Table of contents

Table of figures

List of tables

I. Introduction

My diploma work was devoted to the development of a software *Harmonic Plus Noise* (*HNM*) for the acoustic unit concatenative speech synthesis. It aims on the one hand the development of a new diphone-based voice based on the Festival speech engine and on the other hand, the improvement of the Hidden Markov Model-based speech synthesis that allows introducing emotions and personal features in the synthetic speech. It seems indeed that the recognition and synthesis speech models tend today towards convergence, and IDIAP, specialized in speech recognition, tries consequently to focus its research in the second domain.

Its carrying out is inserted into an Erasmus exchange with the Ecole Polytechnique Fédérale de Lausanne (EPFL, Switzerland), which permits to insert me in the IDIAP (*Institut Dalle Molle d'Intelligence Artificielle Perceptive*) institute, a research centre associated to the EPFL and based in Martigny (Switzerland) under the management of Dr Hervé Bourlard, director of the site and professor at the EPFL. My local supervision was been in charge by Drs John Dines et Jithendra Vepa, who gave me support during all this project.

This report develops first the general description of the TTS ("Text-To-Speech") engines. Its explains next different methods for the speech waveform generation and insert the *Harmonic Plus Noise* model into. It shows then the advantages and drawback of this synthesis which allows reaching a high quality of artificial speech and also a speech coding.
In a first time, a theoretical study is broached. It is based on the thesis of Yannis Stylianou, author of the model and on more recent articles that have come completing it of the last advances in this domain. This work is also enriched by information that Yannis Stylianou had the kindness to bring me by mail. This study is illustrated of a lot of figures which will help me to avoid all ambiguity (among other things the different time instants that intervene in this model).
In a second time, it is a practical approach treating with the HNM implementation that is developed. This chapter, which constitutes an essential point of my work, shows the concrete running of this model, with encountered difficulties and their solving. It develops on the other hand some modifications compared with the theoretical approach. The HNM waveform synthesis is based on the architecture of Festival that provides the diphones and necessary prosodic modifications.
The results obtained with this implementation are then presented.

To put an end at this project, a more research part is broached. Its concerns the introduction of the *Harmonic Plus Noise* model into the Hidden Markov Model-based speech synthesis, which permit the addition of personal features as emotion and styles in the artificial voice. The current synthesizers allows to realize only a neutral voice, but of high quality. The HMM-based synthesis can modify this neutral feature by particularizing the voice, but remains of average quality since it is based on filter. The *Harmonic Plus Noise* synthesis could consequently constitute a interesting advance by combining the high natural voice with the personalization of the speaker introduced by the HMMs. Some integration avenues of the HNM waveform synthesis into the HMM-based speech synthesis are developed at the end of this report

At the same time of the technical background I learned in the IDIAP institute, this Erasmus exchange constituted a rewarding cultural experience since this research centre groups together a lot of nationalities among its researcher workers ; it was so possible to mix, as well as the Swiss society, with a great number of other cultures and different outlooks. On the other hand, English language so constituting the common language of the institute, I could improve it with interest, through my contacts with my supervisors and other researchers, the translation and the oral presentation of this work.

II. Framework of the project

In our society, the speech recognition and speech synthesis represent today a field of a great importance : the interaction with the machine is often wished to be faster and faster and close to the human oral language. It is true that the manipulation of the keyboard becomes a tedious tool in our consumption society where we would like everything / rapidly /. So, it is nice to imagine the facility, the flexibility and the time saving that the possibility of interaction with our computer as with a human being should mean, and which new possibilities it should bring…

Today, the synthesis quality and recognition rate are so that commercial applications are following from them. So, for speech synthesis, we can cite Telecommunications, Multimedia and Automobile with for example for Telecommunications, the vocalization of SMS, the reading of mails, the phone access to fax and e-mails, the consulting of databases, automatic answerphones (ex : Chrono Post) ; for Multimedia, the speech interface between man and machine, the help for teaching reading or / new languages (educational tools and / software), the help for teaching reading to blind people, bureaucratic tools ; and at last for the Automobile, the alert and video surveillance systems, the Internet access among others for the mail reading. Some companies as Nuance, Scansoft and Acapela-Group.are present on the business market.

This diploma work concerns the speech synthesis (Text-To-Speech, TTS). It aims at the development of software implementing a speech processing model called *"Harmonics Plus Noise Model*" (HNM). It's in fact a hybrid model since it decomposes the speech frames into a harmonic part and a noise part. It normally has to produce a high quality of artificial speech. Some projects have to ensue from this HNM synthesis at the IDIAP institute, in particular the creating of a new diphone-based voice based on the architecture of Festival and another one based on the Hidden Markov Models, which enable to produce a personalization of the speaker. These two points are also discussed in this work.

To define the framework of this project more exactly, I would like to briefly remind you of the functioning of a speech synthesizer.

III. Description of a TTS synthesizer

A TTS (Text-To-Speech) synthesizer is made up of different units each one carrying out a particular function [10,11,12]. We distinguish a processing language system, followed by a letter-to-sound (phonetization) unit and a prosody targets computation unit, and finally ended by the speech waveform generation. The whole system is represented on figure 1.

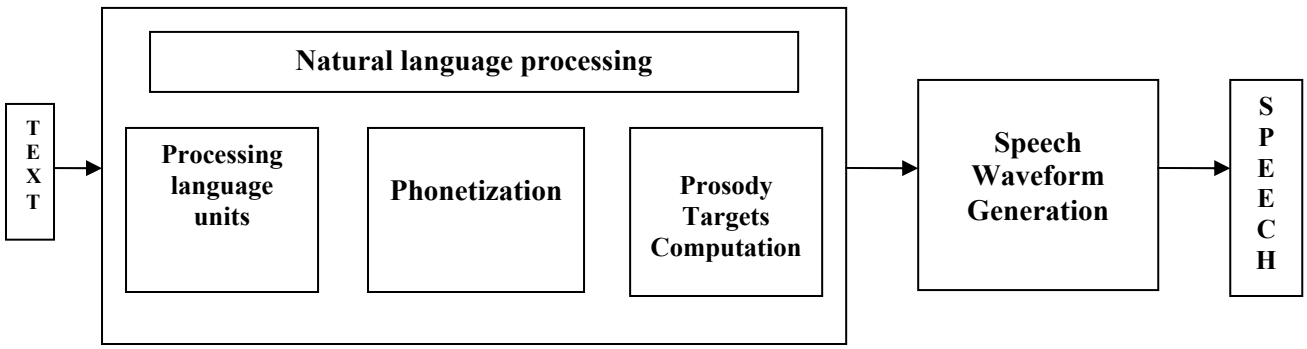


Figure 1 : General system of a TTS synthesizer

The step of *phonetization* consists into the decomposition of the input text into phonemes. This first step appears simple at first sight but is actually complex. According to the context, it is true that we can have several phonetic transcriptions for the same word (for example, “The” is pronounced differently if the following word starts with a vowel or with a consonant. The heterophonic homographs also pose a problem (so words which are pronounced in a different way are there written like “the”) ; proper nouns and the phonetic liaisons.

The unit following the *phonetization*, concerns the *prosody targets computation*. We mean by this term the intonation and the duration. For each group of words previously computed, these two parameters have to be estimated.

The complexity of this step and for the prosody targets computation (developed below), justify the previous module, namely the *processing language units*. This module is composed of 4 successive units : the *pre-processing*, the *morphological analyser*, the *contextual analyser* and the *syntactic-prosodic parser* (*hierarchical description)*. Each one is useful to facilitate the steps of *phonetization* and *prosody targets computation*.

The *pre-processing* unit aims at detecting the end of sentences and the abbreviations. A full stop does not necessarily mean the end of a sentence and the translation of an abbreviation is different according to the context. We also detect the numbers and the acronyms. The *morphological analyser* is useful to assign part-of-speech labels to words. Many hypotheses can be put forward and the purpose of the *contextual analyser* will be to eliminate possibilities in order to retain only one label per word. At last, the *syntactic-prosodic parser* permits to recognize the group of words which are together ; what is important for the intonation rhythm. All these data are sent to the letter-to-sound module and to the prosody target computation whose task is now reduced.

The last unit deals with the *speech waveform generation*. On the basis of the phonetization and the computed prosody, the synthesizer has to generate the correct signal. Many approaches are applicable.
The first one is based on rules, we also call it the *formant synthesizer*. These rules are adjusted by the phoneticians on the spectrograms they are able to decode. They can plot during time the formants evolution, which are stable on a period of about 100 ms. The quality obtained in this way is pretty average but some of these synthesizers are still sold today.
The second one is based on the concatenation of acoustic units. The idea is appeared in the seventies years and has led to a better result. We indeed notice that the main difficulty in the artificial voice production is the coarticulation mimicry, that is to say the physical vocal tract mimicry. Considering the inertia of this one, a sound will have a different spectrum according to the context, that is to say one or more sounds which / come before and latter. Consequently all the phonemes act upon one another and it is not sufficient to fetch / one after another in a database and concatenate them (we should obtain an incomprehensible speech). It is this kind of synthesis which will kept us us busy along this training course.

