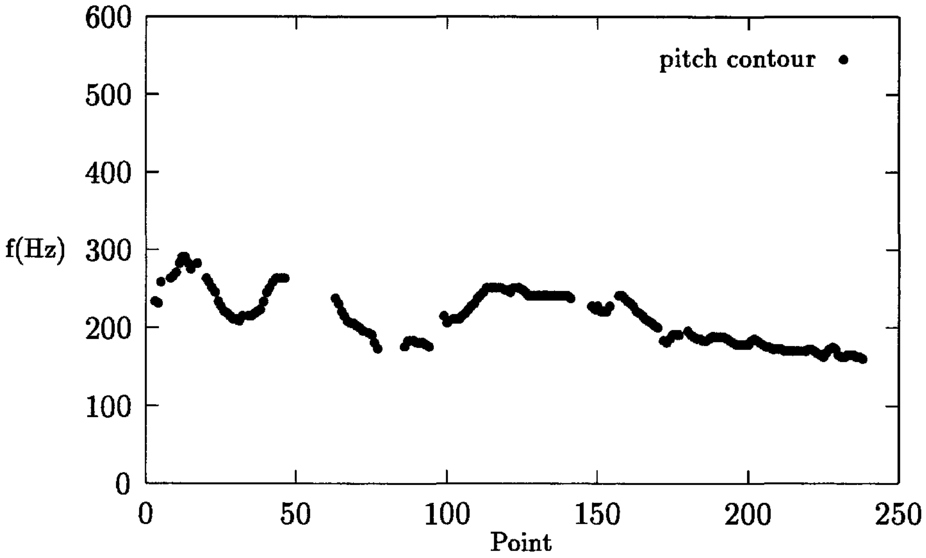

Figure 4.5: Graph of the pitch contour of the sentence *T16* as determined by the *PDT* pitch determination algorithm, only the pitch estimates that were qualified as voiced are shown. These constitute 74 percent of the total pitch contour.

### 4.3.2 Autoregressive model

First of all, it should be explained what is meant by the *autocorrelation method*, the *correct autocorrelation method* and the *autocovariance method*. All these are iterative restoration methods that iteratively estimate prediction coefficients from the incomplete data alternated by estimating the unknown samples from the prediction coefficients and the incomplete data.

With the *autocorrelation method* a restoration method is meant, where

1. the prediction coefficients are calculated using the autocorrelation method described in the paragraph **Autocorrelation method** of Subsection 2.4.3,

2. the restoration is made by taking the forward and backward prediction coefficients equal to the prediction coefficients just calculated and subsequently minimizing (2.47) with respect to the unknown samples.

With the *correct autocorrelation method* a restoration method is meant, where

1. the prediction coefficients are estimated as for *autocorrelation method*,

2. restoration is accomplished by inversion of $\mathbf{R}$, as described in Subsection 2.4.5.

With the *autocovariance method* a restoration method is meant, where

1. The forward and backward prediction coefficients are calculated using the autocovariance method described in the paragraph **Autocovariance method** of Subsection 2.4.3,

2. The restoration is made by taking the forward and backward prediction and minimizing (2.47) with respect to the unknown samples.

Most of the time, the las method, i.e. the *correct autocorrelation method* will not be used, since it will be shown that the restoration results are hardly different than those for the *autocorrelation method*. To make a restoration, using a restoration method derived from the autoregressive model, one must choose the order of prediction. The prediction coefficients do not need to be chosen in advance, since they can be estimated from the data, using the iterative restoration method. The number of iterations has to be chosen too. In Subsubsection 4.3.2.1 and Subsubsection 4.3.2.2, it is argued why the prediction order will be chosen 10 and the number of iterations will be chosen 3 for making restorations of pitch contours. For these values of order of prediction and number of iterations. Polar plots of the zeros of the autoregressive filter are shown in Subsubsection 4.3.2.3. In [9] it was argued that the closer these zeros are to the unit-circle in the complex plane, the better the restorations will be. Finally Subsubsection 4.3.2.4 compares the different methods with each other using the quality measure introduced in Section 4.2 and the PDT pitch determination method as the reference. Furthermore it examines the effects of making the restorations in the ERB or the Hz scale, the effects of reducing the number of known samples in the pitch contour and once more the effect of varying the number of iterations and the order of prediction.

#### 4.3.2.1 Order of prediction

Several methods exist in the literature, to choose the order of prediction. For example, for the restoration of audio signals, in [9], the order of prediction was set to $p \cong 3m$, with $p$ the order of prediction and $m$ the number of unknown samples that have to be estimated, with a maximum for $p$ of 50.

In this report the following approach is taken. The estimation methods for the prediction coefficients, i.e. the *autocorrelation method* and the *autocovariance method* can be formulated as the minimization of a quadratic expression in both samples and prediction coefficients. In this report, this quadratic expression is usually referred to as **Q**, see for Chapter 2. One now takes a typical *complete* pitch contour, i.e. without unknown points. This may equally well be a restored pitch contour, for which after the restoration all points are chosen to be known. Subsequently the prediction coefficients for this pitch contour are estimated for a number of different prediction orders. In this report this is done for prediction orders between 1 and 100. For each order this then yields a **Q**. The **Q** should be a monotonic non-increasing function of the order of prediction. If a graph of **Q** against the order of prediction then shows for example a sudden drop of **Q** at a certain order, this could indicate that this contour might be modeled as an autoregressive process of approximately that order.

To make a more direct comparison between the *autocorrelation method* and the *autocovariance method* of estimating the prediction coefficients, not a graph of **Q**, but rather a graph of $\hat{\sigma}_e^2$ against the order of prediction is made. The expression $\hat{\sigma}_e^2$ will also be referred to as the estimate for the variance of the excitation noise. This can be related to **Q** using (2.38)

$$\hat{\sigma}_e^2 = \frac{1}{N-p-m}\mathbf{Q}^{(f)}(\mathbf{a}^{(f)}, \mathbf{x}),$$

for the *autocorrelation method*, or (2.39)

$$\hat{\sigma}_e^2 = \frac{1}{N+p-m}\mathbf{Q}_{-\infty}^{\infty}(\mathbf{a}, \mathbf{x}),$$

,for the *autocovariance method*. Furthermore, it is examined whether previously subtracting the declination has advantages, i.e. reduces $\hat{\sigma}_e^2$. This would then enable to get the same performance at a lower prediction order, which is computationally more efficient. For a number of pitch contours, the following methods of estimating the prediction coefficients are used to give estimates for $\hat{\sigma}_e^2$:

1. *Autocorrelation method*, declination not subtracted.
2. *Autocorrelation method*, declination subtracted.
3. *Autocovariance method*, declination subtracted.
4. *Autocovariance method*, declination subtracted.

This is applied to the original pitch contours as produced by the SHS and PDT algorithms first, and thereafter this is done for reconstructed pitch contours that were produced by the SHS and PDT algorithm.

**Original:** In Figure 4.6, this graph is shown, using the pitch contour of Figure 4.1. This pitch contour still contains a lot of noise in the form of pitch-estimates for unvoiced segments. Therefore it is likely that the restoration results for this pitch contour are a lot poorer than for a pitch contour without these errors, for example as produced by the PDT pitch determination algorithm. These erroneous pitch estimates are expected to give a higher estimate for the variance of the excitation noise, i.e. that it is harder to model it as an autoregressive process. It is likely that a restored pitch contour would look like a pitch contour as produced by the PDT pitch determination algorithm. Therefore the graph from Figure 4.6 should be compared to a similar graph made for the pitch contour as determined by the PDT algorithm. This graph is depicted in Figure 4.7.

**Reconstructed:** These graphs can be compared to Figure 4.8 and Figure 4.9. These are the graphs for reconstructed pitch contours. Three iterations where used to make the reconstructed pitch contours that were the basis for both graphs. In Figure 4.8, the pitch contour, reconstructed from Figure 4.3 was used, while for Figure 4.9 the pitch contour reconstructed from Figure 4.5 was used. The scales of Figures 4.7 through 4.9 are such that the graph for the case that the *autocorrelation method* is applied without previously subtracting the declination, does not lie in the displayed part[1].

**Discussion:** When Figures 4.6 through 4.9 are compared, the following things can be remarked. First of all, Figure 4.6 has the expected larger estimates for the variance of the excitation noise. This can be attributed to modeling the pitch estimates for unvoiced segments. Since the objective was to model the voiced pitch-estimates, this means, that Figure 4.6 can not be used to get an impression as to how well pitch contours can be modeled as an autoregressive process. This leaves us with the other three graphs.

It is striking that, for the *autocorrelation method* of estimating the prediction coefficients, previously subtracting the declination from the pitch contour, yields a significantly lower estimate for the variance of the excitation noise, as opposed to the case where the declination is not previously subtracted. This can not be observed for the *autocovariance method* of estimating the prediction coefficients. A relatively simple explanation exists for this phenomenon. In order to determine the prediction coefficients for the *autocorrelation method*, the data segment (here the pitch contour) is padded with zeros. Subsequently, the prediction coefficients are estimated for the combination of the data segment and the padded zeros. Usually there is a transition from the data segment to the zeros. A sharper transition from the data segment to the zeros is harder to model, which reflects itself in a higher estimate for the variance of the excitation noise. Usually there is quite a sharp transition from the zeros to the data segment, unless the declination is removed. Therefore the estimate for the variance of the excitation noise is higher if the declination is not removed. Since the autocovariance method only uses the available data segment, this effect is not present there. Sometimes a (much smaller) effect can be seen that is due to the fact that subtracting the declination already partly models the data. With the same order of prediction, this leaves more prediction coefficients to model the behavior of the data, yielding a better model and thus a lower estimate for the variance of the excitation noise.

Furthermore, it can be observed that except for maybe the transition from order 1 to 2, the estimate for the variance of the excitation noise, hardly gets lower with increasing order. This indicates that even a very low order autoregressive process describes a pitch contour very well. It is therefore expected that a prediction order of 10 would be more than sufficient to model the pitch contour. It is noted again, that for the sentences T53 and T59, similar results are gained. The final choice for the order of prediction will therefore be 10.

[1] In Appendix A, these graphs, on another scale so that all the curves are within the displayed part can be seen in Figures A.1 through A.3 respectively.

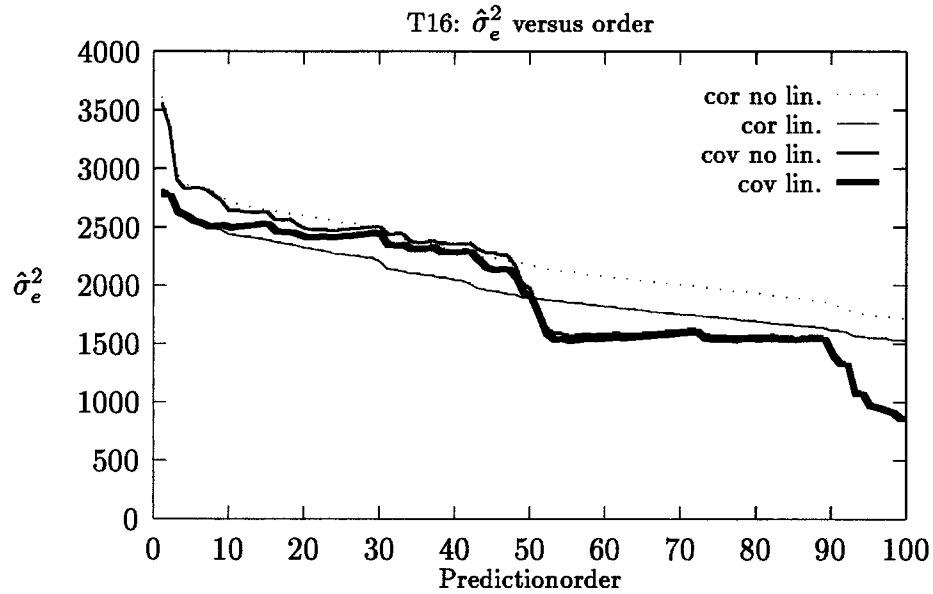

Figure 4.6: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the order of prediction for the case, where the autoregressive parameters are determined using (cor) the ***autocorrelation method***, and (cov) the ***autocovariance method.*** This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence ***T16***, using the ***SHS*** pitch determination method. Note that the vertical from this graph is 40 times that of the graphs that follow.

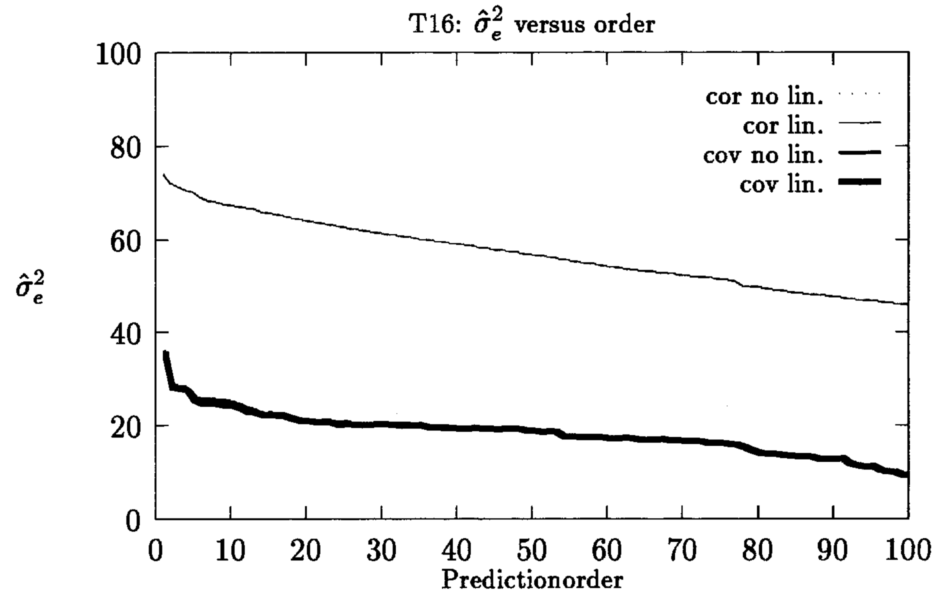

Figure 4.7: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the order of prediction for the case, where the autoregressive parameters are determined using (cor) the *autocorrelation method*, and (cov) the *autocovariance method*. This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence *T16*, using the *PDT* pitch determination method. The lines (lin.) and (no lin.) for (cov) almost coincide and can therefore not be distinguished. The line for (cor no lin.) is not within the displayed region.

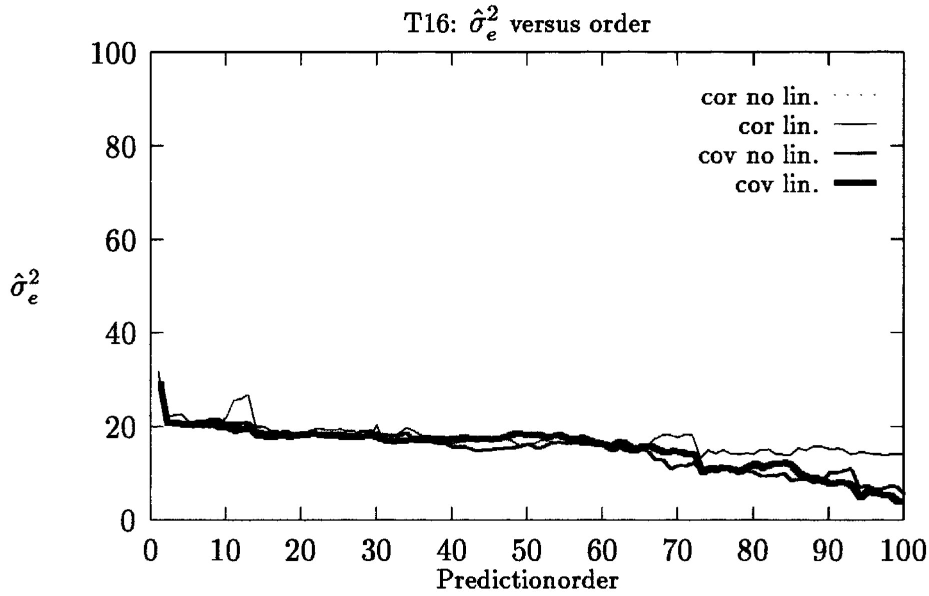

Figure 4.8: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the order of prediction for the case, where the autoregressive parameters are determined using (cor) the ***autocorrelation method***, and (cov) the ***autocovariance method***. This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence ***T16***, using the ***SHS*** pitch determination method. Only the pitch estimates shown in Figure 4.3 are used. From this a **restoration** was made, using three iterations. The line for (cor no lin.) is not within the displayed region.

#### 4.3.2.2 Number of iterations

The other parameter that has to be determined is the number of iterations. Here, like in choosing the order of prediction, the estimate for the variance of the excitation noise is studied, but now as a function of the number of iterations. In Figure 4.10, restorations, from the pitch contour of Figure 4.3, using prediction order ten where made, while for Figure 4.11 the restorations where made from Figure 4.5. The scales of Figures 4.10 and 4.11 are such that the graph, for the case that the *autocorrelation method* is applied without previously subtracting the declination, does not lie in the displayed part[2]. When Figures 4.10 and 4.11 are compared, the following things can be remarked. The first, most important remark is that there is no need to do more than three iterations, possibly often two iterations will suffice. Furthermore it can be seen that for the *autocovariance method*, the estimate for the variance of the excitation noise decreases monotonically, as it should, since the restoration method can be rewritten as a method that iteratively minimizes the estimate for the variance for the excitation noise as a function of the prediction coefficients and the unknown samples. For the *autocorrelation method*, however, such a relation only exist if there are at least $p$ known samples on either side of the data segment. This is not the case here. The performance for the *autocovariance method* of estimating the prediction

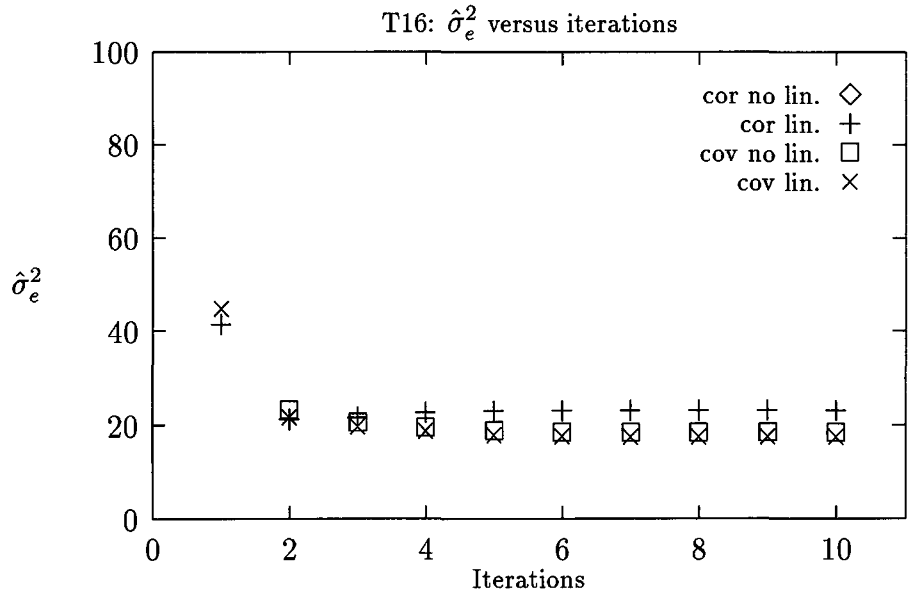

Figure 4.10: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the number of iterations made in restoring the pitch contour. This is done for case, where the autoregressive parameters are determined using (cor) the *autocorrelation method*, and (cov) the *autocovariance method.* This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence *T16*, using the *SHS* pitch determination method. Only the pitch estimates shown in Figure 4.3 are used. From this a **restoration** was made, using a prediction order of ten. The hill in the graph for (cor. lin.) is due to the fact that the restoration method can not be written as an iterative minimization of one quadratic expression.