By simplifying the previous model and taking as hypothesis that only the last sound will act upon the next one, we come to the *diphone-based speech synthesis*. A diphone groups together two phonemes, or more exactly the second half of the first phoneme (cut on its stable area), the transition (quite difficult to generate by signal processing) and the first half of the second phoneme (cut on its stable area). For example, the diphone “e_g” will contain / half of the phoneme “e”, the transition between the “e” and “g” and / half of the phoneme “g”. The influence of the “e” on the “g”, that is to say the inertia of the vocal tract during the transition and during the pronunciation of the “g” is taken into account and the coarticulation is so respected (under the hypothesis).

The perfecting of such a system assumes the segmentation of / diphones as a whole (peculiar to each language) on their stable area (middle on the phoneme) and the recording of them in a database. This one contains so a unit representative of every diphone and from the result of the text phonetization, these last ones are put end to end to generate the resulting speech signal. However, it is unlikely that the computed duration and intonation (by *prosody targets computation*) correspond to the original

ones of the stored diphone since this one was recorded in a different context. A speech processing is then needed and this is at this point that my work comes up. The *Harmonic Plus Noise Model* (HNM) is a signal processing that is able to modify the parameters of the encoded voice and to provide a high quality of speech (in relation to the method, so for example the diphone-based synthesis).
Let's cite three previous approaches : filtering by Linear Prediction Coding (LPC, 80's years), PSOLA (developed by France Telecom around 1990) and MBR-PSOLA (developed by the Faculté Polytechnique of Mons around 1993). These are summarized in the next point.

We can complicate the diphone-based model by considering that several adjacent sounds influence the current one. The inertia of the vocal tract is thus estimated with a more important "history". The Unit Selection-based synthesis is resulting, that is a synthesis where recorded acoustic units are longer than the diphone and multi-represented in the database (so registered in different context), the idea is to touch / to the signal the least possible in order to improve yet on naturalness ; the Viterbi algorithm based on cost minimization (target cost and concatenation cost) is running for the choice of the different representatives. Again a signal processing as HNM will be necessary because some discontinuities are always remaining at the concatenation points. Nevertheless we can understand that the HNM modifications will be smaller since the presence of multi-represented units precisely permits to avoid as much as possible this processing by the concatenation of prosodic close units. Ultimately a gigantic database would allow storing all the possible representatives and it would avoid any speech signal modification.

Other advantages of the HNM model will be presented during the development of this model (point **IV.4**).

The databases necessary to the diphone-based synthesis take up about 5 Mbytes of memory, when the ones necessary to the unit selection-based synthesis need between 150 Mbytes and 1500 Mbytes according to the number of stored representatives. The HNM synthesis developed in this project is based on the natural language processing modules of *Festival*. The free databases are diphone databases (one unit per diphone) ; and the diphones-selection algorithm is also based on uni-represented diphones. Consequently, the HNM synthesis will be a diphone-based synthesis, although nothing stops its application on a unit selection-based synthesis. The chosen voice is English, so all the matter developed in this project is a priori valid for this language.

Let us note that the previously introduced algorithms, used for the different units of the natural language processing, are / most frequently based on training techniques (decision trees, neural networks,…) since the rule-based algorithms cannot provide such a good quality (but it remains dependent on the language).

Finally, let indicate that the current synthesizers provide a very good quality of speech although some mistakes are remaining. They carry out a neutral intonation, without any pragmatic information like mood, humour and emphasis[1]. We will see that the HMM-based synthesis will be able to provide a solution to this problem.

[1] An emphasis can even so be introduced by a marker language as "Speech Synthesis Markup Language" (SSML) ; but this way imposes a preliminary knowledge of the text to synthesize.

IV. Speech waveform synthesizers

IV.1 Synthesis by Linear Prediction Coding (LPC)

IV.1.1 LP Analysis and synthesis

The *synthesis by Linear Prediction Coding* (LPC) decouples the modelling of the excitation (air forced throughout the glottis) and vocal tract [10,11,12,14]. The excitation is represented either by a Dirac impulse train (voiced frames) or by white noise (unvoiced frames). For voiced frame, the distance between two impulses is equal to the fundamental period. The vocal tract is modelled by an autoregressive filter (all poles).

The analysis step thus consists of classifying the analysis frames as voiced or unvoiced, then calculating the pitch (for voiced frames) and finally the coefficients of the filter (and the associated gain).
The synthesis step consists of the retrieval of these parameters stored in a database and of the computation of the frame of speech by filtering the appropriate excitation by the LPC filter. The signal frames are then concatenated (actually synthesized by a OverLap and Add process, OLA) to produce the resulting speech.

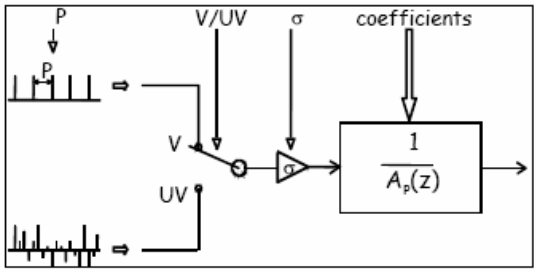


Figure 2 : Synthesis by Linear Prediction (LPC)

The autoregressive filter tries to represent the formants of the speech signal, which are modelled by the resonances introduced by the pairs of complex conjugate poles of the denominator A(z). The train of Dirac impulses tries to reproduce the periodic glottal opening/closure (and will be useful as excitation for sounds like vowels). The white noise constitutes our excitation for unvoiced frames modelling airflow through the constricted glottis.

The speech signal is not stationary (fortunately, otherwise it wouldn't contain any information). However, it can be observed that physiological constraints impose on it is a maximum rate of articulation (in general 30 ms). So we can speak about a pseudo-stationary signal. Generally this important property allows a lot of simplifications in the analysis algorithms and a more efficient coding thanks to interpolation possibilities.

So, typical frame lengths will be equal to 30 ms. These are separated of 10 ms to assure a sufficient overlap and modification frequency of the excitation and of the parameters of the filter during the synthesis process. At last, we shall weight the frame by a hamming window in order to take into account of its local characteristics (maximize its influence at its centre). See figure 3.

Figure 3 : Analysis frames weighted by a Hamming window

The coefficients of the autoregressive filter (AR) are obtained from an LP analysis (Linear Prediction). The method for obtaining this filtering is to use the autocorrelation function (temporal domain). The computation of the AR parameters is realized in such a way that it minimizes the variance (the energy) of the output signal (residual signal) which corresponds to the quotient between the spectrum of the original signal and the envelope obtained from the filter.

The coefficients are determined by the Yule-Walker equations :

$\sum_{j=1}^{P} a_j \phi_x(i-j) = -\phi_x(i)$ for $i = 1,...,P$; where $\phi_x(i)$ represents the autocorrelation function on i points.

In matrix notation :

$$\begin{pmatrix} \phi_x(0) & \phi_x(1) & \dots & \dots & \phi_x(P-1) \\ \phi_x(1) & & & & \dots \\ \dots & & & & \dots \\ \dots & & & & \dots \\ \phi_x(P-1) & \dots & \dots & \dots & \phi_x(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \\ \\ a_P \end{pmatrix} = - \begin{pmatrix} \phi_x(1) \\ \phi_x(2) \\ \dots \\ \dots \\ \phi_x(P) \end{pmatrix}$$

Note that the first coefficient is fixed : $a_0 = 1$.

The prediction order P representing the number of resonances created by the filter (from 0 to F_e Hz), chosen based on the sampling frequency in accordance to the heuristic law : $P_{opt} = 2 + F_e(KHz)$, which corresponds to one formant per kHz of band pass. The factor 2 represents the 2 poles introduced by the glottal opening/closure. For example we use an order of 18 for speech sampled at 16kHz.
Also notice that the matrix is a Toeplitz matrix, which means that fast algorithms as Levinson or Schür may be applied for the linear system resolution in order to reduce its complexity from p^3 to p^2.

Let us consider the following example (a temporal frame) (figure 4.a) to which we apply the LP analysis and verify the complex conjugate poles of the filter (symmetric with the horizontal axis) in the complex plane (figure 4.b).