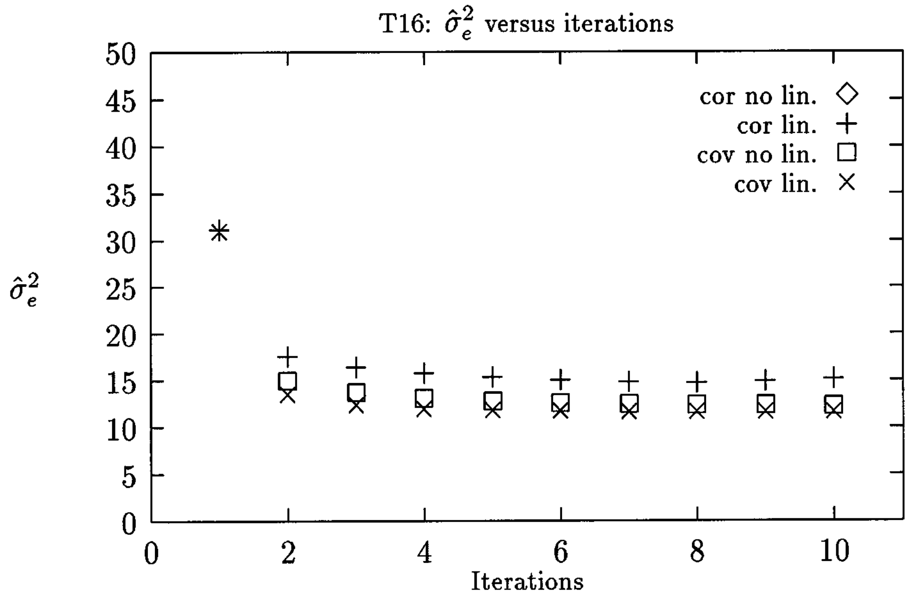

Figure 4.11: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the number of iterations made in restoring the pitch contour. This is done for case, where the autoregressive parameters are determined using (cor) the *autocorrelation method*, and (cov) the *autocovariance method.* This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence *T16*, using the *PDT* pitch determination method. Only the pitch estimates shown in Figure 4.5 are used. From this a **restoration** was made, using a prediction order of ten.

#### 4.3.2.3 Plots of the zeros of the autoregressive filter

In this paragraph the plots of the zeros of the autoregressive filter will be shown, i.e. the complex zeros of the polynome

$$A(z) = \sum_{k=0}^{p} a_k z^k.$$

The autoregressive filter is calculated for restored pitch contours of the sentence T16. The pitch contours were determined using the PDT pitch determination algorithm. Fifty percent of the voiced pitch estimates were restored, using order of prediction ten and three iterations. The fifty percent was achieved by raising the threshold value to 0.92 (for T16). These graphs of the zeros of the autoregressive filter are shown for the *autocorrelation method* in Figure 4.12 and Figure 4.13. The former for the case that the declination was not removed, the second for the case that the declination removed. For the *autocovariance method* these graphs are shown in Figure 4.14 and Figure 4.15 respectively. Only the zeros for the autoregressive filter formed from the forward prediction coefficients is shown for this method. It can be seen, that for the *autocovariance method* the zeros tend to be a little bit closer to the unit circle, which would indicate a possibly better restoration [9]. Removing the declination also yields zeros a little bit closer to the unit circle, again indicating a possibly better restoration. If all the graphs from this and the two previous paragraphs are compared, it indicates that the *autocovariance method* performs better than the *autocorrelation method* and that removing the declination yields better results than not doing so.

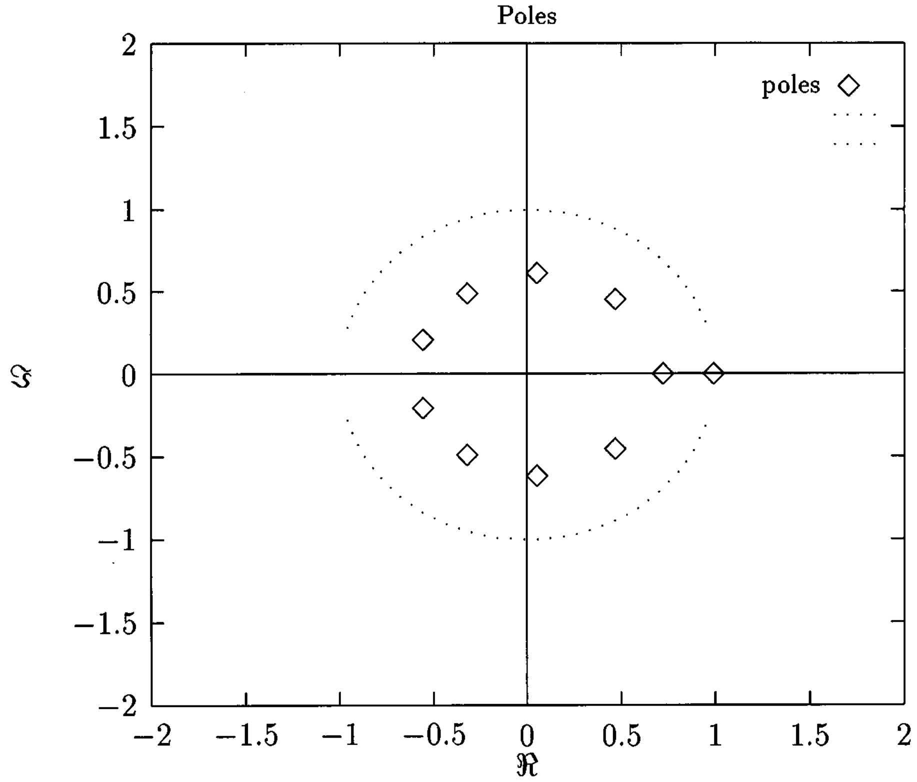

Figure 4.12: Zeros of the autoregressive filter after **restoration** of the pitch contour of the sentence *T16* from fifty percent of the voiced segments. The *autocorrelation method* without removing the declination was used both for the reconstruction and determining the autoregressive filter.

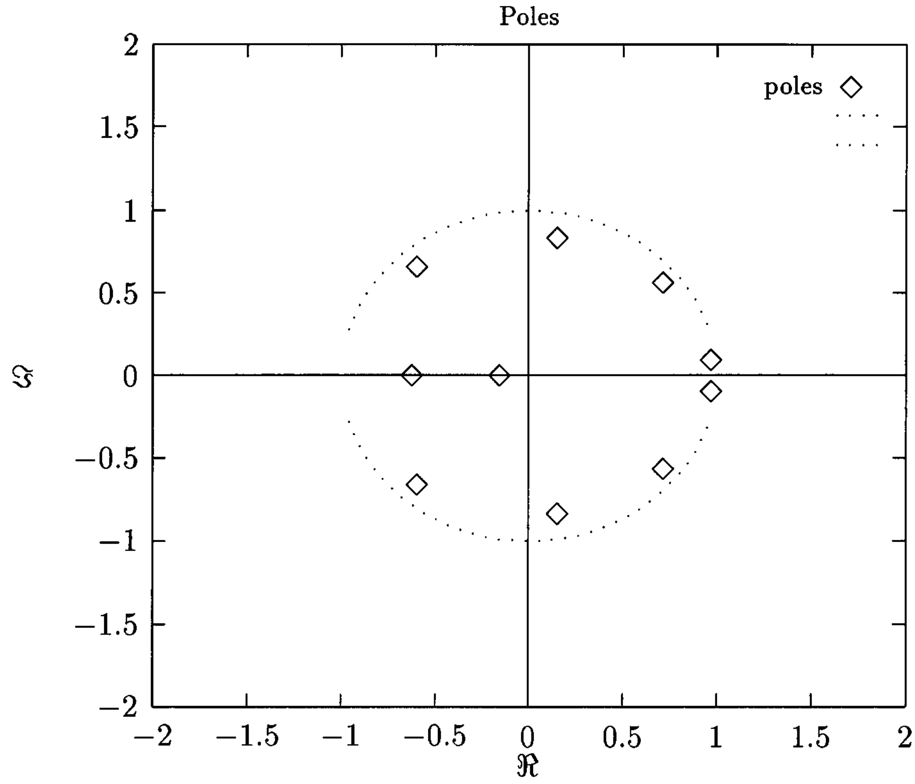

Figure 4.13: Zeros of the autoregressive filter after **restoration** of the pitch contour of the sentence *T16* from fifty percent of the voiced segments. The *autocorrelation method* with removal of the the declination was used both for the reconstruction and determining the autoregressive filter.

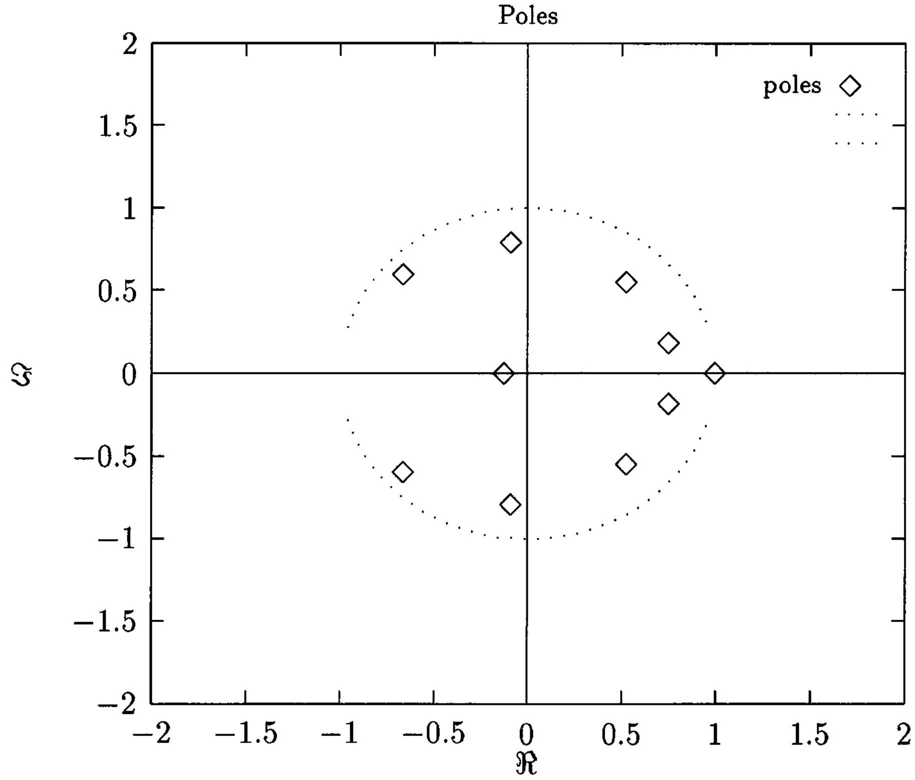

Figure 4.14: Zeros of the autoregressive filter after **restoration** of the pitch contour of the sentence *T16* from fifty percent of the voiced segments. The *autocovariance method* without removing the declination was used both for the reconstruction and determining the autoregressive filter.

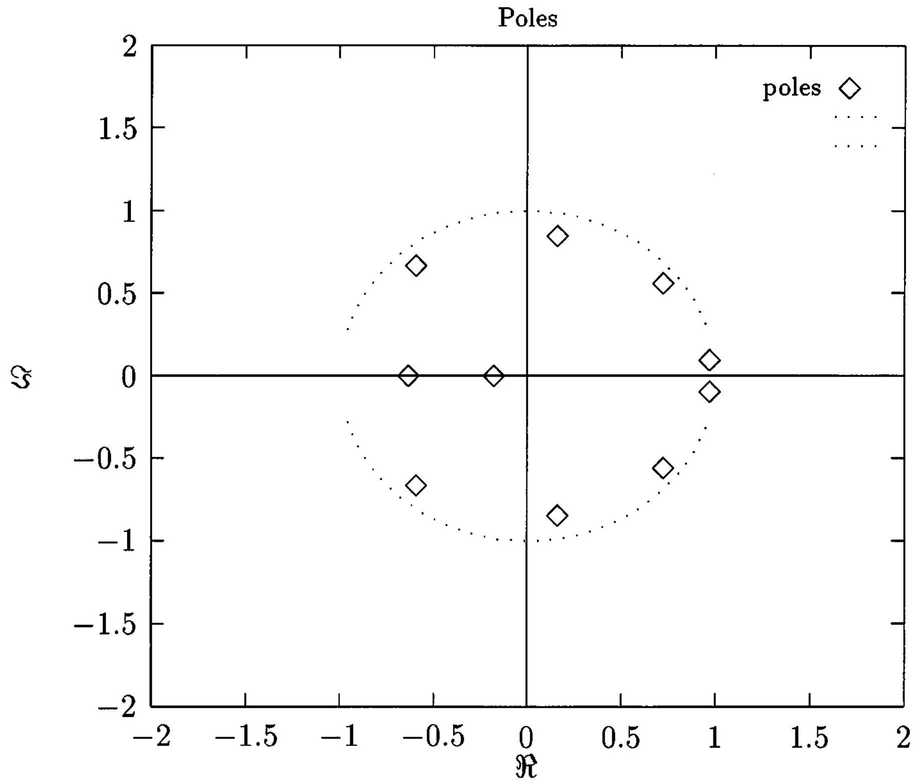

Figure 4.15: Zeros of the autoregressive filter after **restoration** of the pitch contour of the sentence *T16* from fifty percent of the voiced segments. The *autocovariance method* with removal of the the declination was used both for the reconstruction and determining the autoregressive filter.

#### 4.3.2.4 Comparison

In this subsubsection it will be examined under what conditions the restorations are good, and under what conditions they get worse. To compare these, the measures introduced in Section 4.2 will be used. This thus means that *all comparing will be done in the ERB scale.* From now on, the abbreviations from Table 4.3 will be used (in other tables). Unless explicitly stated otherwise, the pitch contours were processed in the Hz-scale. When the pitch contours were processed in the ERB scale, this is denoted by a suffix ERB after the abbreviations as introduced in Table 4.3. In the following paragraphs, subsequently it will be examined if the restoration results depend on

Table 4.3: Abbreviations.

| Method | Abbreviation with previously subtracting the declination | Abbreviation without previously subtracting the declination |
|---|---|---|
| *autocorrelation method* | CorLin | CorNoLin |
| *autocovariance method* | CovLin | CovNoLin |
| *correct autocorrelation method* | CorNoLinCorrect | CorNoLinCorrect |

1. The restoration method, i.e. using the *autocovariance method* or the *autocorrelation method* and whether or not to subtract the declination prior to restoration,

2. Restoration method should be fed in Hz or in ERB's,

3. Using the *correct autocorrelation method* or the *autocorrelation method*,

4. Order of prediction,

5. Number of iterations,

6. The amount of voiced segments that has to be interpolated.

**Influence of the restoration method:** First of all, it is examined whether the particular choice of the restoration method influences the restoration results. For this purpose, Table 4.4 should be examined. Here the quality of the restoration is judged for a restoration of the pitch contours of T16, T53 and T59 as produced by the SHS pitch determination algorithm. Three iterations and a prediction order of ten were used to make the restorations. Fifty percent of the voiced segments had to be interpolated. It can be seen from this table that it hardly makes any difference whether the *autocorrelation method* or the *autocovariance method* is used. The same can be said about subtracting or not subtracting the declination prior to making a restoration. For the pitch contours of For both pitch contours fifty percent of the voiced segments had to be restored. This was done by adjusting the voiced-unvoiced decision threshold until fifty percent of those segments that were originally classified voiced remained. These were then taken to be the known points. For the pitch contour of T59, still one point had to be removed manually, since it was clearly a faulty pitch estimate, before the restorations were made.

**Hz or ERB:** Restoration performed as in the previous paragraph. Now only for the pitch contour of the sentence T16. Then it is examined, by using Table 4.5 whether or not the input of the restoration method should be given with the pitch estimates in Hz (no suffix) or in ERB's (suffix ERB). It can be seen from Table 4.5 that this hardly makes any difference, i.e. the performance does not depend on this choice. This was more or less predictable, since as can been seen from Figure A.8, for pitches under consideration, say 50 to 250 Hz, the transformation from Hz to ERB is almost linear, and the restoration methods do not depend on multiplication by a constant of the data.

**Which autocorrelation method:** Restoration as in the previous paragraph. It can be seen from Table 4.6 that it does not matter whether the - mathematically more correct - *correct autocorrelation method* is used instead of the *autocorrelation method.*

Because it already became clear that the restoration results hardly depend on the actual method that was chosen to make the restoration, and whether the pitches were provided in the Hz or the ERB scale, only one method is chosen to study the effects of the order of prediction and the number of iterations on the quality of the restorations. For this purpose, the restorations were made with the *autocorrelation method* and no declination was subtracted prior to making the restoration. Restoration was done with pitch estimates in Hz. This is expected to be the worst restoration method as expected in the previous subsubsections with respect to the restoration method and because the pitches are provided in Hz, for which one would intuitively expect the restorations to be poorer than for pitch provided in ERB's. Fifty percent of the voiced segments had to be interpolated.

**Dependency on the order of prediction:** Restorations were performed as described above, i.e. the worst-case scenario. From Table 4.7 it follows that the restoration results hardly depend on the order of prediction in a very broad range. The choice to take a prediction order of 10 is thus good.