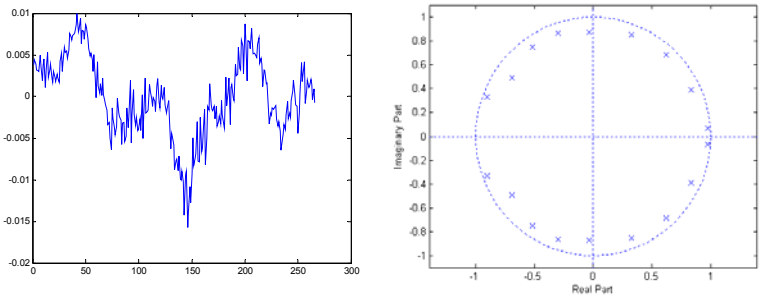


Figure 4 : Temporal frame (left), number of samples for X-axis and amplitude forY-axis ; complex conjugate poles in the complex plan (right)

We know that the pairs of complex conjugate poles near the unit circle create resonance at the frequencies where the poles appear hence the resulting amplitude envelope (see figure 5.a) is expected. Superposing the amplitude envelope to the amplitude spectrum of the original signal (figure 5.b), we notice that the formants are correctly modelled by the filter.

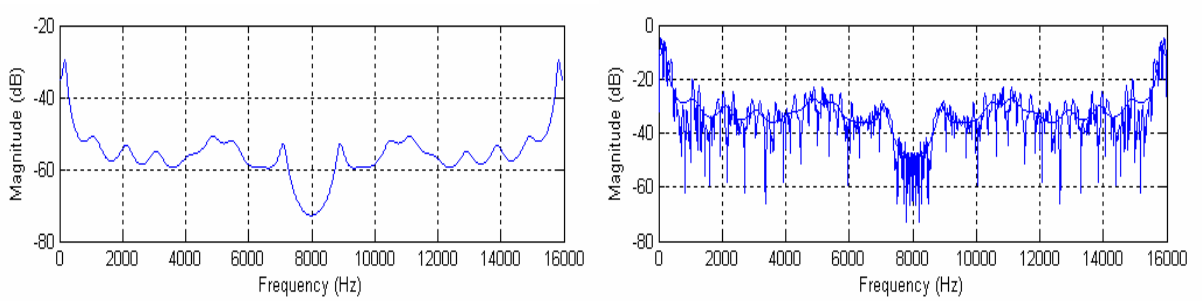


Figure 5 : Amplitude envelope obtained by the LP analysis (a) and superposing with the original amplitudes envelope (b)

If the filter is excited by the residual signal obtained during the LP analysis, we exactly regenerate the original signal. We get then to the RELP synthesis (Residual Excited Linear Prediction).To simplify synthesis, this excitation can be replaced by a Dirac impulse train or by white noise.

Looking again at the previous example and adding the correct excitation (white noise because our frame is unvoiced).

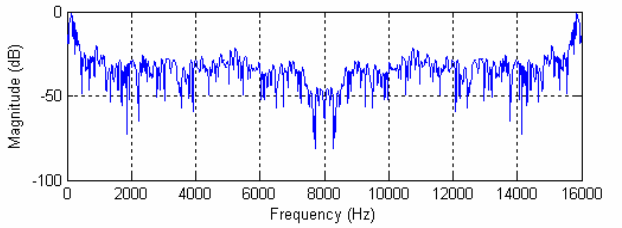


Figure 6 : Spectral frame synthesized by LP

During the concatenation of diphones, prosodic modifications have to be introduced. The modification of the pitch is made by modifying the distance between the Dirac impulses (fundamental period). The duration is changed by the repetition of the omission of frames.

IV.1.2 Drawbacks of the LPC synthesis

There are 3 major problems of the LP analysis : it cannot model the antiformants (like 'm' or 'n') because the filter doesn't possess any zero (in its numerator). These antiformants are obtained by the emission of the speech by the oral and nasal tracts at the same time. These tracts have a different length, so the signals travel through ways whose length is different which causes phase oppositions and decreases the frequency content on some bands. This phenomenon is illustrated at figure 7.a.

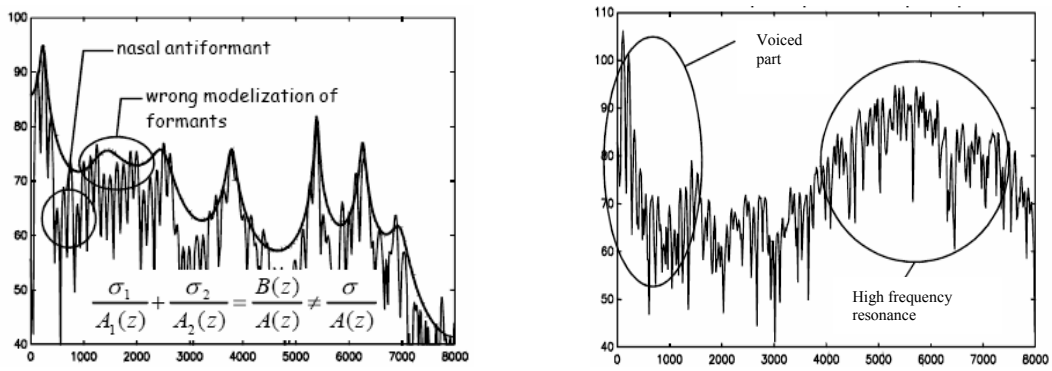


Figure 7 : Bad modeling of the antiformants by the LP model (a) and mixed sounds (b)

The second problem occurs with the mixed sounds (like ‘v’). That is to say sounds which possess important voiced and unvoiced frequency components. For these sounds, the vocal cords vibrate (voiced part, low frequencies) but don’t totally close and a stochastic signal appears in high frequencies (unvoiced part). Well, the LP synthesis separately considers voiced and unvoiced frames. To correctly model these sounds, the excitation has to be modified by a weighted sum of white noise and an impulse train, which increases the complexity of the model, which is then called "hybrid" or "mixed-excitation".

Lastly, the LP model decouples the modelling of the excitation and the vocal tract, which explains the average quality it can provide. We will see that HNM allows solving this problem.

IV.2. TD-PSOLA Synthesis

IV.2.1 TD-PSOLA Synthesis

The TD-PSOLA (Time Domain Pitch Synchronous Overlap and Add) synthesis [8,10,12,23], well known for its simplicity and efficacy, was developed in the France Telecom laboratories at the beginning of the 90’s. This algorithm doesn’t use speech coding but directly acts on the original samples. It is based on a pitch synchronous division of the speech signal, that is to say in frames whose length corresponds to a fundamental period and whose centre is taken up by the burst of the signal (maximum energy). This is illustrated in figure 8 (left). The synthesis is processed by overlap and add (OLA) and by weighting frames with a Hanning window which allows to reduce discontinuities during the overlap thanks to its zero value and its zero derivative value at its extremities. The length of the window approximately corresponds to two times the fundamental. Figure 8 also emphasizes the Poisson’s sum formula, that is the inverse of the sampling theorem. From top to bottom, the right part shows the spectrums of the entire temporal window (of the left part), of one temporal window and of the entire temporal window. This theorem applies a discretization in the frequency domain (emphasized from the second to the third graphs), which is expressed by a duplication (sum) of the temporal frame in question.

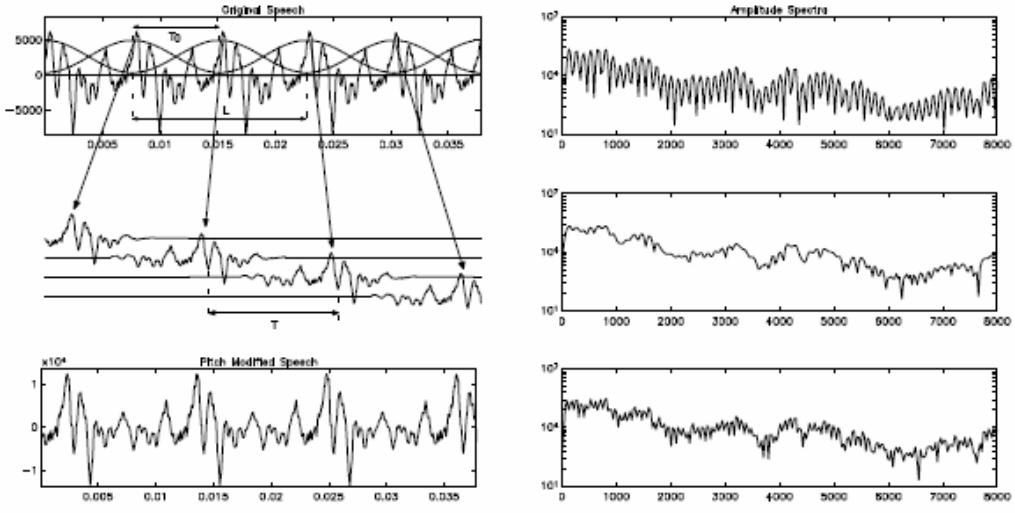


Figure 8 : Principle of the TD-PSOLA synthesis ; temporal (left) and frequency (right) domains

The prosodic modifications are brought about by the following way : a pitch changing will compress or expand the space between the frames. The duration will be modified by duplicating or removing synthesis frames. At last, a gain modification can be applied by modifying the scale of the weighted window. We have to pay attention that these prosodic modifications don’t introduce any additional computation, which constitutes an important advantage of this method.