**Dependency on iterations:** Restorations as for the paragraph above. From Table 4.8 it follows that 1 iteration is definitely not enough to get an optimal restoration, since the restoration results become radically worse. For 3 and 10 iterations there is hardly any difference. It thus follows that it was a good choice to take 3 iterations to make the restorations.

**Effect of how much pitch estimates have to be restored:** In Table 4.9 it is examined for the *autocorrelation method* applied without previously subtracting the declination of the pitch contour, how the restoration results depend on the percentage of the voiced segments that had to be interpolated. It can be seen that, the results hardly depend on the actual percentage of voiced segments that has to be restored. Only if unvoiced segments are used to make the restoration, the restoration results become worse very fast. This indicated that a good algorithm has to be used to divide the pitch contour in known and unknown segments.

A discrepancy may be found between the first and the second column. This is a result of the fact that for the second column, the percentage of voiced segments is determined using the SHS method, while in the third column this is done using the PDT method. It should be remarked that the restoration results may be a little flattered for this pitch contour, since no long consecutive parts with missing voiced segments are present. Even when only 41 percent of the voiced pitch estimates is used to make the restoration, all the unknown voiced pitch estimates are distributed uniformly over the pitch contour.

**Final remark:** All this indicates that all the methods based on an autoregressive model, are stable methods to make restorations of pitch contours, if the following requirements are met:

1. No faulty pitch estimates (in particular pitch estimates for unvoiced segments) are used as known points in the pitch contour that is used for restoration.
2. The number of iterations is 3 (or larger).

Not important for the restoration results are:

Table 4.4: This is a table to show the effect on the restoration errors of the *method* used. Restorations were made for the pitch contours of T16, T53 and T59, determined using the SHS pitch determination algorithm. Fifty percent of the voiced segments had to be restored. The restoration method was the *autocorrelation method* without removing the declination. Three iterations and a predictionorder of ten were used.

| Sentence | Method | $\mathsf{E}$[ERB$^2$] | $\sigma(\mathsf{E})$[ERB$^2$] | $\mathsf{E}_{\mathrm{max}}$[ERB$^2$] |
|---|---|---|---|---|
| T16 | CorNoLin | 0.015 | 0.044 | 0.32 |
| T16 | CorLin | 0.023 | 0.056 | 0.37 |
| T16 | CovNoLin | 0.022 | 0.055 | 0.41 |
| T16 | CovLin | 0.024 | 0.062 | 0.41 |
| T53 | CorNoLin | 0.030 | 0.092 | 0.44 |
| T53 | CorLin | 0.025 | 0.071 | 0.31 |
| T53 | CovNoLin | 0.035 | 0.12 | 0.64 |
| T53 | CovLin | 0.035 | 0.11 | 0.51 |
| T59 | CorNoLin | 0.162 | 0.47 | 2.6 |
| T59 | CorLin | 0.090 | 0.24 | 1.4 |
| T59 | CovNoLin | 0.11 | 0.32 | 1.9 |
| T59 | CovLin | 0.08 | 0.23 | 1.4 |

1. The particular choice for the prediction order.
2. Whether restorations are made using pitches given in the ERB scale or in the Hz scale,
3. Whether the *autocorrelation method*, the *correct autocorrelation method* or *autocovariance method* is used,
4. Whether the declination is modeled separately by subtracting it from the pitch contour prior to making a restoration ot not doing this.

Table 4.5: This is a table to show the effect on the restoration errors of providing the pitch estimates in *Hz or in ERB's.* Restorations were made for the pitch contour of T16 determined using the SHS pitch determination algorithm. Fifty percent of the voiced segments had to be restored. The restoration method was the *autocorrelation method* without removing the declination. Three iterations and a predictionorder of ten were used. The errors should be compared as pairs, as indicated in the table.

| Method | E[ERB$^2$] | $\sigma$(E)[ERB$^2$] | E$_{max}$[ERB$^2$] |
|---|---|---|---|
| CorNoLin | 0.015 | 0.044 | 0.32 |
| CorNoLin ERB | 0.015 | 0.041 | 0.29 |
| CorLin | 0.023 | 0.056 | 0.37 |
| CorLin ERB | 0.024 | 0.065 | 0.44 |
| CovNoLin | 0.022 | 0.055 | 0.41 |
| CovNoLin ERB | 0.021 | 0.050 | 0.40 |
| CovLin | 0.024 | 0.062 | 0.41 |
| CovLin ERB | 0.025 | 0.070 | 0.47 |

Table 4.6: This is a table to determine whether the *correct autocorrelation method* yields significantly better results than the *autocorrelation method.* Restorations were made for the pitch contour of T16 determined using the SHS pitch determination algorithm. Fifty percent of the voiced segments had to be restored. The restoration method was the *autocorrelation method* without removing the declination. Three iterations and a predictionorder of ten were used. The errors should be compared as pairs, as indicated in the table.

| Method | E[ERB$^2$] | $\sigma$(E)[ERB$^2$] | E$_{max}$[ERB$^2$] |
|---|---|---|---|
| CorNoLin | 0.015 | 0.044 | 0.32 |
| CorNoLinCorrect | 0.016 | 0.050 | 0.37 |
| CorLin | 0.023 | 0.056 | 0.37 |
| CorLinCorrect | 0.019 | 0.038 | 0.24 |

Table 4.7: This table is to study restoration errors as function of the *order of prediction.* Restorations were made for the pitch contour of T16 determined using the SHS pitch determination algorithm. Fifty percent of the voiced segments had to be restored. The restoration method was the *autocorrelation method* without removing the declination. Three iterations were used.

| Order of prediction | E[ERB$^2$] | $\sigma$(E[ERB$^2$]) | E$_{\mathrm{max}}$[ERB$^2$] |
|---|---|---|---|
| 1 | 0.021 | 0.080 | 0.58 |
| 5 | 0.019 | 0.066 | 0.50 |
| 10 | 0.015 | 0.044 | 0.32 |
| 50 | 0.014 | 0.041 | 0.32 |
| 100 | 0.013 | 0.036 | 0.26 |

Table 4.8: This table is to study restoration errors as function of the *number of iterations.* Restorations were made for the pitch contour of T16 determined using the SHS pitch determination algorithm. Fifty percent of the voiced segments had to be restored. The restoration method was the *autocorrelation method* without removing the declination. A predictionorder of ten was used.

| Iterations | E[ERB$^2$] | $\sigma$(E)[ERB$^2$] | E$_{\mathrm{max}}$[ERB$^2$] |
|---|---|---|---|
| 1 | 0.21 | 0.24 | 1.1 |
| 3 | 0.015 | 0.044 | 0.32 |
| 10 | 0.015 | 0.044 | 0.32 |

Table 4.9: This is a table to determine if the restoration results depend on the percentage of voiced segments that has to be restored. Restorations were made for the pitch contour of T16 determined using the SHS pitch determination algorithm. The restoration method was the *autocorrelation method* without removing the declination. Three iterations and a predictionorder of ten were used.

| Part of the voiced segments restored [%] as determined by SHS | Part of the voiced segments restored [%] as determined by PDT | E[ERB$^2$] | $\sigma$(E)[ERB$^2$] | $E_{max}$[ERB$^2$] |
|---|---|---|---|---|
| 6 percent unvoiced segments used | 2.6 | 11 | 13 | 29 |
| 7 | 7.9 | 0.044 | 0.086 | 0.30 |
| 19 | 19 | 0.023 | 0.061 | 0.30 |
| 33 | 33 | 0.018 | 0.058 | 0.34 |
| 45 | 44 | 0.014 | 0.040 | 0.27 |
| 59 | 58 | 0.018 | 0.041 | 0.34 |

### 4.3.3 Band-limited model

Restorations using a band-limited model proved not feasible. Although sometimes the results were encouraging, the method was not stable enough to provide a reliable restoration method for pitch contours. Not removing the declination always gives bad results. This is because the known segment is padded with zeros and since the transition is full of high-frequency noise, none of the band-limited methods will work if this transition is not made small by removing the declination. The available pitch contour itself is relatively well band-limited. This can be shown by determining the power density spectrum of the contour using (in this case) a forward covariance spectral estimation technique of order 50, which is in fact the *autocovariance method*, used in this report. An example for the pitch contour of T16, determined using the PDT pitch determination algorithm is given in Figure A.6, Appendix A. Then for each frequency, a graph is made of how much energy is left in the high-pass region from that frequency to the Nyquist frequency. For this, too, an example for the pitch contour of T16, determined using the PDT pitch determination algorithm is given in Figure A.6, Appendix A. Subsequently it can be determined at which frequency 90 or 99 percent of the energy is concentrated in the low-pass region. This is shown in Table 4.10. Notwithstanding this excellent band-limitedness, that in many cases gave satisfactory restorations, it proved not sufficient to guarantee a stable restoration algorithm. This is shown for a simple example for the most stable method. Therefore the pitch contour of T16, determined using the SHS pitch determination algorithm was used, for which fifty percent of the voiced samples had to be restored. Here a bandwidth of 32.5 percent[3] of the fundamental band was taken. A windowed version of the $g_k$ was used, with

Table 4.10: Cut-off frequencies for the pitch contours of T16, T53 and T59 for which 90, respectively 99 percent of the energy is Low-Pass.

| Pitch Contour | Relative Bandwidth [ % ] at which 90 percent of the energy is limited to that passband | Relative Bandwidth [ % ] at which 99 percent of the energy is limited to that passband |
|---|---|---|
| T16 | 3.8 | 17.9 |
| T53 | 11.0 | 32.5 |
| T59 | 6.7 | 25.4 |

[3] the maximum from Table 4.10 to have every pitch contour for at least 99 percent band-limited.

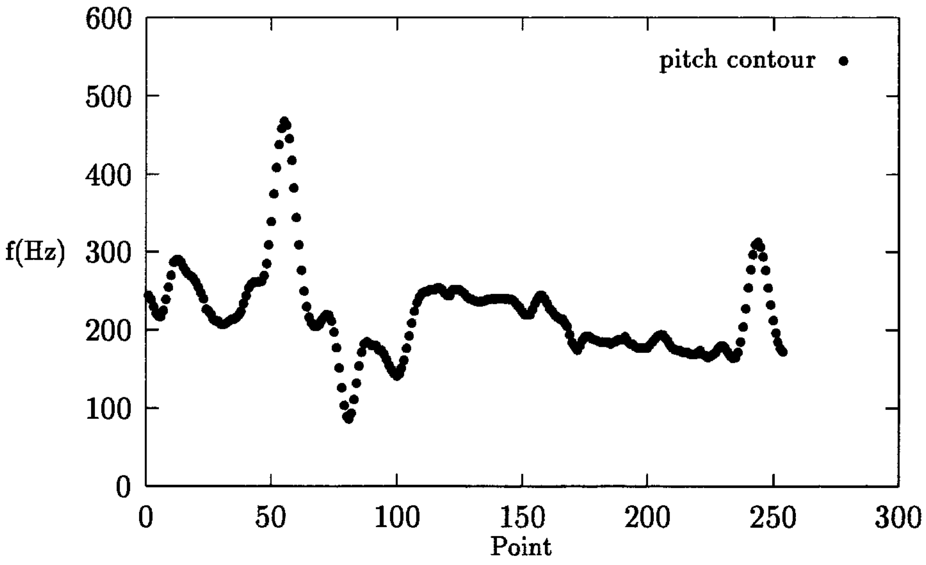

Figure 4.16: ***Restored*** pitch contour of the sentence T16. The pitch contour was determined using the SHS pitch determination algorithm. To make the **restoration** the band-limited method was used, with the bandwidth set to 32.5 percent of the fundamental interval. A windowed version of the $g_k$ was used, with windowing function $W(x) = \mathrm{e}^{-x^2}$, and $\gamma = 10^{-2}$. This means that the windowed $g_k(\gamma)$, were given by $g_k(\gamma) = W(\sqrt{\gamma}k)g_k$. Fifty percent of the voiced samples had to be restored.

windowing function $W(x) = \mathrm{e}^{-x^2}$, and $\gamma = 10^{-2}$. This means that the windowed $g_k(\gamma)$, were given by $g_k(\gamma) = W(\sqrt{\gamma}k)g_k$. This was known to be one of the better methods in [1]. A graph of the restored pitch contour is then given in Figure 4.16. It can be seen that it suffers from the pulse-shaped errors as described in [9] and [1]. Because of these unpredictable errors, this method could not be used for restorations.

All the methods based on a band-limited method suffer from these errors. None of these is therefore fit to make restorations of pitch contours.

### 4.3.4 Choice of the restoration method

It is argued in Subsection 4.3.3 why the restoration methods based on an autoregressive model will not be used to make restorations of pitch contours because they are not reliable enough. This leaves the autoregressive methods as discussed in Subsection 4.3.2 as the only possible choices. Based on the arguments in Subsection 4.3.2 the restorations will be made with the so-called *autocovariance method.* Before the restorations are made, the declination is removed (and added again afterwards). Furthermore, the errors are measured in ERB's as discussed in Section 4.2. The pitch estimates, however are fed to the restoration methods in the Hz-scale and not in the ERB scale. This is a matter of conveniency, and it does not have any measurable (negative) effect on the restoration results, as pointed out in Subsection 4.3.2.

## 4.4 Algorithm

In this section an algorithm to make a division between *known* - that is voiced segments for which a good pitch was determined - and *unknown* - that is unvoiced segments and unreliable voiced segments - is presented. In Subsection 4.4.1 it will be shown that the method of adjusting the threshold of the voiced-unvoiced detection criterion as used in Section 4.3 cannot be used for all pitch contours. In Subsection 4.4.2 then the actual algorithm will be discussed.

### 4.4.1 Limits of using the voiced-unvoiced decision

It will be shown, using the pitch contour of one of the "bad" contours, namely T36, that adjusting the voiced-unvoiced decision cannot be used to make a division in known and unknown points.

In Figure 4.17 the whole pitch contour of T36 as determined by the PDT pitch determination algorithm is shown. This gives a good impression of how the pitch contour for this pitch contour should look like, as judged by an researcher experienced in the field of speech perception research.

In Figure 4.18 the voiced segments of the pitch contour of Figure 4.17 are shown, i.e. the voiced segments of the pitch contour of T36 as determined by the PDT pitch determination algorithm is shown. It should be noted that the quality of the voiced-unvoiced decision is disputable, since this is an example of a so-called creaky voice, that contains lots of noisy sounds even in the segments that should normally be voiced, i.e. the vowels. For example, the drop in Figure 4.17 is judged unvoiced. This is in the middle of the /o/ of *come* (remember that the sentence was *John says he can't come*). This is normally a voiced syllable. In this sentence, however, it is very noise, and therefore *unjustified?* judged unvoiced.

In Figure 4.19 the voiced segments of the pitch contour of T36 as determined by the SHS algorithm is shown. This clearly still contains to much faulty pitch estimates. Furthermore it is interesting to see that in a beginning, there is a place where the pitch drops exactly one octave for a number of segments. This is an example of a so-called octave-failure. Although in some cases this could be an artifact of the pitch determination algorithm, in this pitch contour, the pitch is actually one octave lower in these places. This can be heard of the piece that contains the octave-failure is listened to separately. However, in running speech, it cannot be heard by an inexperienced listener. They hear this part in the "expected" pitch. Therefore, these pitch estimates should be classified unvoiced and restored so that they match the "expected" pitch.

If one tries to remove both octave-failures and other faulty pitch-estimates from the pitch contour by adjusting the voiced-unvoiced decision, one finally ends up with a pitch contour as in Figure 4.20. This still contains lots of pitch estimates that are wrong. This includes some of the octave failures and on the end some other faulty pitch estimates. Even if they would have been removed by this procedure, it is the experience of the author that the good pitch estimates that remain in Figure 4.20 are not sufficient to make any reliable restoration at all.

From this it is clear that some other algorithm has to be developed to divide the pitch contour in known and unknown points. The algorithm that was developed for this purpose is presented in the next subsection.

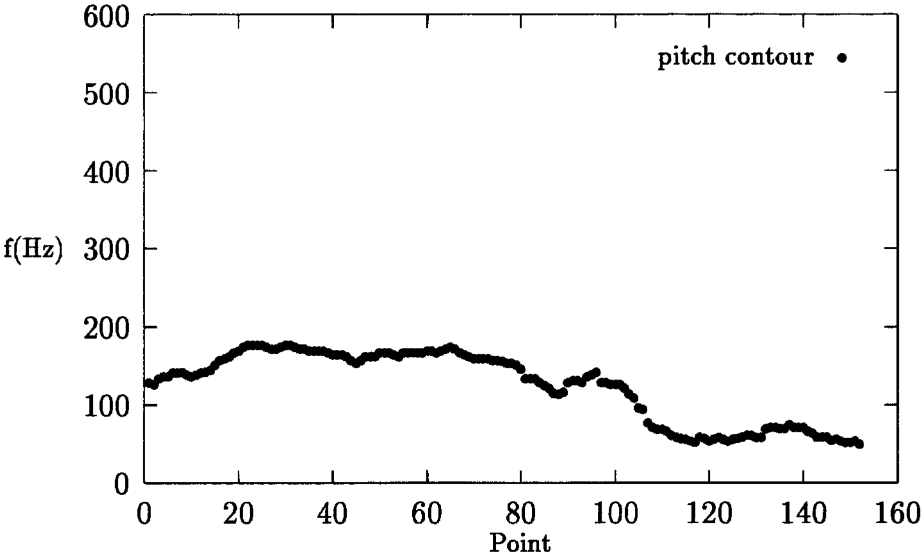

Figure 4.17: Graph of the pitch contour of the sentence *T36* as determined by the *PDT* pitch determination algorithm.

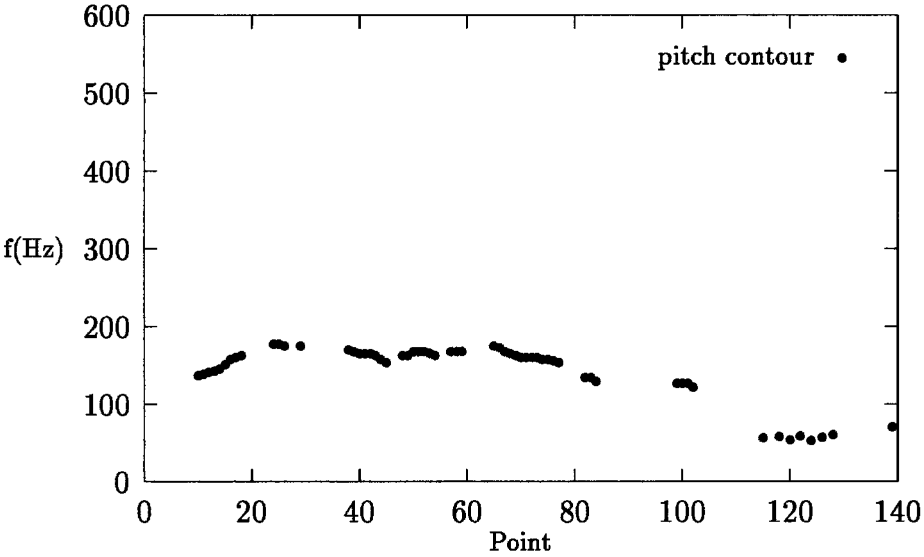

Figure 4.18: Graph of the pitch contour of the sentence *T36* as determined by the *PDT* pitch determination algorithm. Only the voiced segments are shown.