IV.2.2. Drawbacks of the TD-PSOLA synthesis

TD-PSOLA synthesis appears thus very simple, nevertheless a first problem arises form applying the same prosodic modification algorithm to the unvoiced frames, which can decrease the speech quality. Additionally some problems can be observed at the concatenation points during the synthesis process. We distinguish three types of discontinuity : the pitch, phase and spectral envelope discontinuities, as is illustrated in figure 9. The upper part of the graph represents the PSOLA analysis on the left acoustic unit ; the middle one, the same analysis on the right acoustic level ; and lastly the bottom part shows the result of the concatenation by the OLA process.

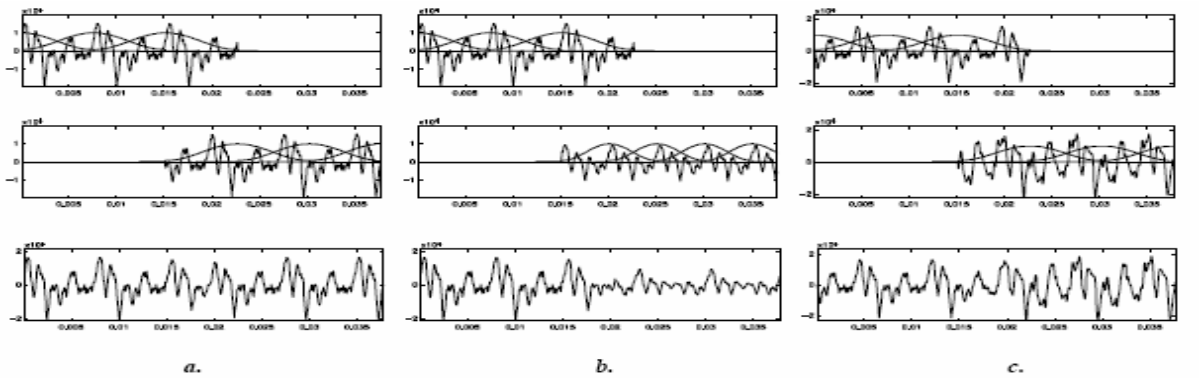


Figure 9 : Phase discontinuity (a), pitch discontinuity (b), and spectral envelope discontinuity (c) introduced by the TD-PSOLA algorithm

Phase discontinuity

The figure 9.a illustrates the phase discontinuity. The speech frames are identical; but the analysis window is not centred on the same position relative to the time of glottal closure. This can be due to mistakes in the evaluation of the fundamental frequency which is expressed as mistakes in the localization of the glottal closure instants (pitch marks). Consequently, a phase mistake will be perceived during the overlap of the frames at the concatenation point. Hence the evaluation of the pitch has so to be very accurate, which needs important recourses, indeed a manual operation.

Pitch discontinuity

This is illustrated in figure 9.b. The concatenation segments have identical spectral envelopes but the pitches are different. If the pitch of the left segment is chosen, the distance between frames of the right acoustic unit is modified, which is expressed by a modification of the spectral envelope of the concatenation frame because of the OLA process. This cannot so be properly realized if the pitch difference is very different on either side of the concatenation point. This factor can be minimized by using the professional speakers to read the corpus with a constant pitch, but this remains a difficult task and results in unnatural sounding speech.

Spectral envelope discontinuity

This last discontinuity is shown in figure 9.c. The left and right frames don't possess the same spectral envelope and a discontinuity appears at the concatenation point. This is due to phonemic variability of the speaker and to coarticulation effects and constitutes one of the biggest drawbacks of concatenative speech synthesis.

Some improvements of the TD-PSOLA synthesis have been developed in order to counter these drawbacks. For example FD-PSOLA (Frequency Domain Pitch Synchronous Overlap and Add), the frequency version of TD-PSOLA and MBR-PSOLA (Multi-Band Re-synthesis Pitch Synchronous OverLap and Add), known as MBROLA.

IV.3 MBROLA synthesis

IV.3.1 MBROLA synthesis

To counter the previous drawbacks the database should exhibit the following features :

- Every word should be pronounced with the same pitch.
- The analysis windows should be centred on the same relative position.
- Spectral interpolations should be possible at the concatenation points.

The basic idea of the MBROLA synthesis [8,9,10] consists of re-synthesizing the database segments in order to provide to the TD-PSOLA synthesis speech frames that don't have any discontinuity during the concatenation process and to allow interpolations so as to smooth the spectral discontinuities.

These segments are re-synthesized from a hybrid analysis carried out in bands (MBE, Multi-Band Excitation) as illustrated by figure 10.a. Only the voiced segments are bound by this analysis ; the unvoiced frames are unmodified. The MBE analysis consists of dividing the whole spectrum into bands (a band centred on each harmonic) and in estimating the pitch and the harmonics by a global and local mean-squared criterion, respectively. Each band is submitted to a voicing decision which aims to decide if the band contains a harmonic or not. During this analysis process two envelopes are so estimated : one voiced and one unvoiced.

In order to provide identical speech frames, the MBE synthesis is performed with constant pitch and a constant origin phase strategy. These phases are arbitrarily chosen. This synthesis rebuilds the voiced signal by re-sampling the voiced envelope at the new frequencies of the harmonics and by the temporally summing of these. The unvoiced samples that represent the unvoiced bands are obtained by the band-pass filtering (corresponding to the unvoiced bands) of a white noise which is then modulated by the unvoiced envelope. We get the resulting signal by a temporal sum of these two parts. This synthesis is called MBR-TDPSOLA (MULTI-BAND RE-synthesis TIME-Domain Pitch Synchronous OverLap Add). It combines the computational efficiency of TD-PSOLA with the flexibility of the MBE model.

These analysis and re-synthesis processes are carried out off-line and doesn't use any resources during the real-time synthesis process.

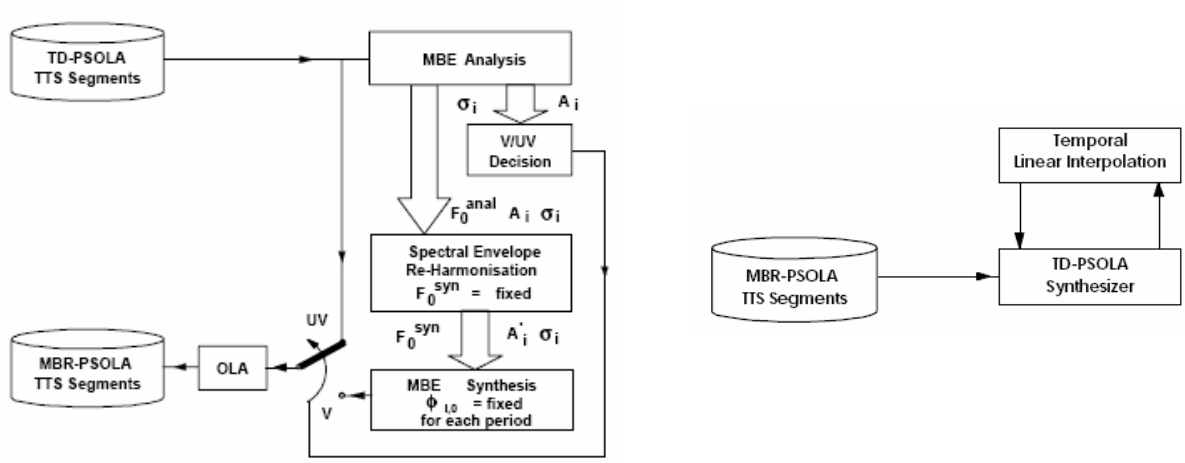


Figure 10 : Band analysis (MBE) and –re-synthesis of the segments (a). MBR-PSOLA synthesis (b)

The phase and pitch discontinuities are removed thanks to the MBE re-synthesis with a constant pitch and a constant origin phase strategy of the harmonics. Note that the pitch-marking (position of the closure glottal instants) is now automatic since we know the pitch and the phases at the origin (so we

know the relative position of the speech segment in the analysis frame). Another important advantage of the re-synthesis with a constant pitch and origin phases is that the interpolation of the complex envelope becomes equivalent to the temporal interpolation [8] (figure 10.b) because we now have similar frames and the spectral interpolation is always equivalent to the complex spectrum interpolation.
The resulting speech quality is quite similar to that obtained with PSOLA.

IV.3.2 Drawback of the MBROLA synthesis

Nevertheless this method doesn't take into account any additional noise component for the voiced segments resulting in less natural speech. In addition, the setting of constant phases at the origin can give a quite unnatural voice, however some improvements have been brought in the choice of the phase so as to obtain a more natural voice.

IV.4 Harmonic Plus Noise Model (HNM)

This section introduces the *Harmonics Plus Noise* Model (HNM) [1,2,3,4,5,6,7]. Its origin and the key concepts are explained and the advantages and drawbacks are set out. Then the HNM analysis process is developed, this one computes the parametrical representation of the speech signal which is stored in a database. It is important to note that this database will replace the database of the original signal. Finally, we will tackle the HNM synthesis, which modifies analysis parameters according to prosody targets to produce a high quality of speech.