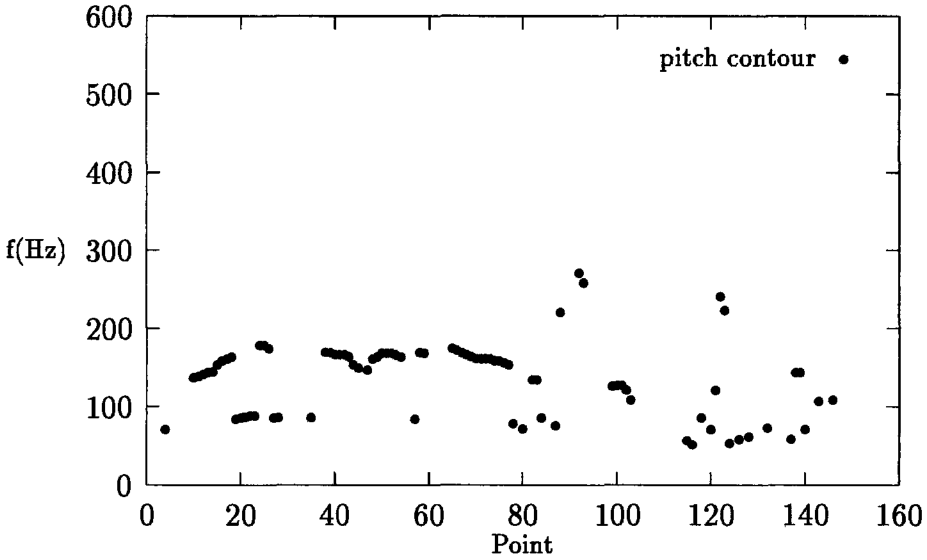

Figure 4.19: Graph of the pitch contour of the sentence *T36* as determined by the *SHS* pitch determination algorithm. Only the voiced segments are shown. Lots of octave-failures can be seen.

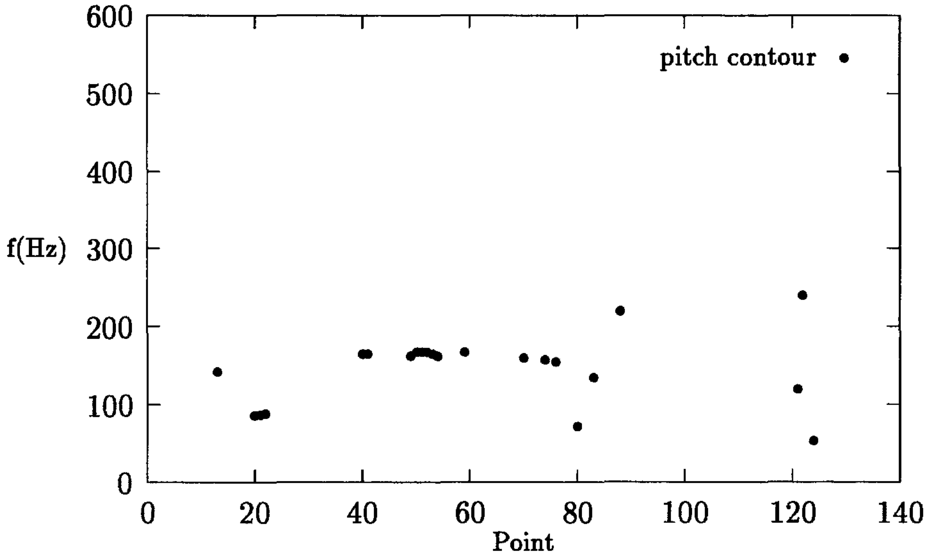

Figure 4.20: Graph of the pitch contour of the sentence *T36* as determined by the *PDT* pitch determination algorithm. Only segments are shown that were judged voiced by the voiced-unvoiced determination algorithm, after significantly raising the threshold value. It can be seen that still lots of faulty pitch estimates remain.

### 4.4.2 The algorithm

Here the algorithm will be presented that divides the segments in known and unknown segments. It is noted here, that several algorithms were tried that did so merely by using characteristics of the pitch estimates for a whole contour. None of these yielded a reliable method that worked for all sentences, although some of them yielded good results for the "good" contours. It was therefore decided to make use of other information, in particular the sampled data.

First of all, it will be outlined, what segments have to be classified as unknown. These are

1. Segments that are unvoiced,
2. Segments with octave-failures, as discussed in Subsection 4.4.1 and can be seen in Figure 4.19,
3. Segments where the speech is so low in volume that environmental noises become audible that might disturb the pitch determination algorithm. An example of this is the fifty hertz noise source in the pitch contour of Figure 4.1.

The algorithm is in fact a modification of the SHS algorithm as described in Subsection 3.1.2. The pitch estimates remain the same, but instead of the voiced-unvoiced decision it outputs a number that, if below 0.05 means that the segment is classified unknown and if above 0.05, it is classified as known. Now, it will be discussed how the above requirements for classifying segments as unknown are met.

**Unvoiced segments:** In Subsection 3.1.2 the SHS pitch determination algorithm is described. From this a pitch estimate is found, call this $\hat{p}$. Now introduce, $H(\hat{p})$, where $H(f)$ is the subharmonic spectrum as described in point 7 of the SHS algorithm. It is expected that, if the segment is voiced, that a great deal of the spectral energy available in the signal, contributes to the virtual pitch, estimated by $\hat{p}$. This means that if one takes

$$\int_{s_0-\Delta s}^{s_0+\Delta s} H(s)\mathrm{d}s,$$

in a neighbourhood $\Delta s$ of $s_0 = \log_2 \hat{p}$, that is expected to encompass much of the peak attributed to the virtual pitch, that this would be not too small compared to

$$\int_0^{\infty} P(s)\mathrm{d}s,$$

with $P(s) = P(f)$, $f = 2^s$. Note that the integrals are taken in a logarithmic scale. In practice $\Delta s$ is taken such that the integral is taken over $\frac{1}{4}$ semitone, which corresponds to $\frac{1}{48}$ of an octave. A segment is now judged voiced if

$$\int\limits_{s_0-\Delta s}^{s_0+\Delta s} H(s)\mathrm{d}s \geq \frac{1}{2}\int\limits_{0}^{\infty} P(s)\mathrm{d}s,$$

If a segment is judged unvoiced, it is classified as an unknown point.

**Octave failures:** For downward octave failures, that is a sudden drop for a few segments of one octave, an interesting phenomenon was observed. It turned out that in octave failures, there was hardly any spectral pitch associated with the virtual pitch that was estimated, i.e. the pitch estimate $\hat{p}$. in other words this means that, that $P(\hat{p}) \ll H(\hat{p})$. For the virtual pitch one octave above the pitch it is then expected that $P(2\hat{p}) \approx H(2\hat{p})$ The following practical implementation was made. If

$$\frac{P(\hat{p})}{H(\hat{p})} < \frac{1}{10}\frac{P(2\hat{p})}{H(2\hat{p})},$$

then the segment was a candidate for an octave failure and classified as unknown too. No such thing could be observed for upward octave failures. It was as yet not impossible to develop a reliable algorithm that could detect upward octave failures, based on the available pitch contour or any additional information. Therefore nothing was implemented to classify upward octave failures as unknown.

**Low volume segments:** For this, first the whole pitch contour was determined. For each segment, $H(\hat{p})$ was determined. Subsequently the average $\bar{H}(\hat{p})$ for the whole pitch contour was determined. If for a segment

$$H(\hat{p}) < \frac{1}{2}\bar{H}(\hat{p}),$$

then the segment was a candidate for a low-volume segment and classified as unknown..

An example of how this performs can be seen in Figure 4.21, where the known points, determined with this algorithm of the pitch contour of T36, determined with the SHS pitch determination algorithm is show. It will be clear from this that still some isolated points are classified known unjustified. To remove this, a simple post-processing algorithm is applied that classifies pitch estimates of segments unknown if:

1. The direct neighbour of that segment has a pitch estimate that is more than one fourth of an octave different in pitch,
2. If in the direct neighbourhood, defined by the two points on either side, there are not at least three adjacent known points,
3. The point was already classified as unknown.

This procedure is iterated until no more points become classified as unknown, i.e. it is stable. This then yields for the pitch contour of T36 the pitch contour as shown in Figure 4.22. It proved to be a fairly reliable and good algorithm. In the authors opinion, the case for the pitch contour of the sentence T36 was one of the most difficult cases. Even for this contour, a fairly satisfactory division is made in known and unknown points. It should be noted that all this (the algorithm as well as the statement that it works rather well) is more based on intuition than on any hard numerical data.

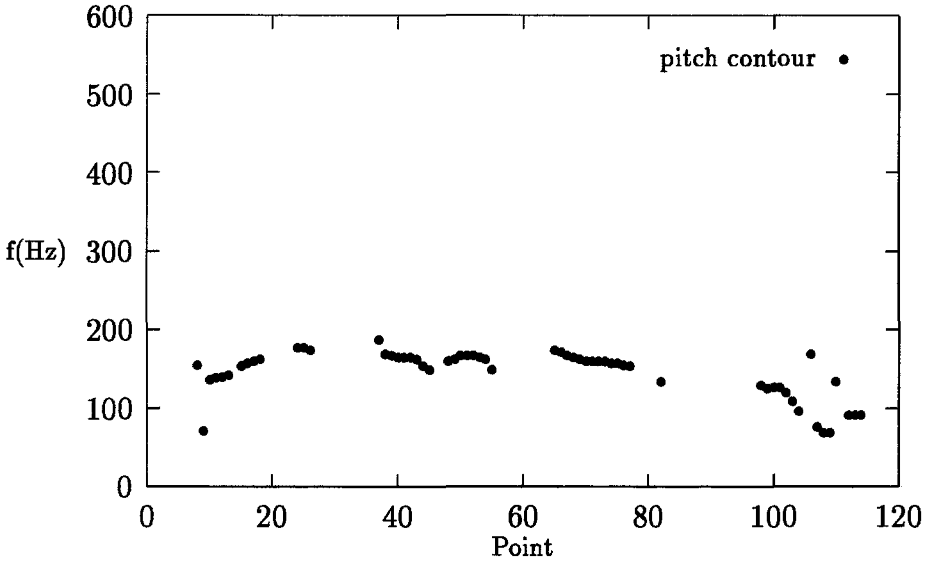

Figure 4.21: Graph of the pitch contour of the sentence *T36* as determined by the *SHS* pitch determination algorithm. Only the known segments are shown, as determined by the algorithm presented in this section, without the postprocessing part.

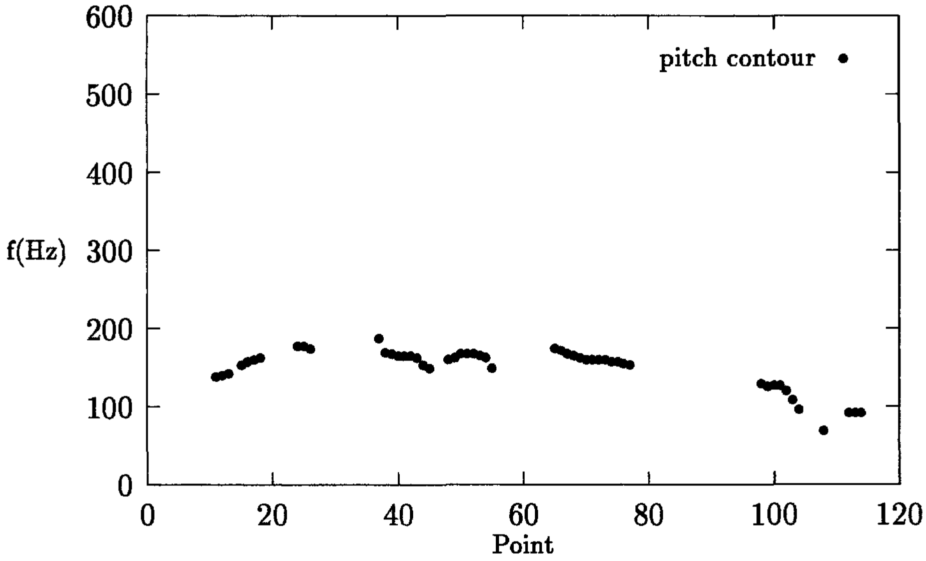

Figure 4.22: Graph of the pitch contour of the sentence *T36* as determined by the *SHS* pitch determination algorithm. Only the known segments are shown, as determined by the algorithm presented in this section, wit the postprocessing part.

## 4.5 Restorations

In this section the results are presented of the restorations of pitch contours. Results are presented for all the pitch contours listed in Table 4.1 and Table 4.2. Since for the "bad" pitch contours it cannot be assumed that the PDT pitch determination algorithm gives good results, the restoration errors, as listed in tablefinal, should not be taken to literally.

Table 4.11: This is the table with the restoration errors for the final restorations that were made for all pitch contours. This is listed for completeness, because for the "bad" contours, these measures are not reliable because the PDT algorithm might not perform well. Therefore it is not a reliable method to compare the restored pitch contours with. Therefore all the restorations are discussed separately in the text.

| Pitch contour | $\mathsf{E}[\mathrm{ERB}^2]$ | $\sigma(\mathsf{E})[\mathrm{ERB}^2]$ | $\mathsf{E}_{\mathrm{max}}[\mathrm{ERB}^2]$ |
|---|---|---|---|
| T16 | 0.037 | 0.067 | 0.27 |
| T53 | 0.060 | 0.096 | 0.29 |
| T59 | 0.12 | 0.24 | 1.3 |
| T6 | 0.10 | 0.31 | 1.1 |
| T14 | 0.8 | 2.2 | 7.7 |
| T19 | 3.1 | 3.5 | 8.0 |
| T36 | 1.0 | 1.0 | 2.8 |

To give a better impression of how the restoration results are, for each sentence the restoration will be compared with the PDT pitch determination by four graphs that have been made for each sentence.

1. First a graph of the pitch contour produced by the PDT pitch determination algorithm is presented, with the voiced segments highlighted. This is taken as the reference.

2. Then a graph is presented that shows the graph of the PDT pitch determination algorithm. Superimposed on this are the pitch estimates as determined by the SHS algorithm that were judged to be known by the algorithm presented in Section 4.4. From this an impression can be formed of how well this algorithm worked for the particular pitch contour.

3. Next a graph is presented of the reconstructed pitch contour, for which the pitch estimates of the known segments were determined using the SHS pitch determination algorithm. The pitch estimates for the unknown segments were estimated from the known pitch estimates, using the *autocovariance method.* The declination had been removed before the restoration method was applied. Along with this are plotted the pitch estimates for the voiced segments of the pitch contour as determined by the PDT pitch determination algorithm. These are the points where a difference in pitch could be heard if they are restored incorrectly. The quality of the restoration has to be judged mainly from this graph.

4. Finally a graph is presented where the pitch estimates for the voiced segments from the original pitch contour as determined by the SHS pitch determination algorithm are shown along with the restored pitch contour. This gives an impression of how "good" or "bad" the pitch contour for the particular sentence is.

For each sentence, these four graphs are then shortly discussed.

First the restorations for the three sentences that produce "good" pitch contours (Table 4.1) will be discussed, followed by the discussion of the sentences that produce "bad" pitch contours (Table 4.2). Restorations were made by

1. Using the SHS pitch determination to make the pitch contour,

2. Using the algorithm from Section 4.4 to divide this pitch contour in known and unknown points,

3. Removing the declination,

4. Making a restoration using the *autocovariance method.*

5. Adding the declination.

**Restored pitch contour of T16:** Figures 4.23, 4.24, **4.25** and 4.26. Not much has to be said about the restoration of this contour. The only main difference between the restored contour and the contour as produced by the PDT pitch determination algorithm is the V-shaped feature, present in the restored and the original SHS pitch contour (Figure 4.25). This is not present in the PDT contour (Figure 4.23. The pitches in the V-shape were determined by the SHS algorithm, and judged known by the algorithm that divides it into known and unknown points. As can be seen from Figure 4.25 it was judged voiced by the original SHS pitch determination algorithm. From Figure 4.23 it follows that it was judged unvoiced by the PDT algorithm. Manual inspection of the wave-form indicates that this V-shape is voiced. It should be investigated if this difference has any perceptual relevance. Without further experiments it cannot be said whether the PDT pitch contour or the restored SHS contour is better.