IV.4.1 Principle of the HNM

The *Harmonics Plus Noise* Model applied to speech fits into the hybrid models that decompose the signal into a deterministic part and a stochastic part. This decomposition allows for the independent modelling and modification of the two parts. The deterministic part is represented by a time-varying harmonic component while the stochastic part is represented by a modulated noise.

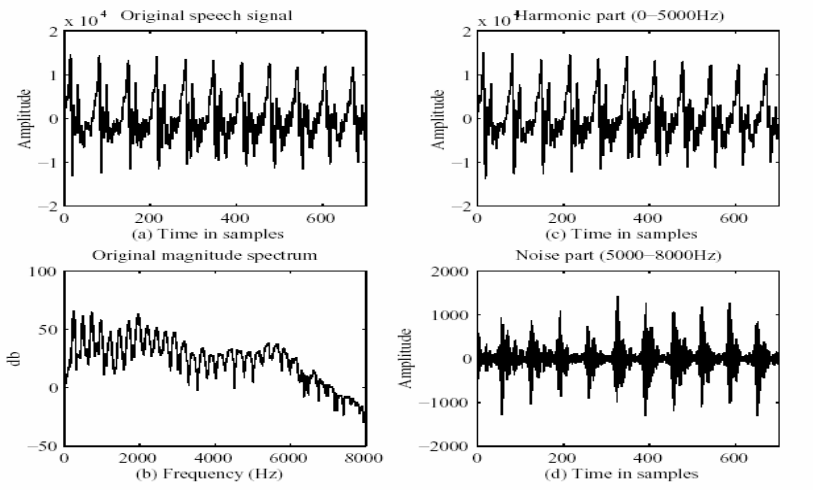


Figure 11 : Voiced signal decomposed into its harmonic and noise parts

To understand this procedure, let's analyse a voiced speech frame (sampling frequency of 16 kHz). The original signal is shown in figure 11(a) and its Fourier Transform in figure 11(b). We can clearly see in low frequencies of the spectrum (0-5000 Hz) the many peaks that represent the harmonics of the signal (the harmonics occur at multiples of the fundamental frequency f_0) ; the high part of the spectrum is dominated by noise. These two components are separated in the frequency domain and

computed again in the temporal domain (figure 11(c) for the low frequencies and figure 11(d) for the high frequencies).We find in the low frequencies again the harmonics of the signal ; which can be represented by a sum of sinusoids. The high frequencies represent the "noise part" of the signal ; it is remarked that the noise bursts are synchronous with the fundamental local period of the speech. This remark will be important in the modelling of the noise signals.

More precisely, this model decomposes the speech signal into voiced and unvoiced frames.

The voiced frames are themselves constituted by a harmonic part and a modulated noise part. The harmonic part accounts for the periodic components of speech and the noise part, the non periodic ones (obstructions in the vocal tract,…). This is modulated to respect the shapes of the noise signal, without forgetting the synchronisation of the noise bursts (see below). For example, the vowels and the fricative consonants (like 'z') will be decomposed into voiced frames. These two parts are separated in the frequency domain by the *maximum voiced frequency* F_M, which is a variable parameter from one frame to another. The spectrum is so divided into two distinct bands.

The low part of the spectrum ($f < F_M$) is represented by only a sum of sinusoids[2] :

$$\underset{\sim}{S}_{h}(t) = \sum_{-L(t)}^{L(t)} \underset{\sim}{A}_{k}(t).e^{jkw_0(t).t}$$

where L is the number of harmonics, A_k(t) and $k.w_0$ the time-varying amplitude (frame by frame) and the frequency of the k^{th} harmonic, respectively.

In contrast to dynamic models (HNM2 and HNM3), this HNM1 model is simple and facilitates (among other things) the smoothing process of the discontinuities during the concatenation of the acoustic units, ensuring at the same time a high quality of speech. So this model was chosen for speech synthesis.

The high part of the spectrum ($f > F_M$) is represented by modulated white noise *b(t)*. We model this part by a time-varying autoregressive (AR) filter *h(t)* and a temporal envelope *e(t)*. The noise part is represented by the following signal : $S_n(t) = e(t).\left[h(t) * b(t)\right]$.

Finally, we obtain for the voiced frames the following synthesized signal : $\hat{S}(t) = S_h(t) + S_n(t)$

Note : it is important to synchronize (by the temporal envelope) the noise part S_n(t) with the harmonic part S_h(t). If it is not done, these two parts will not be fused correctly and two distinct sounds will be perceived. But let's be careful about the bursts of the harmonic and noise parts which are (more or less) temporally aligned. Actually this alignement is generally manually confirmed but remains speaker-dependant. So a synchronous shift is characteristic of everybody and the temporal envelope will have to be adapted to the speaker.

The unvoiced frames are only represented by a noise part, so : $S(t) = \left[h(t) * b(t)\right]$. We don't use the envelope here because the signal has neither to respect a particular shape nor to be synchronised.

The HNM model supposes an analysis process on the different acoustic units recorded in a database. This analysis will replace this first database by a second composed of all the parameters generated by the analysis. Then the synthesis process will retreave these parameters and maybe modify them (if necessary), before synthesizing the new speech signal. Let's begin, before the description of the analysis and synthesis process, by summarizing the advantages and drawbacks of the HNM model.

[2] The model presented here is called HNM1, it is sufficient to provide a high quality of artificial speech. Other models (HNM2 and HNM3) can take into account dynamic characteristics of the speech signal by the amplitudes, A(t), depending on the time and changing in the same frame.

IV.4.2 Advantages and drawback of the *Harmonic Plus Noise Model*

We can summarize the advantages and drawbacks of the Harmonic Plus Noise Model by this way :

- The decomposition of the speech signal into a harmonic and a noise components allows for the production of more natural sounds, since the periodic and stochastic energy can be simultaneously conserved. We call thus a hybrid model.
- The modelling of the excitation and of the vocal tracts are here coupled in single system, in contrast to the filter-based models (like LPC). It is true that a decoupling introduces an inevitable loss of quality in the resulting speech signal. This decoupling imposes purely speech processing techniques (FFT,…) ; the hybrid model is then described as a phenomenological model.
- The parametric structure of this model permits simple prosodic modifications of the signal and also enables smoothing discontinuities at the concatenation points.
- It also allows a speech coding, with a bit rate that can reach 10kb/s.

The main drawback of this model remains in its computational load which is related to its complexity.

IV.4.3 HNM Analysis

The analysis consists of estimating the harmonic and noise parameters. By the decomposition into two independent parts, these are estimated separately. We first have to separate the voiced frames from the unvoiced frames and then compute the parameters used for the synthesis.
The steps of the analysis process are the following :

- *First Estimation of fundamental frequency*
- *Voiced/Unvoiced decision for each frame*
- *Estimation of the maximum voiced frequency* F_M
- *Refinement of the fundamental frequency*
- *Setting of the positions of the analysis time instants*
- *Computation of the amplitudes and phases of the harmonics for voiced frames*
- *Estimation of the complex spectral envelopes*
- *Parameters of the noise part*
- *Modification of the phase*

All these steps are realised for each frame of speech data on the first database. This is an off-line process, so the synthesis will not be affected by this computational load. The four first operations are carried out at constant analysis frame rate, when the five last steps are carried out pitch synchronously. The quality of the HNM synthesis is largely dominated by the accurate measurement of the pitch and the voiced decision. So we have to be careful for these both estimations.

IV.4.3.1 First Estimation of the fundamental frequency (pitch)

The first step consists of the estimation of the pitch f_0. This parameter is estimated every 10 ms. We will see that the length of the analysis window will depend on this local pitch. One method co-developed with the HNM algorithm is explained here. This method is based on an autocorrelation approach and is obtained by fitting the original signal with another defined by a pitch (sum of harmonics) in the frequency domain :

$$\varepsilon = \frac{\int_{-\frac{1}{2}}^{\frac{1}{2}} \left[|S_w(f)| - \left| \hat{S}_w(f) \right| \right]^2 .df}{\int_{-\frac{1}{2}}^{\frac{1}{2}} |S_w(f)|^2 .df}$$

Where $S_w(f)$ is the Short Term Fourier Transform of the speech segment S(t) (Blackman weighted segment whose length is equal to 3 times the maximum fundamental period $T_0^{\max}$) and $\hat{S}_w(f)$, the Short Term Fourier Transform of a purely harmonic signal obtained from a fundamental frequency F_0.

Ideally, the error spectrum should be flat (as a function of T_0) for a noise signal and we should obtain a maximum sensitivity in harmonics'areas for voiced frames. To avoid a greater sensitivity for low and high pitch, a corrective factor is introduced :

$$\varepsilon = \frac{\int_{-\frac{1}{2}}^{\frac{1}{2}} \left[|S_w(f)| - \left| \hat{S}_w(f) \right| \right]^2 .df}{\int_{-\frac{1}{2}}^{\frac{1}{2}} |S_w(f)|^2 .df \left[1 - \mathrm{T}_0 . \sum_{t=-\infty}^{+\infty} w^4(t) \right]}$$

where w(t) corresponds to the weighted window (typically a Blackman window whose length is 3 times the maximum T_0). By replacing integrals by summations, we finally obtain the following criterion :

$$\mathrm{E}(\mathrm{T}_0) = \frac{\sum_{t=-\infty}^{+\infty} s^2(t).w^2(t) - \mathrm{T}_0 . \sum_{l=-\infty}^{+\infty} r(l.\mathrm{T}_0)}{\left[\sum_{t=-\infty}^{+\infty} s^2(t).w^2(t) \right] \left[1 - \mathrm{T}_0 . \sum_{t=-\infty}^{+\infty} w^4(t) \right]}$$

where S(t) corresponds to the original signal.

The window *w(t)* is subject to the following constrain : $\sum_{t=-\infty}^{+\infty} w^2(t) = 1$.

With also, $r(k) = \sum_{t=-\infty}^{+\infty} s(t).w^2(t).s(t+k).w^2(t+k)$.

We go in the discrete time by replacing the time by the successive samples (n). We evaluate the error as a function to T_0 in the interval $\left[\frac{f_e}{f_{0\max}}, ..., \frac{f_e}{f_{0\min}} \right]$ - which corresponds to different T_0 converted into a number of samples – where f_e is the sampling frequency. The length is in practice limited by the number of samples of the weighted window ($3T_{0\ \max}$). Typical values of the maximum and minimum pitch are [60-230Hz] for male voices and [180-400Hz] for female voices.

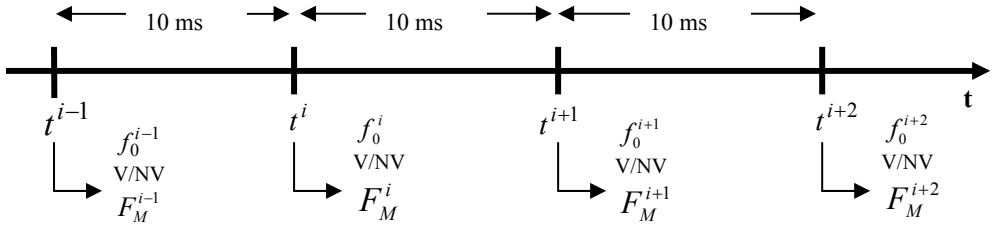


Figure 12 : Estimation of the pitch, V/UV decision and estimation of F_M

Note that the minimization of the error by the previous criterion is equivalent to maximizing the function : $\Psi(T_0) = T_0 . \sum_{l=-\infty}^{+\infty} r(l.T_0)$

Of course, the fundamental period is also present in the denominator of the initial expression, but it's only for correcting the bias introduced by the great and small values of T_0.
So there are many things in common with an autocorrelation method but the variable parameter here is the fundamental period.

Figure 13 gives examples of the application of the algorithm on voiced and unvoiced frames. E(p) is here normalized so that a zero value corresponds to a purely periodic signal and one to a purely noise one.

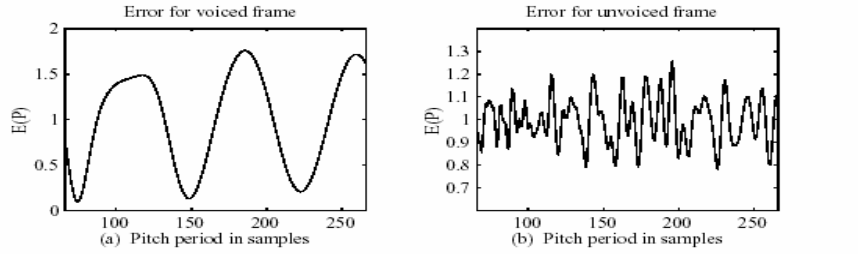


Figure 13 : $E(T_0)$ for a voiced and unvoiced frame

To avoid some pitch errors, a "peak tracking" method is needed. This has to look at two frames forward and backward from the current one. The minimum error path is found and by this way the pitch is associated.

IV.4.3.2 Voiced/Unvoiced decision

The frames extracted every 10 ms (whose length is always 3 times T_{0max}, see figure 12) have then to be classified as "voiced" or "unvoiced". We apply the Short-Term Fourier Transform (STFT) with a number of points N_{FFT} equal to 4096 (with zero padding) to the current frame and we call it S(f).
From this first STFT, we can evaluate the first four amplitudes of the harmonics (the first of which is the fundamental). We note $\hat{S}(f)$, which is a set of amplitudes of harmonics (of f_0).

The following criterion is applied :

$$E = \frac{\int_{0.7\hat{f}_0}^{4.3\hat{f}_0} \left(|S(f)| - \left|\hat{S}(f)\right| \right)^2 df}{\int_{0.7\hat{f}_0}^{4.3\hat{f}_0} \left(|S(f)| \right)^2 df}$$

The frame is declared "voiced" if E is less than the threshold of -15dB, and "unvoiced" otherwise.
Figure 14 shows the speech signal overlaid by voiced/unvoiced decisions which are taken manually (above) and the result given by the algorithm (below). We can see the successful application of the criterion.

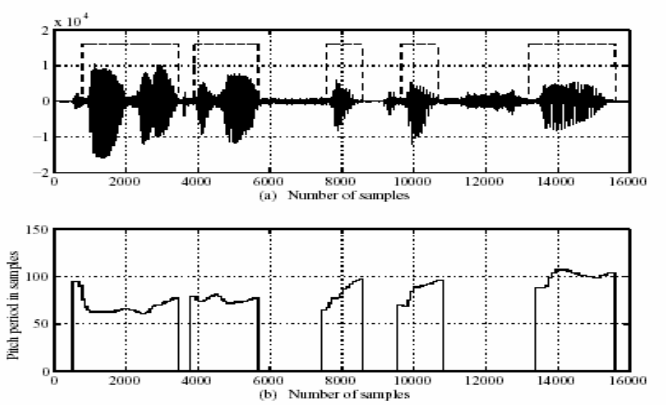


Figure 14: Speech signal overlaid by manually taken voiced/unvoiced decisions (upper) and voiced/unvoiced decisions produced by the algorithm (bottom)

To close this point, note that it's preferable to detect as voiced an unvoiced frame than the opposite.

IV.4.3.3 Estimation of the maximum voiced frequency F_m

This parameter is also estimated every 10 ms (see figure 12). At the beginning, we work in the interval $\left[\frac{\hat{f}_0}{2},\ldots,\frac{3.\hat{f}_0}{2}\right]$ of the absolute spectrum. We look for the greatest amplitude and the corresponding voiced frequency in this interval, which we denote A_m and f_c, respectively. We also compute the sum of the amplitudes (called the cumulative amplitude A_{mc}) located between the two minima around the greatest voiced frequency (see figure 15). The other peaks in the band are also considered (occurring at frequencies denoted by f_i) in the same interval, with the two types of amplitudes $A_m(f_i)$ and $A_{mc}(f_i)$. We compute the mean of these cumulative amplitudes, denoted by $\overline{A_{mc}(f_i)}$.

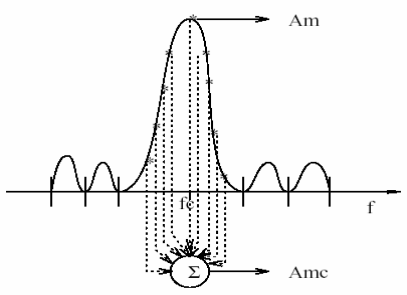


Figure 15 : Computation of the cumulative amplitude

Then, we apply the following test to the greatest frequency f_c :

If $\frac{A_{mc}(f_c)}{\overline{A_{mc}(f_i)}} > 2$ and $A_m(f_c)\text{-}\max\{A_m(i)\} > 13\text{dB}$

Then if $\frac{\left|f_c - L.\hat{f}_0\right|}{L.\hat{f}_0} < 20\%$ with L being the number of the nearest harmonic of f_c.

Then the frequency f_c is declared "voiced" and the next interval $\left[\frac{3.\hat{f}_0}{2},\ldots,\frac{5.\hat{f}_0}{2}\right]$ is considered and the same criterion is applied. The highest voiced frequency found will correspond to the maximum voiced frequency F_M. However, to avoid mistakes, a 3 points median smoothing filter is applied.

As well, the frequency F_M can vary greatly from one frame to the next. In order to reduce abrupt jumps we can also use another median filter on this time-varying frequency. Five points is in general used here.

Below (top of figure 16), we can appraise an example of the previous algorithm on a single frame, the stars correspond to the voiced detected frequencies. The bottom picture shows the evolution of the maximum voiced frequency during time. The corresponding speech signal is shown in the middle.