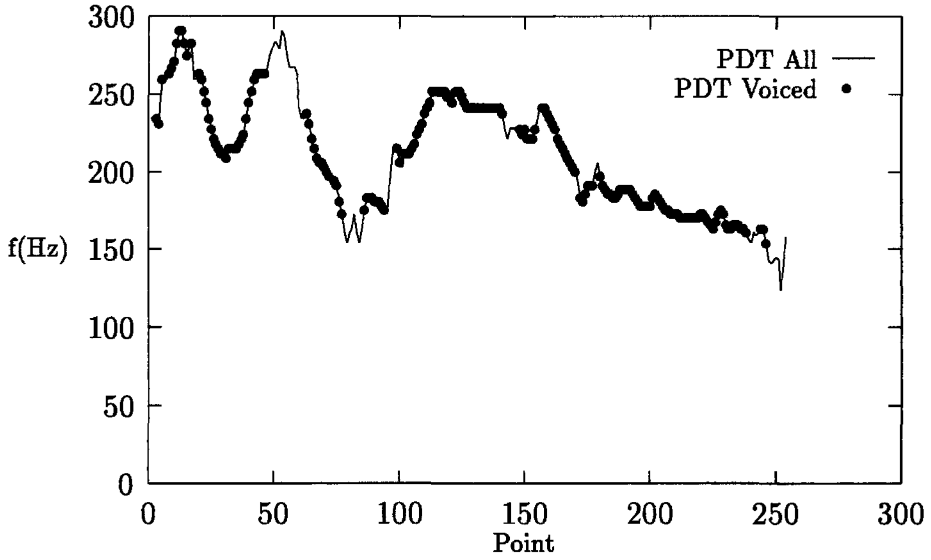

Figure 4.23: Graph of the pitch contour of the sentence *T16*. The line is from the PDT pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

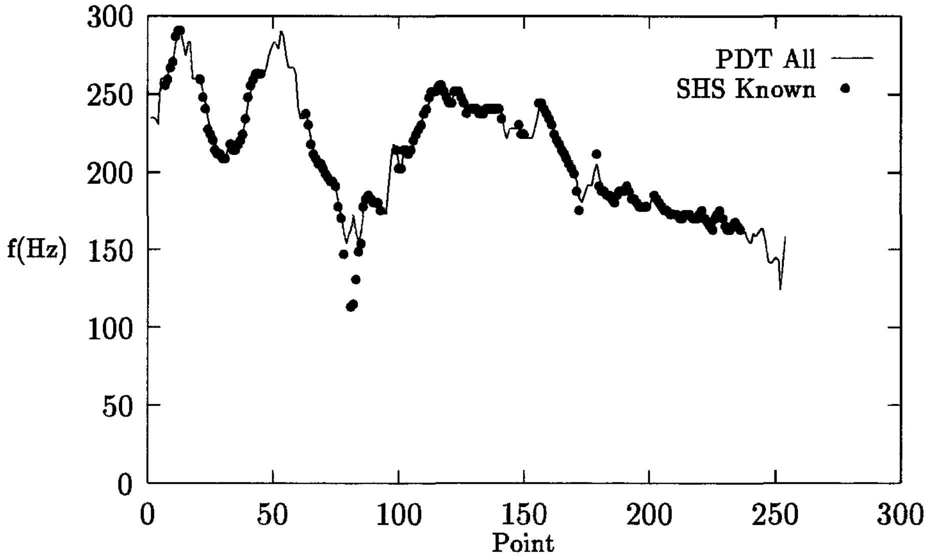

Figure 4.24: Graph of the pitch contour of the sentence *T16*. The line is from the PDT pitch determination algorithm. The points are from the known segments of the SHS pitch determination algorithm.

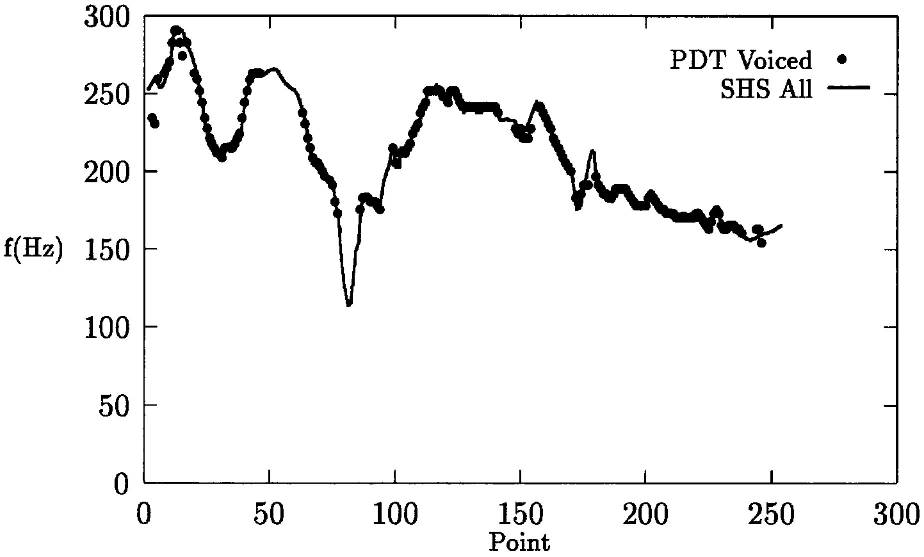

Figure 4.25: Graph of the pitch contour of the sentence *T16*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

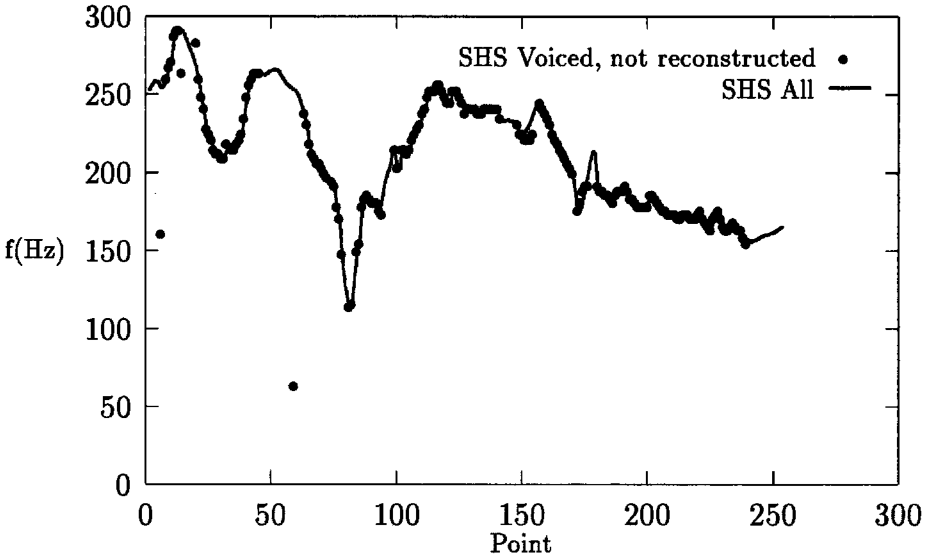

Figure 4.26: Graph of the pitch contour of the sentence *T16.* The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the pitch contour from SHS, that was not yet reconstructed.

**Restored pitch contour of T53:** Figures 4.27, 4.28, **4.29** and 4.30. It can be seen from Figure 4.29 that this is a good reconstruction. Interesting to see is that the "pitch-estimates" for the unvoiced segments on the end of the PDT contour in Figure 4.27 show a sudden upward movement. Although perceptually not relevant, since these are unvoiced segments, it is nice to see that the restoration algorithm gives the result that one would expect. From Figure 4.30 it can be seen that the original SHS pitch contour contained some strange pitch estimates that, moreover were judged voiced, on the end of the pitch contour. These should have been classified unvoiced, since it contains the /f/ and /t/ from *heeft*. The algorithm that divides the pitch estimates in known and unknown ones had no trouble classifying these pitch estimates as unknown.

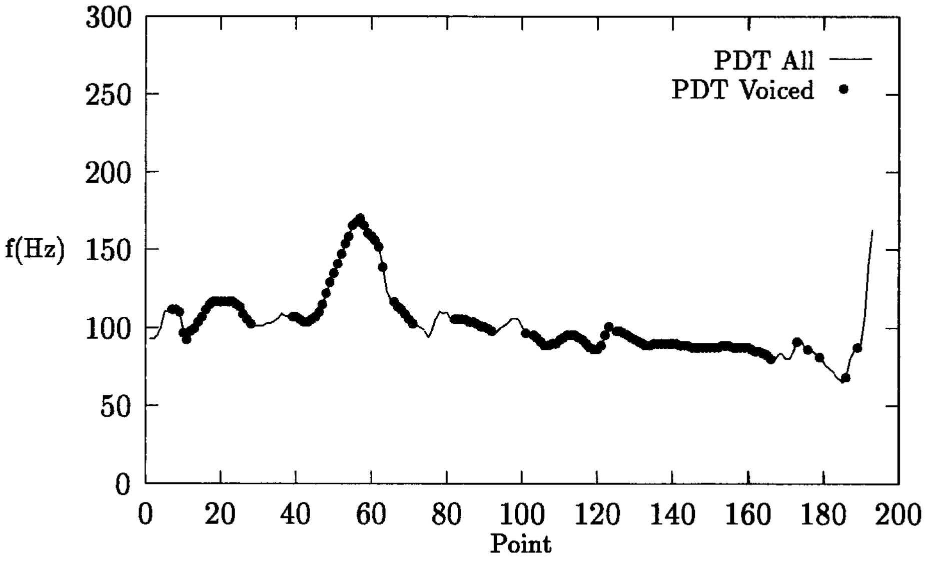

Figure 4.27: Graph of the pitch contour of the sentence *T53*. The line is from the PDT pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

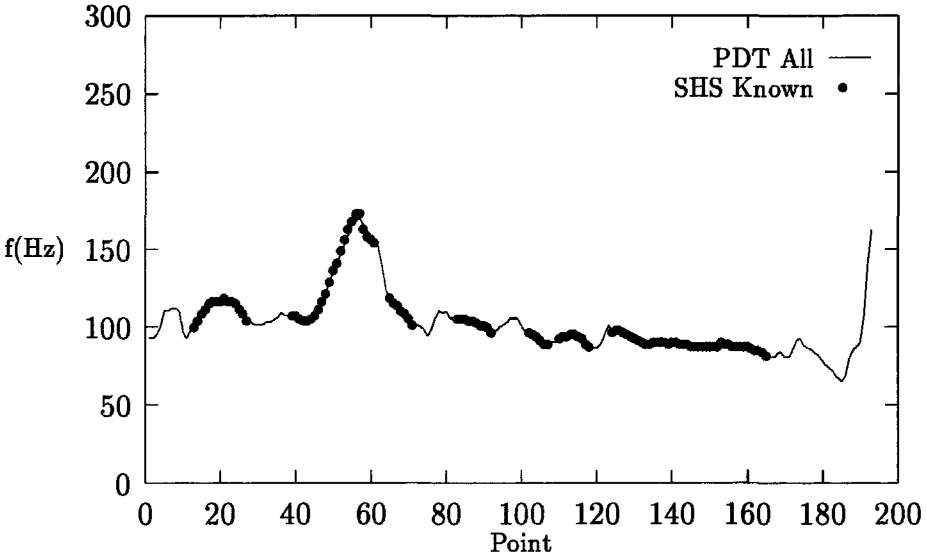

Figure 4.28: Graph of the pitch contour of the sentence *T53*. The line is from the PDT pitch determination algorithm. The points are from the known segments of the SHS pitch determination algorithm.

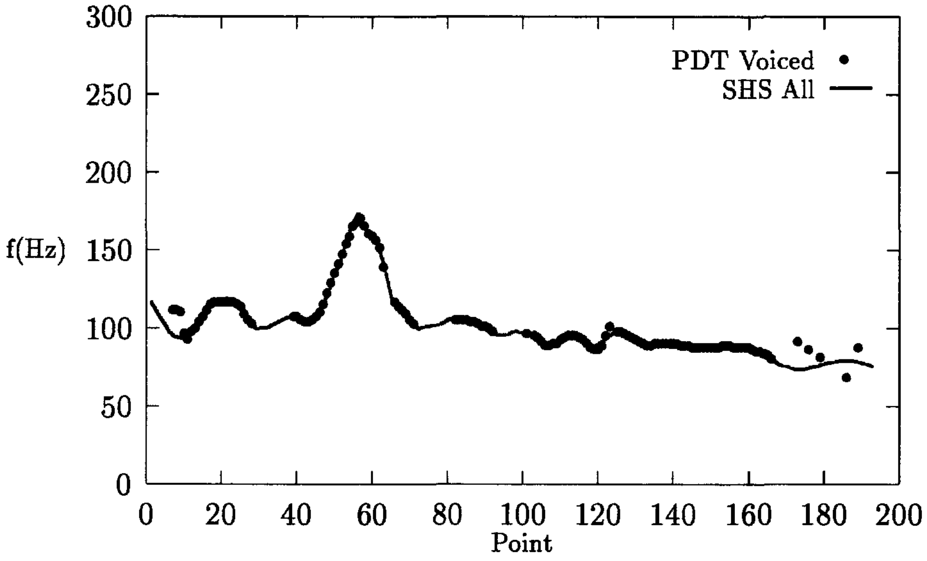

Figure 4.29: Graph of the pitch contour of the sentence *T53*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

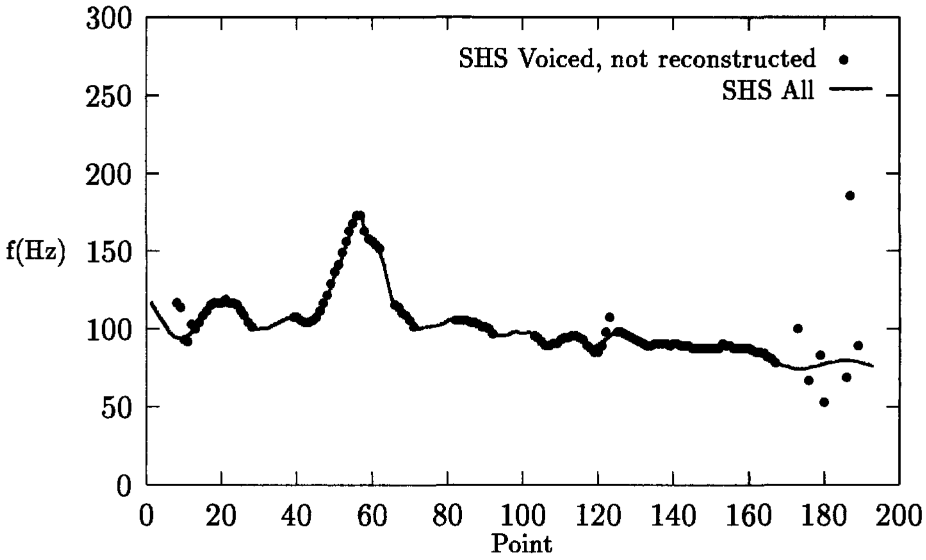

Figure 4.30: Graph of the pitch contour of the sentence *T53*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the pitch contour from SHS, that was not yet reconstructed.

**Restored pitch contour of T59:** Figures 4.31, 4.32, **4.33** and 4.34. Although this pitch contour was difficult for the algorithm that divides the pitch contour in known and unknown points, as can be seen from Figure 4.34, the restoration results are excellent, when compared to the pitch contour produced by the PDT pitch determination algorithm. This can be seen from Figure 4.33.

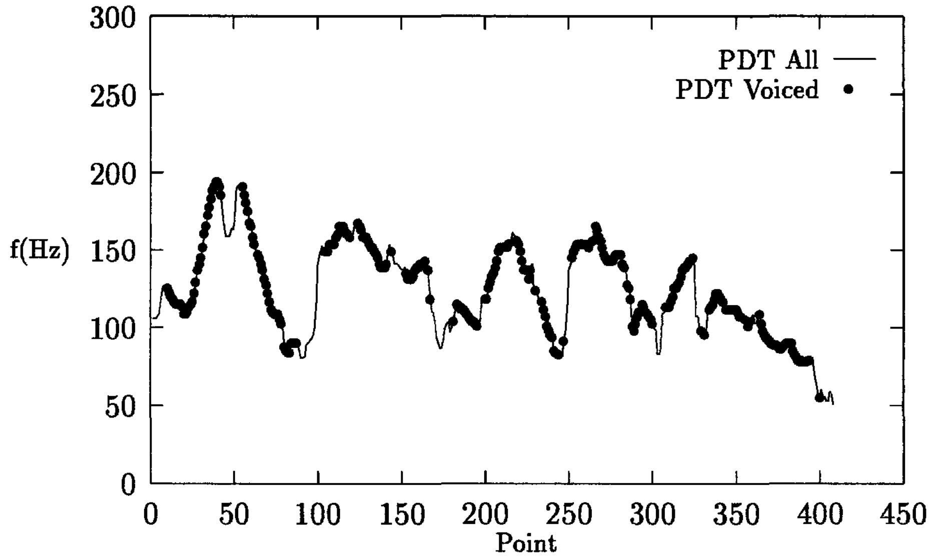

Figure 4.31: Graph of the pitch contour of the sentence *T59*. The line is from the PDT pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

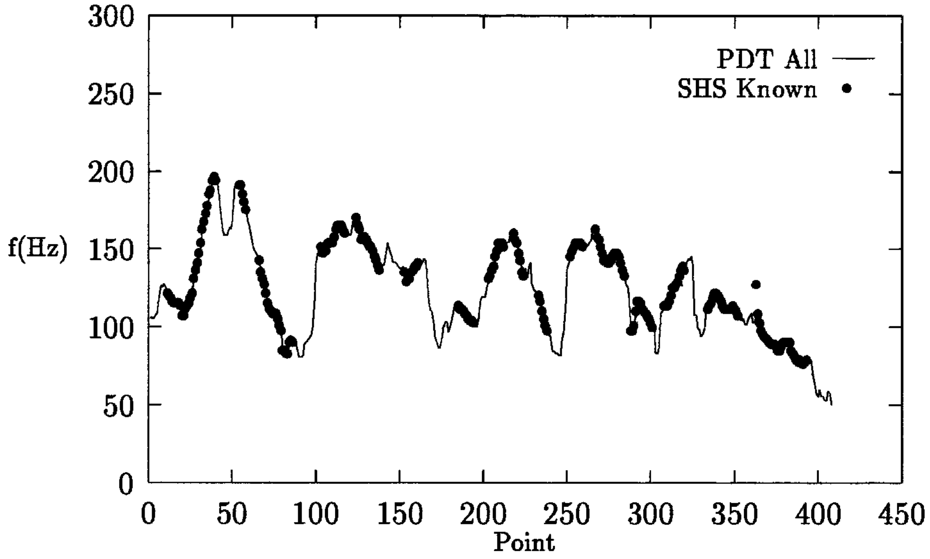

Figure 4.32: Graph of the pitch contour of the sentence *T59*. The line is from the PDT pitch determination algorithm. The points are from the known segments of the SHS pitch determination algorithm.