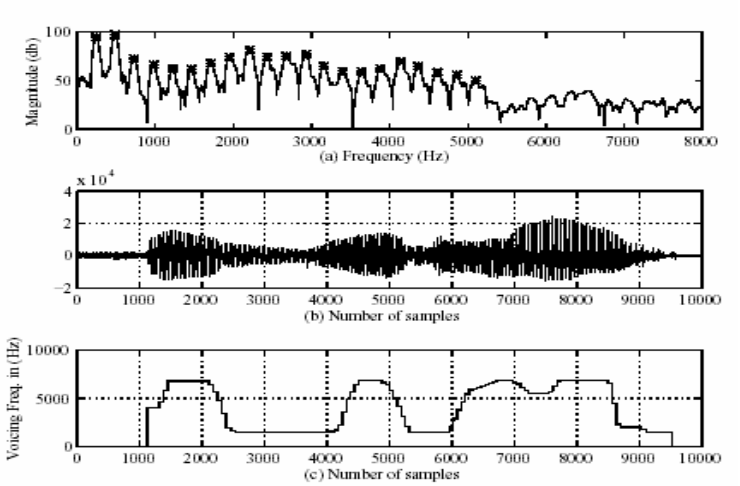


Figure 16 : determination of the voiced frequencies on a frame and evolution of the maximum voiced frequency F_M on a speech signal

The number of harmonics (L) is determined by $L_M = \frac{F_M}{f_0}$ for one analysis frame.

Note that the frequencies declared as voiced (called now f_i) by this criterion are found independently from the initial estimation of the fundamental frequency. This property is at the root of the following criterion that re-estimates the pitch.

IV.4.3.4 Reestimation of the fundamental frequency

Using the frequencies (f_i) declared as voiced by the previous step, we try to minimize the following function : $E(\hat{f}_0) = \sum_{i=1}^{L(i)} \left| f_i - i.\hat{f}_0 \right|^2$

with L(i) representing the number of voiced frequencies and f_0 the initial estimation of the pitch. The minimum is reached for the new estimation of the pitch.

On figure 17, we can see the original spectrum (continuous line) overlaid with the synthetic spectrum (dotted line) which is composed of the sum of the sinusoids whose frequencies are multiples of the pitch, for the first estimation of f_0 (top) and the second one (bottom). We can observe for the first one, the progressive shift of the synthesized harmonics with regard to the originals (actually, the product $i.f_0$ increases with the number of the harmonics i).

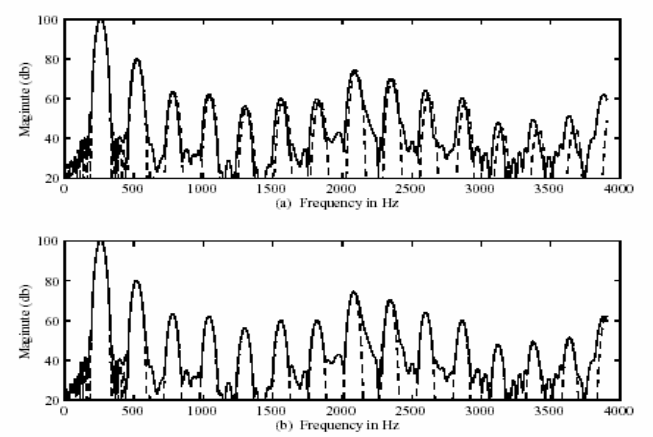


Figure 17 : Reestimation of the pitch

This is a recapitulative algorithm for the 4 previous points (figure 18).

Recapitulative algorithm (every 10 ms)

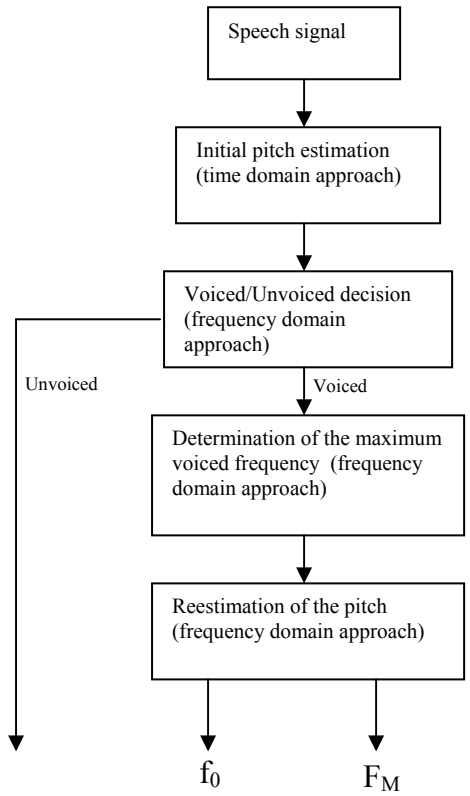


Figure 18 : Recapitulative algorithm for the pitch, voiced decision and max. voiced frequency

IV.4.3.5 Positions of the analysis time instants

Using the estimated fundamental frequencies, we are now able to determine the analysis time instants. These are synchronized according to T_0, the local fundamental period : $t_a^{i+1} = t_a^i + T_0^i$.

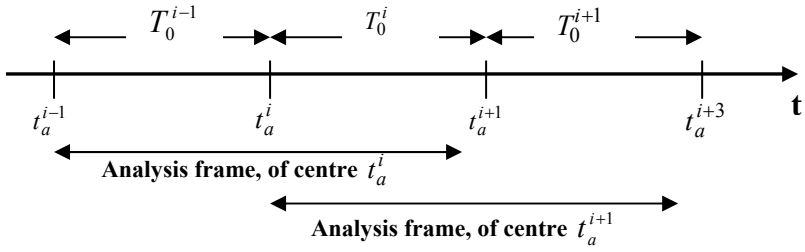


Figure 19 : Setting of the positions of the analysis time instants

We will consider the length of an analysis frame (at time i) equal to $2.T_0^i$ and we will suppose it centred on the analysis time instant t_a^i (see figure 19). Notice that the length of the frames vary between 20 ms (50Hz) and 2 ms (500Hz), depending on the local pitch of the considered frame.

It is clear that these analysis time instants do not correspond to the different instants t^i (see figure 11), which are all separated by 10 ms. In this case, the values of the previous evaluated parameters (at time t^i), so f_0, V/UV decision and F_M , must be associated to the analysis time instants t_a^i.
For the pitch and the maximum voiced frequency, we will use either an interpolation between the two pitch values estimated at the two times t^i around t_a^i or the nearest time t^i. On the other hand, for the V/UV decision we have to choose the parameters of the nearest time t^i.

The unvoiced frames, are incremented by a constant factor of 10 ms : $t_a^{i+1} = t_a^i + 10ms$.

IV.4.3.6 Computation of the amplitudes and phases of the harmonics for voiced frames

The prosodic modification (carried out during synthesis) involves a modification of amplitudes and phases of the harmonics, hence, it is necessary to estimate these correctly.

We suppose first that the pitch and the maximum voiced frequency F_M are constant on the analysis window and equal to those estimated at the considered analysis time instant.

$$T_0(t) = T_0(t_a^i)$$
$$F_M(t) = F_M(t_a^i)$$

where T_0 is the fundamental period and F_M, the maximum voiced frequency.
And for the phase of the k[th] harmonic, we have : $\phi_k(t) = \phi_k(t_a^i) + k.2\pi.f_0(t_a^i).(t - t_a^i)$

Taking these hypotheses into account, the "harmonics" signal can be written by :

$$\hat{\underline{S}}(t) = \sum_{k=-L_M}^{+L_M} \underline{A}_k(t_a^i).e^{j.2\pi.k.f_0(t_a^i).(t-t_a^i)}$$

with $L_M = \frac{F_M(t_a^i)}{f_0(t_a^i)}$, number of voiced frequencies and A_k, the amplitude of the harmonic k.

These complex amplitudes can be evaluated in the time domain by using a method based on a least-squares criterion. We weighted the window by a hamming window to maximize the correspondence between the computed coefficients and the centre of the analysis window.

We try to minimize the following function : $E = \sum_{t=t_a^i-T_0^i}^{t_a^i+T_0^i} w^2(t).[S(t)-\hat{S}(t)]^2$

Where we consider T_0 expressed as the number of samples corresponding to the local period.