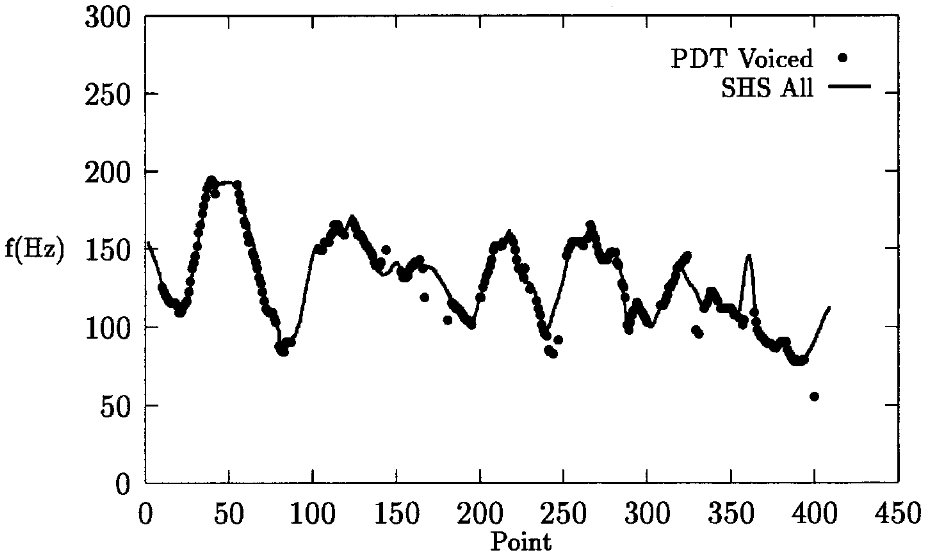

Figure 4.33: Graph of the pitch contour of the sentence *T59*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

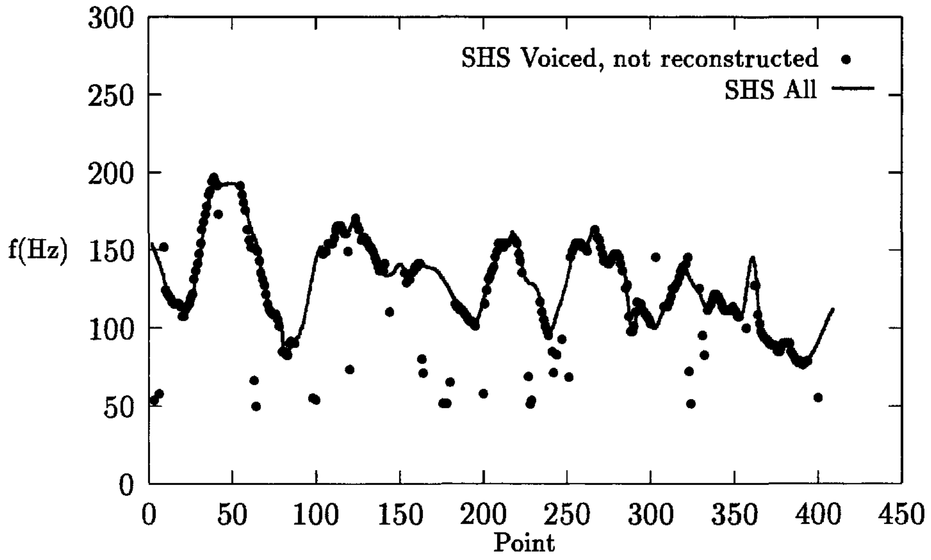

Figure 4.34: Graph of the pitch contour of the sentence *T59*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the pitch contour from SHS, that was not yet reconstructed.

**Restored pitch contour of T6:** Figures 4.35, 4.36, **4.37** and 4.38. This is the first "bad" contour that has to be restored. As can be seen from Figure 4.37, there is only one point in the restored pitch contour that is clearly different from the pitch contour determined by PDT. First of all, this should have been classified unvoiced by PDT because it is in the /t/ of *werd*. Secondly, it is the pitch of a noise-source rather than the pitch of the speech. Therefore, for this pitch contour the restored pitch contour is slightly better than the pitch contour as produced by PDT. Whether or not this can be heard will have to be investigated by doing a perceptual experiment.

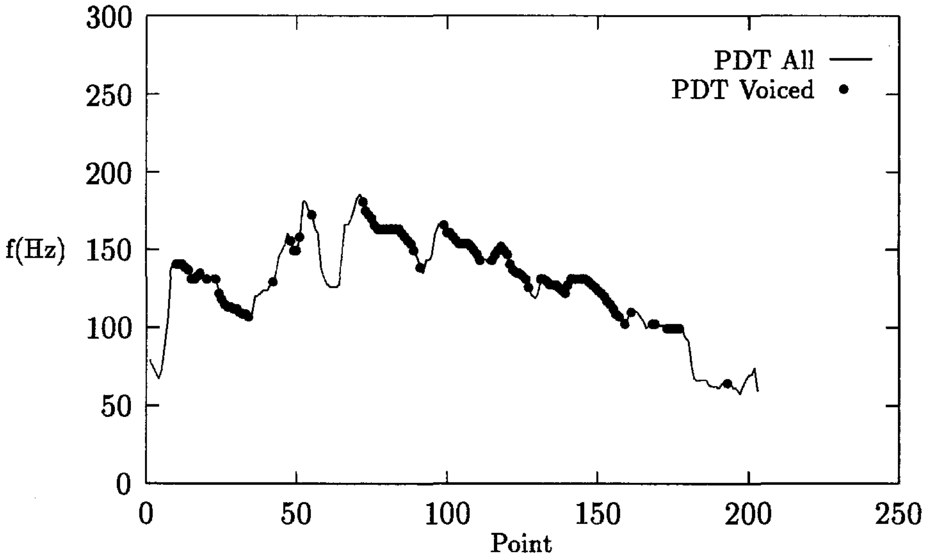

Figure 4.35: Graph of the pitch contour of the sentence *T6*. The line is from the PDT pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

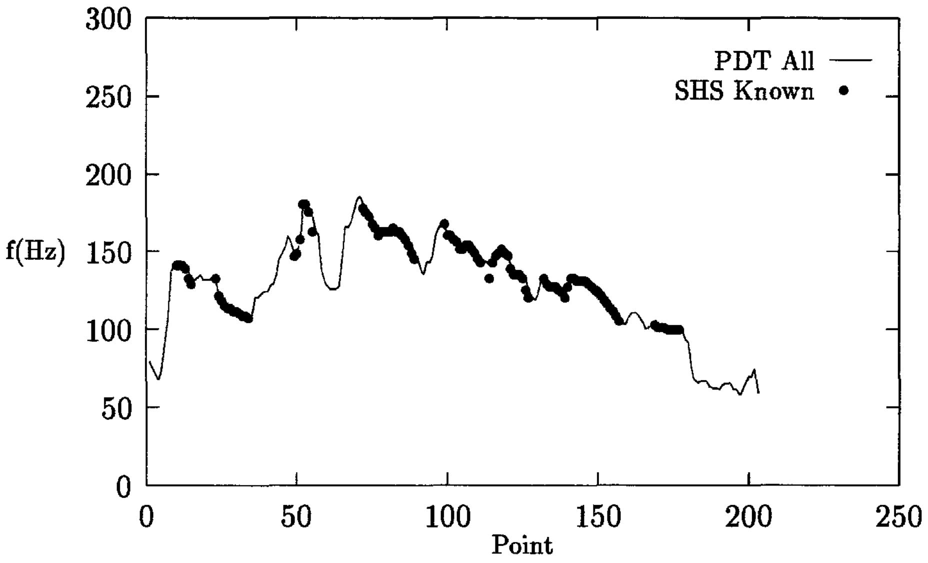

Figure 4.36: Graph of the pitch contour of the sentence *T6*. The line is from the PDT pitch determination algorithm. The points are from the known segments of the SHS pitch determination algorithm.

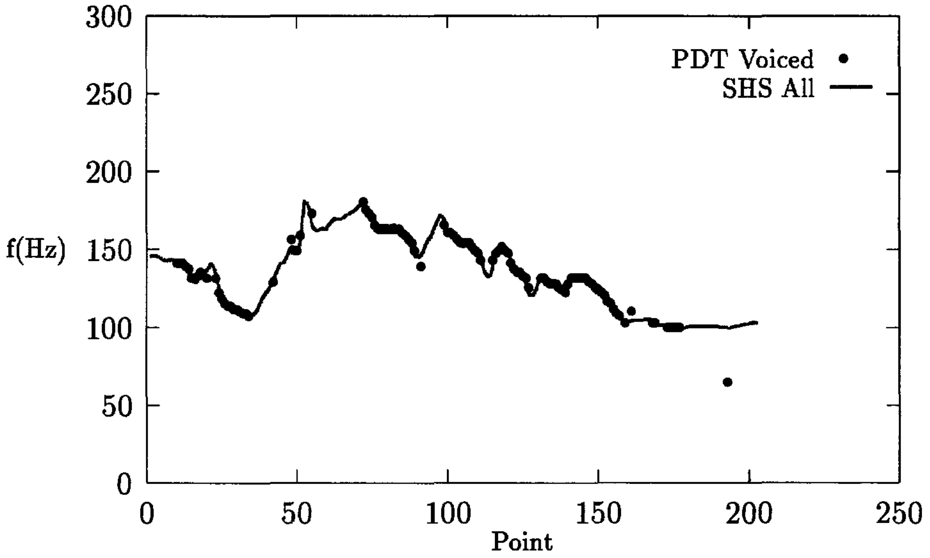

Figure 4.37: Graph of the pitch contour of the sentence *T6*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

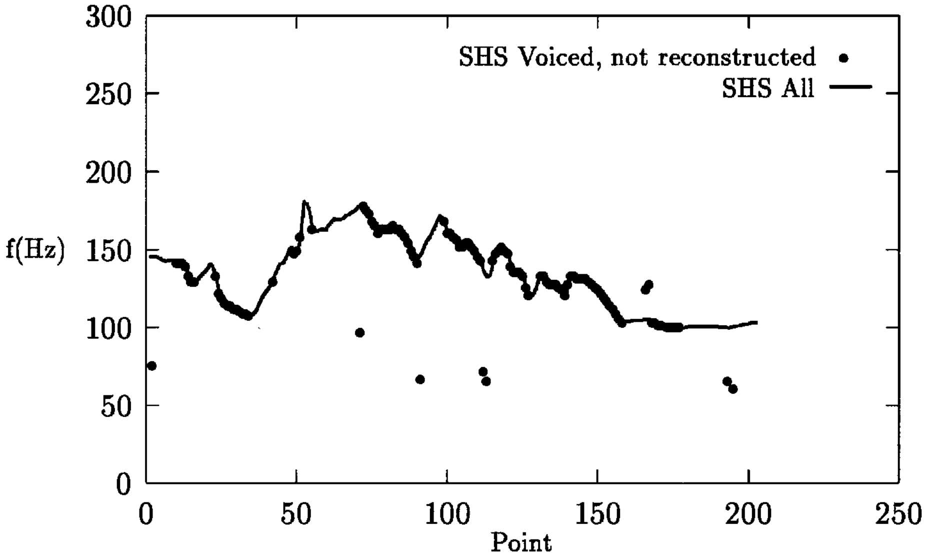

Figure 4.38: Graph of the pitch contour of the sentence *T6*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the pitch contour from SHS, that was not yet reconstructed.

**Restored pitch contour of T14:** Figures 4.39, 4.40, 4.41 and 4.42. From Figure 4.39 it can be seen that there is around point 60 one faulty voiced pitch estimate in the pitch contour from PDT, which was due to a low-frequency noise-source. This is not present in the restored pitch contour, as can be seen from Figure 4.41. However, on the end, the restored pitch contour shows a sudden upward movement which is, according to the author, due to an upward octave failure that was not removed by the algorithm that divides the pitch contour into known and unknown pitch estimates, since no detection algorithm could be implemented to detect upward octave failures. This is probably a very serious shortcoming of the present detection algorithm. It is likely to produce audible errors, so here the restored pitch contour is inferior to that produced by the PDT pitch determination algorithm due to the shortcomings of the algorithm that has to detect the faulty pitch estimates.

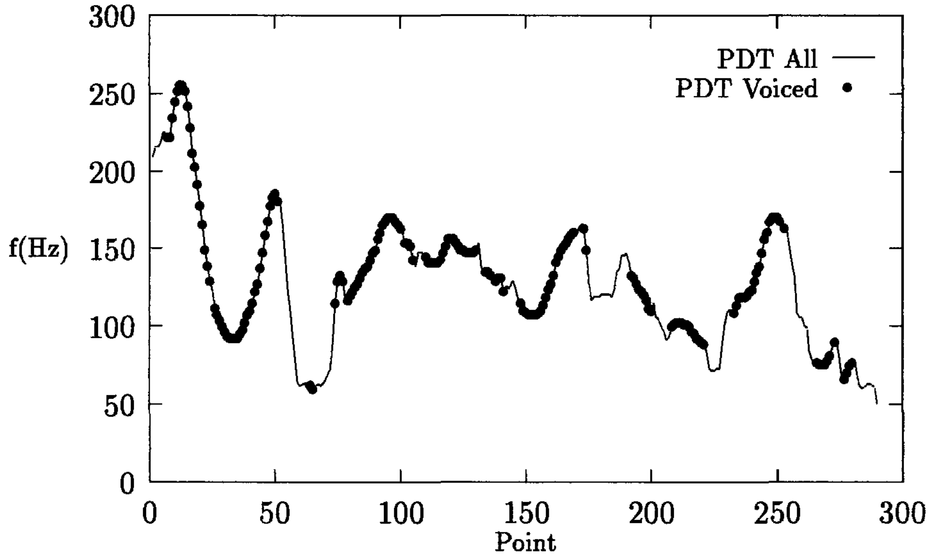

Figure 4.39: Graph of the pitch contour of the sentence *T14*. The line is from the PDT pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

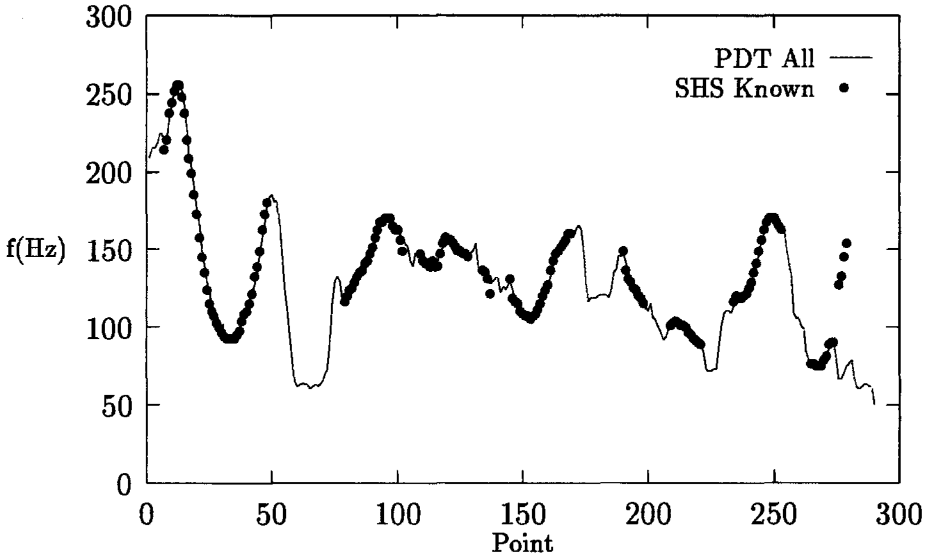

Figure 4.40: Graph of the pitch contour of the sentence *T14*. The line is from the PDT pitch determination algorithm. The points are from the known segments of the SHS pitch determination algorithm.

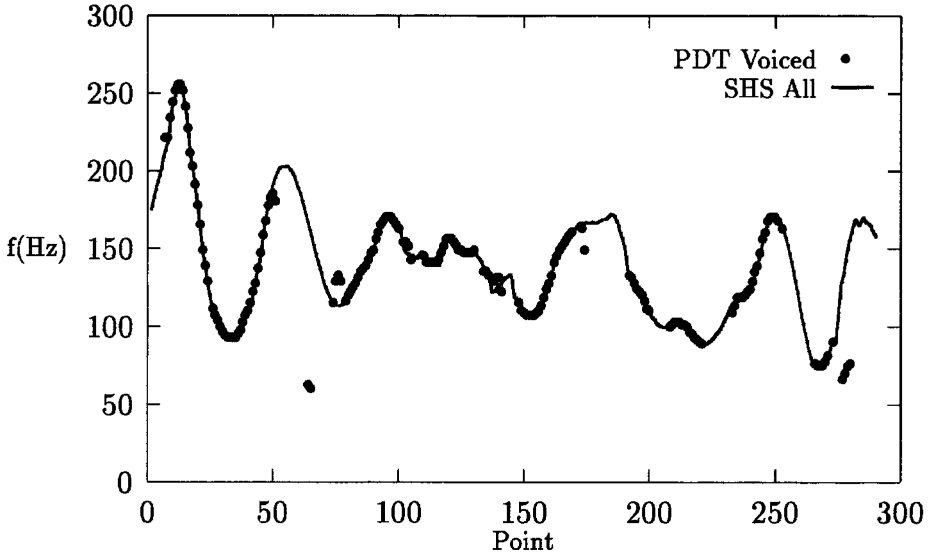

Figure 4.41: Graph of the pitch contour of the sentence *T14*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

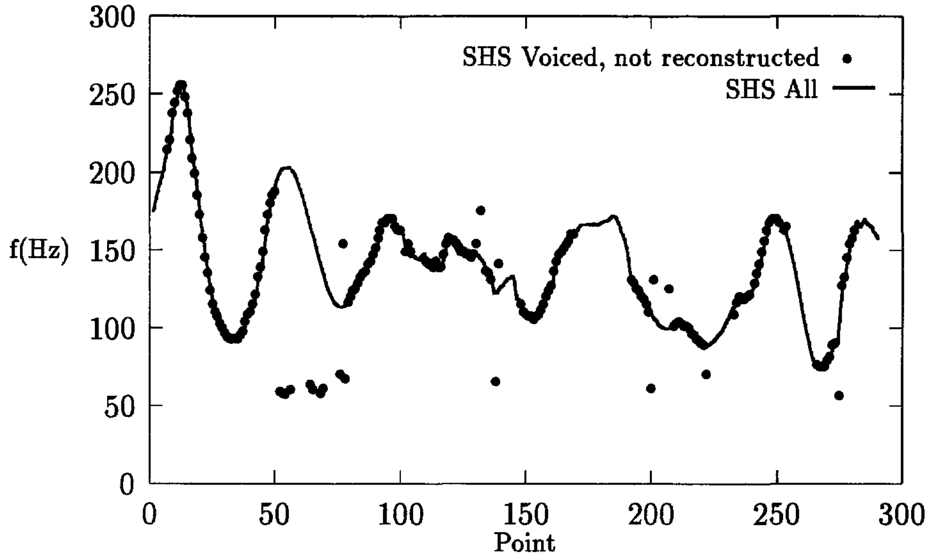

Figure 4.42: Graph of the pitch contour of the sentence *T14*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the pitch contour from SHS, that was not yet reconstructed.

**Restored pitch contour of T19:** Figures 4.43, 4.44, **4.45** and 4.46. Here there is only a difference on the end of the pitch contour, as follows from Figure 4.45. The PDT pitch determination algorithm here follows an octave failure. This is wrong. The algorithm managed to classify these octave failures correctly as unknown. However, the restoration that was made from this goes up as can be seen from Figure 4.45. This is likely to yield a resynthesization that has a pitch that will be perceived to high at the end of the sentence. A simple form of post-processing was tried, that would compare the two graphs of Figure 4.46 and said: " well as can be seen, some of the pitch estimates that were classified unknown are pretty close to the restored pitch contour, why not classify them as known and make a new restoration." Although this will obviously work for this contour, the threshold for reclassifying points as known, will have to be so high that it makes things worse for most of the contours, because it then classifies wrong pitch estimates as known pitch estimates again. This therefore proved not possible. A perceptual experiment whether the PDT pitch contour or the restored pitch contour is perceived as being a better representation of the original.

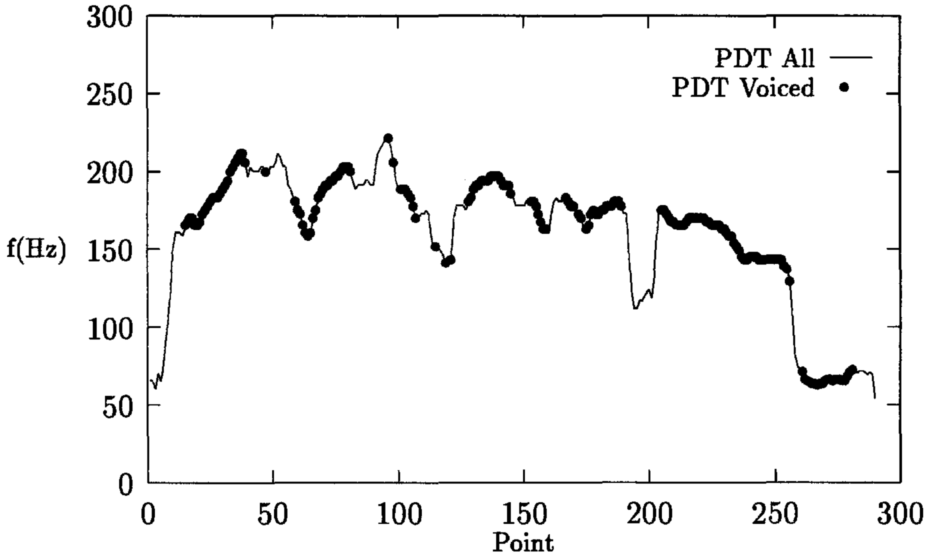

Figure 4.43: Graph of the pitch contour of the sentence *T19*. The line is from the PDT pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

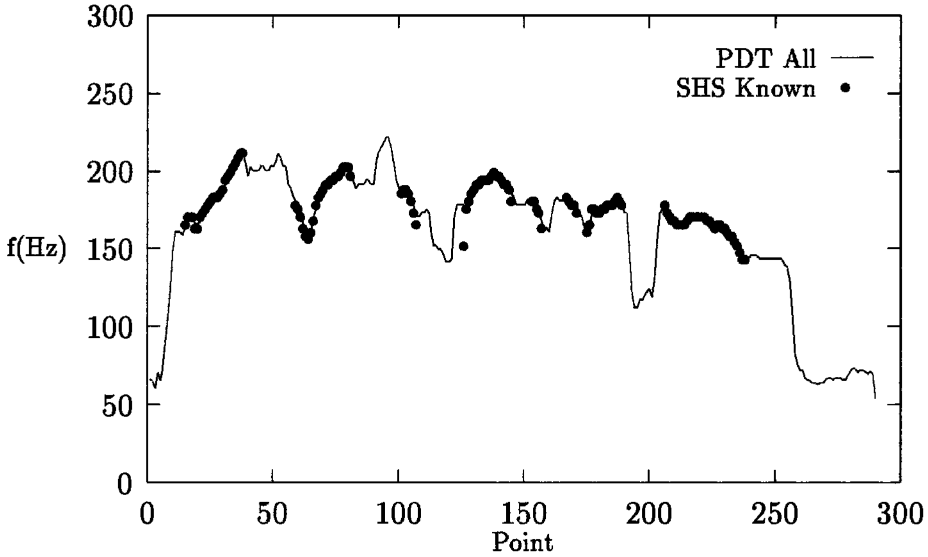

Figure 4.44: Graph of the pitch contour of the sentence *T19*. The line is from the PDT pitch determination algorithm. The points are from the known segments of the SHS pitch determination algorithm.

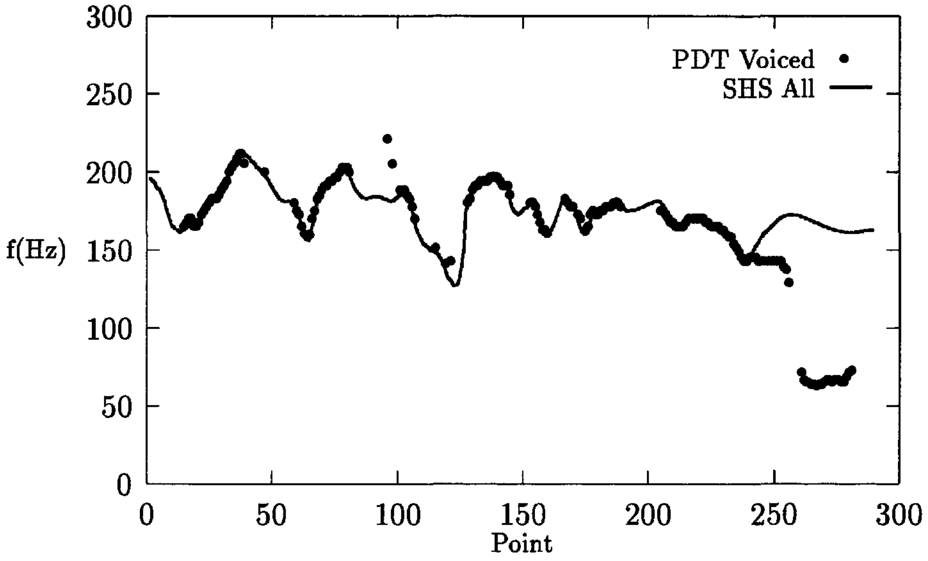

Figure 4.45: Graph of the pitch contour of the sentence *T19*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

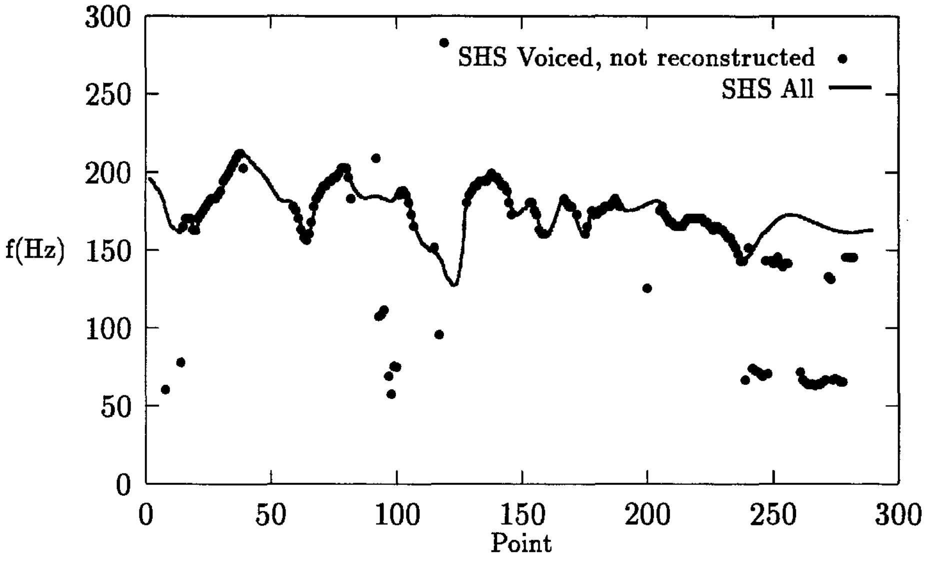

Figure 4.46: Graph of the pitch contour of the sentence *T19*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the pitch contour from SHS, that was not yet reconstructed.

**Restored pitch contour of T36:** Figures 4.47, 4.48, **4.49** and 4.50. Although the method that has to distinguish between reliable pitch estimates and unreliable pitch estimates does a pretty good job, as can be seen from Figure 4.50. The restoration is too high on the end of the pitch contour, as can be seen from Figure 4.49. The PDT pitch algorithm is known to yield a rather good pitch contour for this sentence, so here it is clear that the restoration method fails.

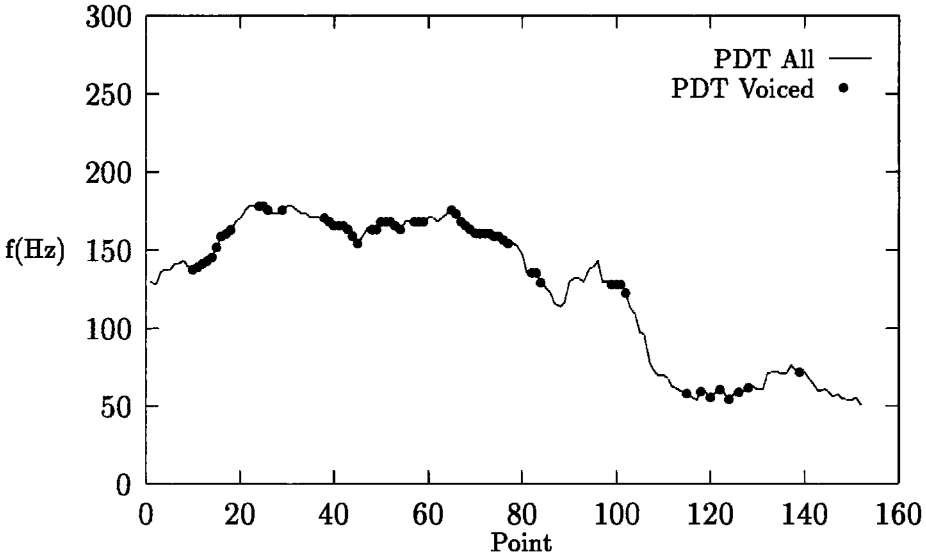

Figure 4.47: Graph of the pitch contour of the sentence *T36*. The line is from the PDT pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

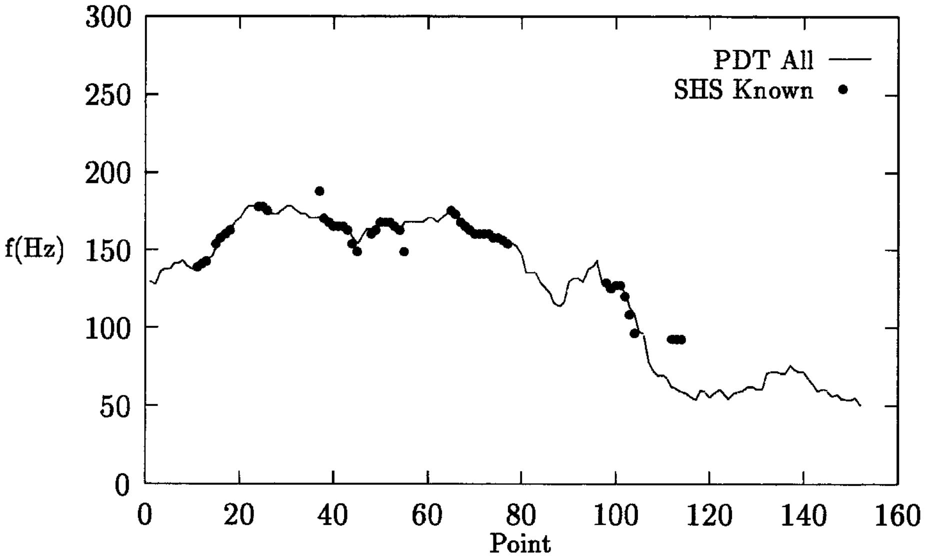

Figure 4.48: Graph of the pitch contour of the sentence *T36*. The line is from the PDT pitch determination algorithm. The points are from the known segments of the SHS pitch determination algorithm.

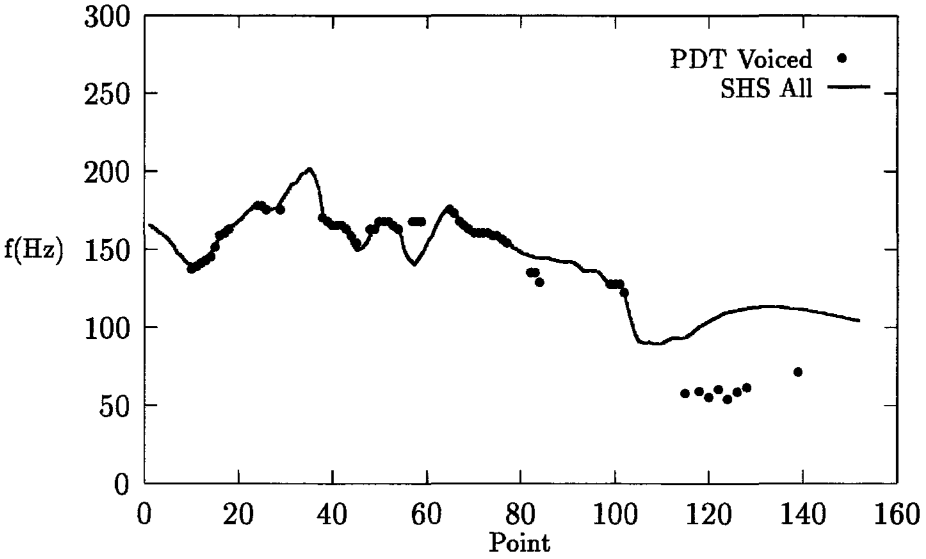

Figure 4.49: Graph of the pitch contour of the sentence *T36*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the PDT pitch determination algorithm.

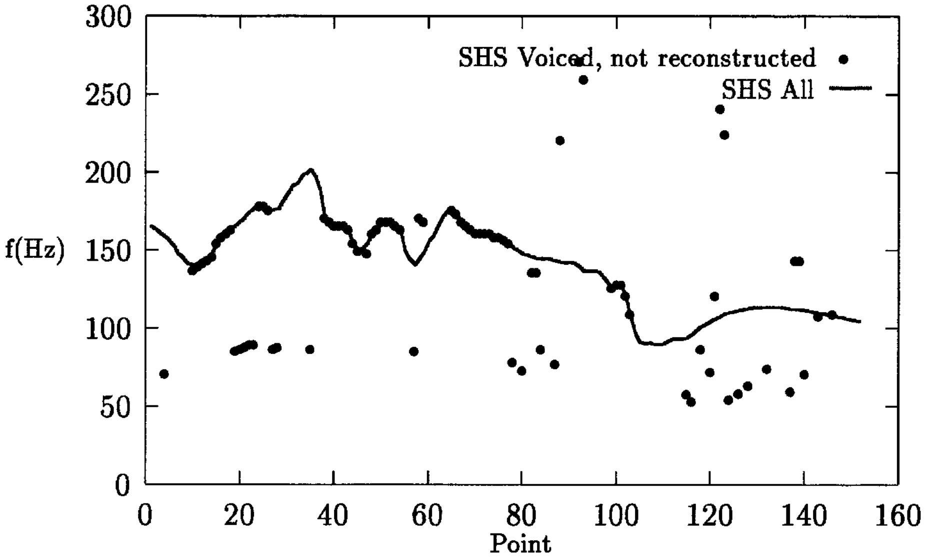

Figure 4.50: Graph of the pitch contour of the sentence *T36*. The line is from the **restored** contour from the SHS pitch determination algorithm. The points are from the voiced segments of the pitch contour from SHS, that was not yet reconstructed.

# Chapter 5

# Conclusions

In this report the restoration of pitch contours produced by the SHS algorithm is discussed. To be able to make a restoration, an algorithm has been developed to decide what pitch estimates have to be restored in the pitch contour and what pitch estimates are already good as delivered by the SHS pitch estimation algorithm.

This algorithm that divides the pitch contour in known and unknown points works fairly well. It fails only in one case (pitch contour of T14), due to upward octave failures, for which no discrimination method was implemented in the algorithm. It could therefore not be expected that the algorithm would recognize these. The algorithm succeeds in classifying as unknown the unvoiced segments, downward octave failures and low-volume segments. Although it was tried to make an algorithm to detect upward octave failures, nothing reliable could be developed. If the existing algorithm for classifying faulty pitch estimates as unknown could be extended with an algorithm to detect upward octave failures, the performance of the algorithm would increase.

The restoration method based on a bandlimited model proved not to be stable enough to make restoration of pitch contours. This is due to the numerical instability and out-of-band components.

The restoration method based on an autoregressive model performs well for "good" pitch contours. For "bad" contours the performance is less good. To measure the quality of the restored "bad" contours, a perceptual experiment has to performed, where sentences re-synthesized with the restored pitch contours are compared to the pitch contours produced by the PDT pitch determination algorithm and the original sentences.

It turned out that restoration did not depend on whether the pitch estimates were fed to the restoration method in Hz or in the psycho-acoustically more correct ERB scale.

This is probably due to the fact that for the frequencies of interest (0-300 Hz), the transformation from one scale to the other is almost linear, and that the restoration methods do not depend on any linear scaling that is applied before the restoration takes place.

# Appendix A

# Graphs

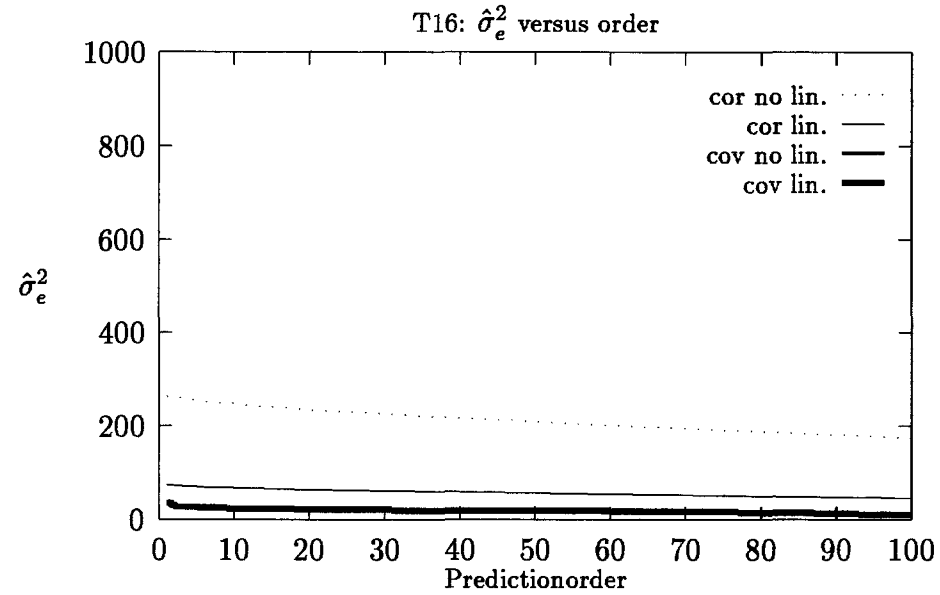

Figure A.1: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the order of prediction for the case, where the autoregressive parameters are determined using (cor) the *autocorrelation method*, and (cov) the *autocovariance method*. This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence *T16*, using the *PDT* pitch determination method. The lines (lin.) and (no lin.) for (cov) almost coincide and can therefore not be distinguished.

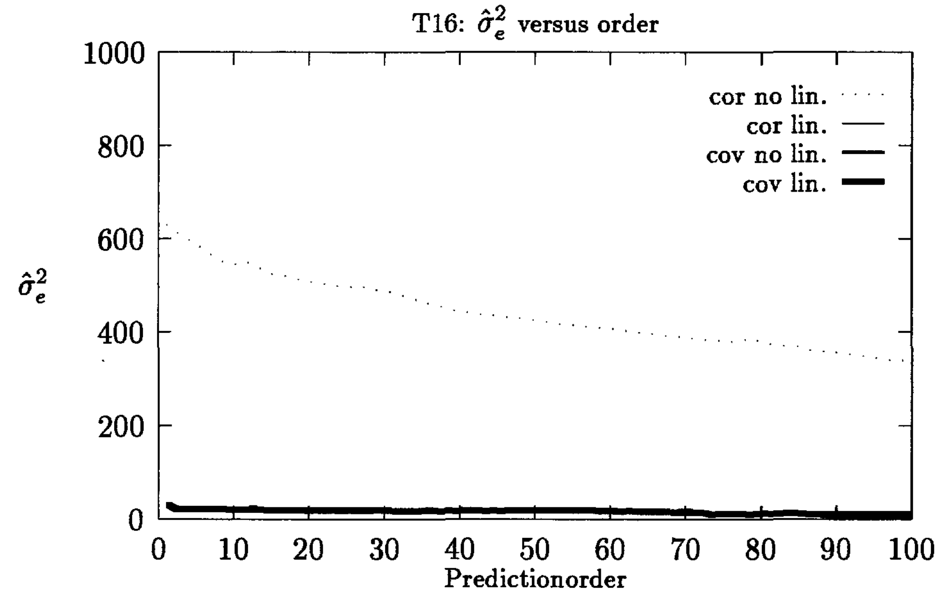

Figure A.2: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the order of prediction for the case, where the autoregressive parameters are determined using (cor) the *autocorrelation method*, and (cov) the *autocovariance method.* This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence *T16*, using the *SHS* pitch determination method. Only the pitch estimates shown in Figure 4.3 are used. From this a **restoration** was made, using three iterations. The lines (lin.) and (no lin.) for (cov), as well as the line for (cor lin.) almost coincide and can therefore not be distinguished.

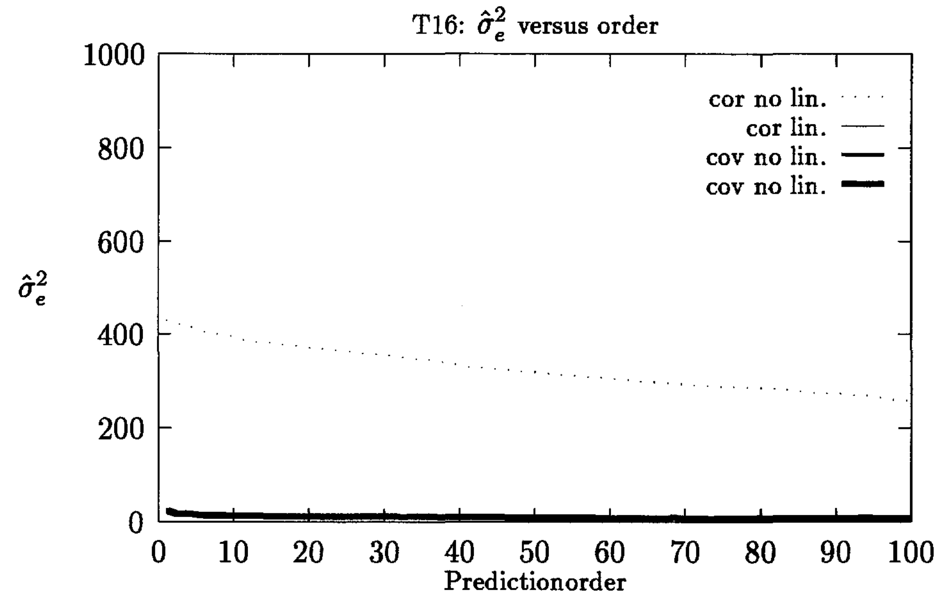

Figure A.3: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the order of prediction for the case, where the autoregressive parameters are determined using (cor) the ***autocorrelation method***, and (cov) the ***autocovariance method.*** This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence *T16*, using the *PDT* pitch determination method. Only the pitch estimates shown in Figure 4.5 are used. From this a **restoration** was made, using three iterations. The lines (lin.) and (no lin.) for (cov), as well as the line for (cor lin.) almost coincide and can therefore not be distinguished.

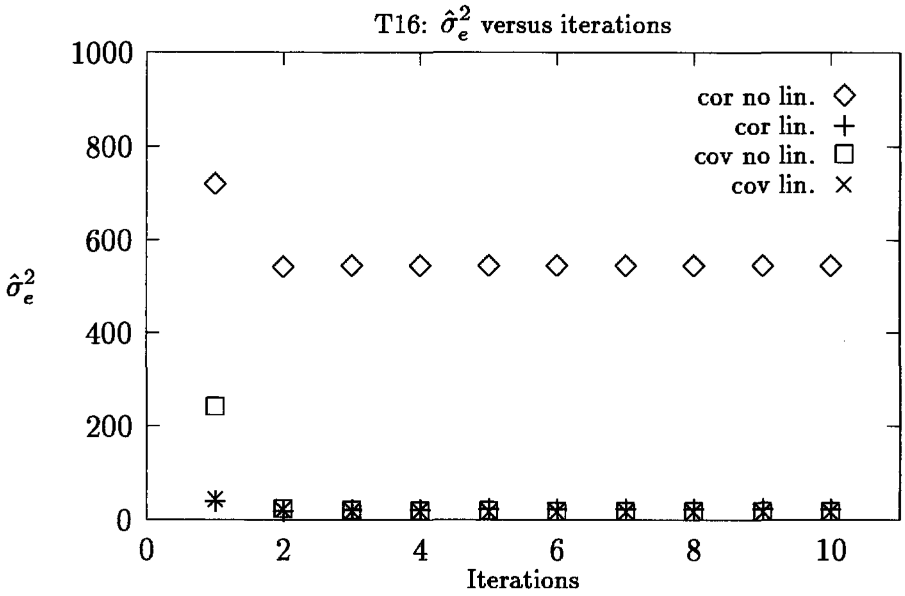

Figure A.4: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the number of iterations made in restoring the pitch contour. This is done for case, where the autoregressive parameters are determined using (cor) the ***autocorrelation method***, and (cov) the ***autocovariance method.*** This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence *T16*, using the *SHS* pitch determination method. Only the pitch estimates shown in Figure 4.3 are used. From this a **restoration** was made, using a prediction order of ten. The hill in the graph for (cor. lin.) is due to the fact that the restoration method can not be written as an iterative minimization of one quadratic expression.

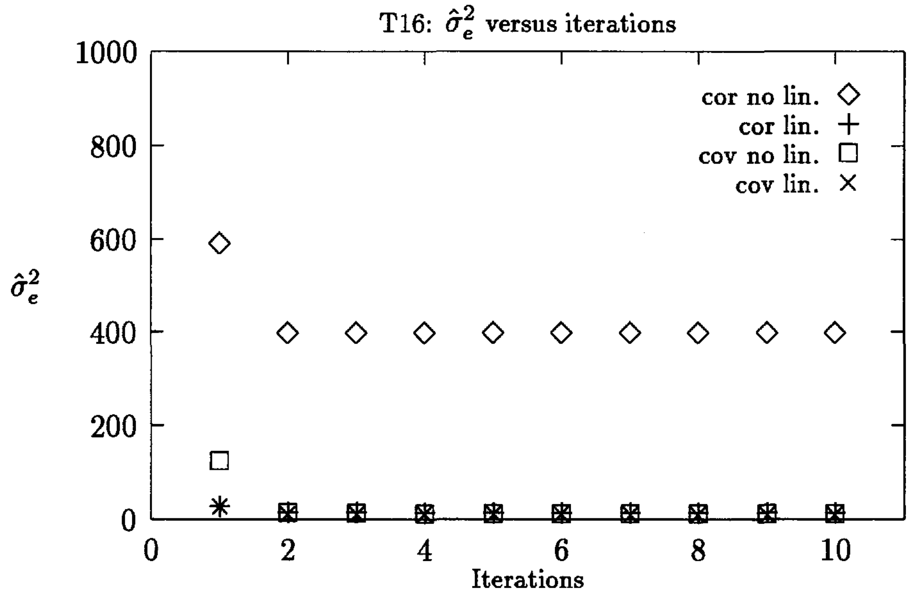

Figure A.5: Graph of the estimate for the excitation-noise variance of the autoregressive filter against the number of iterations made in restoring the pitch contour. This is done for case, where the autoregressive parameters are determined using (cor) the ***autocorrelation method***, and (cov) the ***autocovariance method***. This is done (no lin.) on the plain data, and (lin.) after subtracting the declination. The data points used, were determined from the sentence *T16*, using the *PDT* pitch determination method. Only the pitch estimates shown in Figure 4.5 are used. From this a **restoration** was made, using a prediction order of ten.

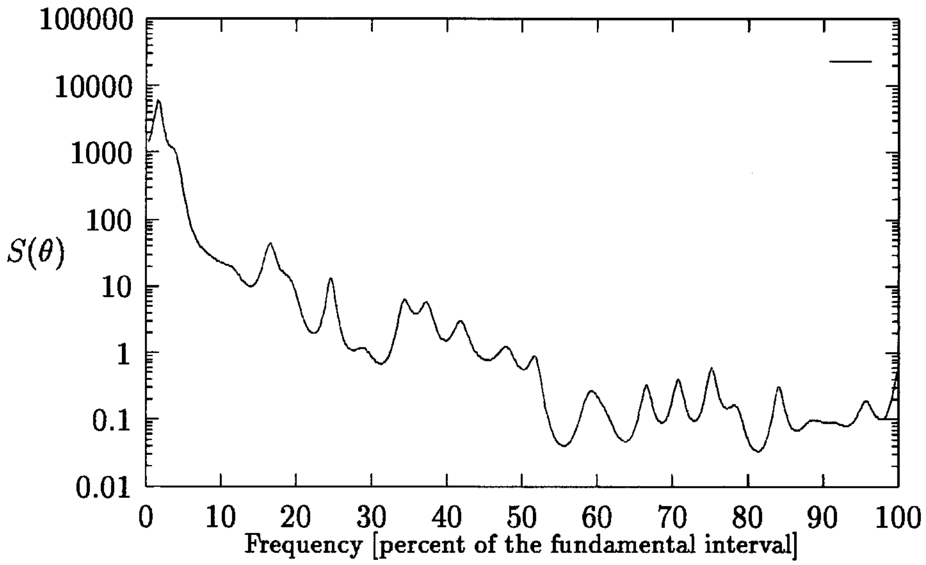

Figure A.6: Power density spectrum of the pitch contour of T16. The power density spectrum was determined from the estimates for the forward prediction coefficients. The order of prediction was 50. The pitch contour was determined using the PDT pitch determination algorithm. Here, before the spectrum was estimated, the linear part was removed from the pitch contour.

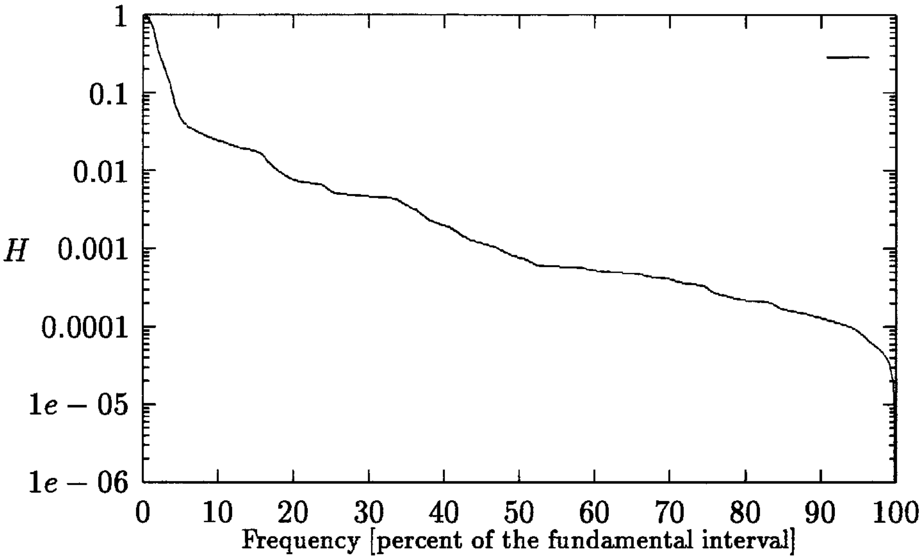

Figure A.7: Fraction of the energy in the high-pass region of Figure A.6 as a function of frequency.

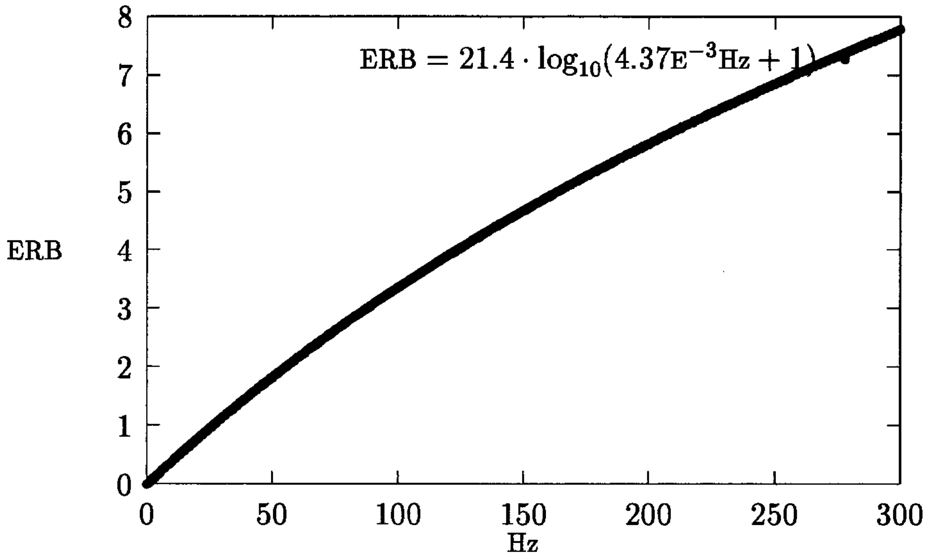

Figure A.8: Graph of the transformation of the Hz scale into the ERB scale. Note the almost linear behavior for these low frequencies.

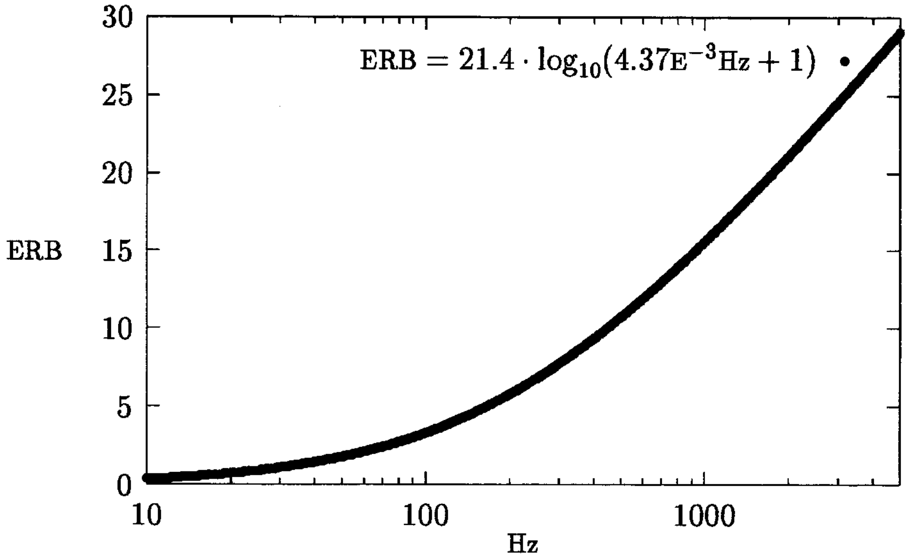

Figure A.9: Graph of the transformation of the Hz scale into the ERB scale. Note the almost linear behavior for low frequencies, while there is an almost logarithmic behavior for the high frequencies.

# References

[1] P. Delsarte, A.J.E.M. Janssen, and L.B. Vries. Discrete prolate spheroidal wave functions and interpolation. *SIAM J. Appl. Math.*, 45:641-650, 1985.

[2] B.R. Glasberg, B.C.J. Moore. Derivation of Auditory filter shapes from notches-noise data. *Hearing Research.*, 47:103-138, 1990.

[3] G.H. Golub and C.F. van Loan. *Matrix Computations.* North Oxford Academic Publishing, Oxford, England, 1983.

[4] D.J. Hermes. Measurement of pitch by subharmonic summation. *J. Acoust. Soc. Am*, 83(1):257-264, 1988.

[5] A.J.E.M. Janssen, R.N.J. Veldhuis, and L.B. Vries. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Transactions on ASSP*, 34(2):317-330, 1986.

[6] A.J.E.M. Janssen and L.B. Vries. Interpolation of band-limited discrete-time signals by minimizing out-of-band energy. In *Proceedings ICASSP-84*, San Diego 1984.

[7] S. Lawrence Marple, Jr. *Digital spectral analysis with applications.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[8] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Book Company, Tokyo, 1965.

[9] R.N.J. Veldhuis. *Adaptive Restoration of Unknown Samples in Discrete-Time Signals and Digital Images.* Eindhoven Druk, Eindhoven, 1988.

[10] E. Terhardt, G. Stoll, and M. Seewann. Algorithm for extraction of pitch and pitch salience from complex tonal signals. In *J. Acoust. Soc. Am.*, 71, 679-688, 1982.

[11] L.L.M. Vogten. *Syllabus van het college Spraaktechnologie 0H050.* IPO, Eindhoven 1988.