In matrix form, we have for $\hat{S}(t)$: $\left[\hat{S}\right]=[B][X]$

with :

$$[B]=\begin{bmatrix} e^{j.(-L).w_0.(t_i-T_0)} & \dots & \dots & e^{j.(L).w_0.(t_i-T_0)} \\ \dots & & & \dots \\ \dots & & & \dots \\ e^{j.(-L).w_0.(t_i+T_0)} & \dots & \dots & e^{j.(L).w_0.(t_i+T_0)} \end{bmatrix}$$ a matrix $[(2T_0+1).(2L+1)]$

and

$$[X]=\begin{bmatrix} \underline{A}_{-L} \\ \underline{A}_{-L+1} \\ \dots \\ \underline{A}_{L-1} \\ \underline{A}_{L} \end{bmatrix}$$ a matrix $[(2L+1].(1)]$, avec $\underline{A}_L^* = \underline{A}_{-L}$

We also define the following diagonal matrix “window”:

$$[W]=\begin{bmatrix} w(-T_0) & 0 & 0 & 0 & 0 \\ 0 & w(-T_0+1) & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & w(T_0-1) & 0 \\ 0 & 0 & 0 & 0 & w(T_0) \end{bmatrix}$$ a diagonal matrix $[(2T_0+1).(2T_0+1)]$

where the w(n) elements correspond to the samples of the hamming window applied to the frame.
At last :

$$[S]=\begin{bmatrix} S(-T_0) \\ \dots \\ \dots \\ S(T_0) \end{bmatrix}$$ a matrix $[(2T_0+1).(1)]$ of the original samples of the frame.

The linear equations can be solved by inverting the following system :

$$\boxed{[R][X]=[b]}$$ with $[R]=[B]^T.[W]^T.[W].[B]$ and $[b]=[B]^T.[W]^T.[W].[S]$

We can show that the matrix [R] is a Toeplitz matrix, so algorithms such as Levinson or Schür can be implemented (reducing the complexity of p^3 to p^2). The amplitudes and phases computed will also be assumed constant on the whole analysis frame (centered on t_a^i).

IV.4.3.7 Computation of the complex spectral envelope

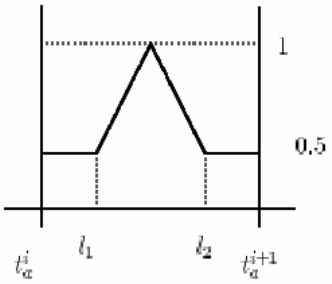
The literature on *HNM* gives us a free choice about the computation of this envelope. We are so developing this point in the practical part (point **V.2**).

IV.4.3.8 Parameters of the noise part

This step consists of the determination of the parameters of the noise part for voiced and unvoiced frames. The spectrum is modelled by an eighteenth order AR-filter (autoregressive). This filter is obtained by an LP analysis, which computes the autocorrelation function on 20 ms of signal (for unvoiced frames) or on the whole analysis frame (for voiced frames). A white noise will excite this filter and the dynamic characteristics (like stops) are considered by using a variance envelope

which modulates the excitation. The variance is estimated every 2 ms (so 10 variances for unvoiced frames and $T_0(ms)$ for voiced ones).

The noise comprising the second part of a voiced spectrum also needs to be modulated by a triangular-like time-domain energy envelope which accounts for energy bursts of the noise part, synchronized with the fundamental period (see figure 20). So, we have in the time-domain: $S_n(t) = e(t)\left[h(t) * b(t)\right]$ for voiced frames.



$$l_1 = 0.15\,(t_a^{i+1} - t_a^i)$$
$$l_2 = 0.85\,(t_a^{i+1} - t_a^i)$$

Figure 20 : triangular-like time-domain energy envelope for the calculation of l_1 and l_2

Observe that this triangular-like time-domain energy envelope is given here for the calculation of l_1 and l_2, which according to the author's thesis, considers the harmonic energy burst at the middle of two consecutive analysis time instants. This envelope will be adapted to the speaker during synthesis to synchronize the harmonic part with the noise one.

To separate the harmonic part from the noise one, a high-pass filter of F_M (maximum voiced frequency) cut frequency is used.

Recapitulative table of the noise parameters for each frame :

	Voiced Frame	**Unvoiced frame**
f_0	**1**	**0**
F_M	**1**	**0**
A_k	**2 * L_M**	**0**
Parameters of the AR filter	**10**	**10**
Variances σ^2	**T_0 (ms)**	**10**

Table 1 : Recapitulative table of the noise parameters for each frame

IV.4.3.9 Phase modification

The analysis time instants are computed synchronously with the local fundamental frequency, but are independent of the position of the glottal closure instants. This simplifies the analysis process because the estimation of these instants is avoided, but it increases the synthesis complexity because it introduces interframe incoherencies (phase mistakes) between voiced frames during concatenation (see figure 21). Thus phases discontinuities should be eliminated during the concatenation process. A method based on the centre of gravity of the frames is exploited [4].

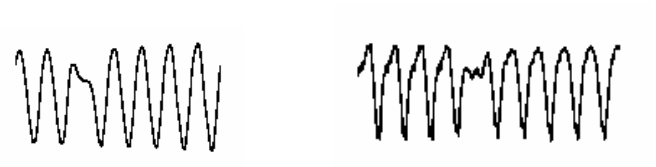


Figure 21 : Examples of concatenation process without any correction

For unvoiced frames, however, this discontinuity is not perceptible, so, no correction is carried out.
The phase correction consists of estimating the centre of gravity of each frame (we consider the frame of two local fundamental periods, centred on t_a^i), and translate it to the centre of the frame.
Let's first define the notion of the centre of gravity and see the link with the phase.

We define the centre of gravity of a function f(t) by : $\eta = \frac{m_1}{m_0}$

Where m_n is the nth moment of f(t) : $m_n = \int_{-\infty}^{+\infty} t^n . f(t).dt$

We can show [4] : $F^{(n)}(0) = (-j)^n . m_n$ where $F^{(n)}(0)$ is the nth derivative of the Fourier Transform of *f(t)* with respect to (the frequency) *w* and evaluated in *w* = 0.

So we have : $\eta = \frac{j.F^{(1)}(0)}{F(0)}$ (where F(0) corresponds to the mean of the signal)

If $F(w) = A(w).e^{j\phi(w)}$, we compute : $F^{(1)}(w) = [A^{(1)}(w) + j.\phi^{(1)}(w).A(w)].e^{j\phi(w)}$
Now, since f(t) is a real signal, , $\phi(0) = 0$ and, by $A(-w) = A(w)$ and $\phi(-w) = -\phi(w)$, we deduce $A^{(1)}(0) = 0$.
We obtain : $F^{(1)}(0) = j.\phi^{(1)}(0).A(0)$
And so :

$$\eta = \frac{j.jA(0).\phi^{(1)}(0)}{A(0)} = -\phi^{(1)}(0) \qquad (1)$$

This means that the centre of gravity of a function depends only on the first derivative of the phase at the origin.
Let's verify this property on the following example :

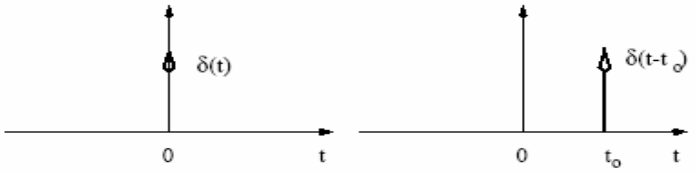


Figure 22 : Delta of Dirac shifted of t_0

We consider two Delta of Dirac, the first centred at the origin and the second in t_0. We naturally understand that the centres of gravity are respectively located in 0 and t_0 . The Fourier Transform of these two functions are 1 and $e^{-j.w.t_0}$. We apply this property and we find out 0 as the derivative of the phase of the first function at the origin (whose sign is inverted) and t_0 for the second one.

Let's apply now this principle to the speech signal. Let's consider the following voiced signal (a) and the linear prediction residual error (LP analysis) (b).

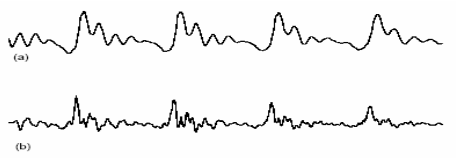


Figure 23 : Voiced signal and linear prediction residual error

We clearly see the energy localizations on the second diagram (b).
If the LP analysis led us to the AR filter h(t) and if r(t) denotes the linear prediction residual error, we have the relation : $s(t) = h(t) * r(t)$. It is demonstrated that $\eta_s = \eta_h + \eta_e$. But as the energy localisations of the original signal and the linear prediction residual error are quite similar (or are located at the same place), we can maintain that their centres of gravity coincide ($\eta_s = \eta_e$), hence, we can work with the linear prediction residual error. Considering previous developments, we have : $\phi_r^{(1)}(0) \approx \phi_s^{(1)}(0)$

Let's consider an energy burst of the linear prediction residual error (which represents the centre of gravity) and let's associate it the following rectangular signal, of which the centre of gravity is t_0. This signal will be useful to us for the rest of this reasoning.

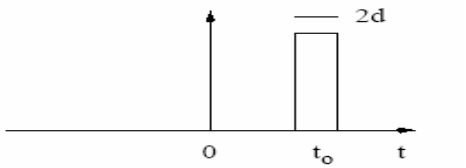


Figure 24 : Burst of the linear prediction residual error approximated by the rectangular signal

The centre of gravity t_0 of a frame is unknown at the beginning, it will be known by the mathematical development previously introduced. Then, the considered frame will have to be synchronized around its centre of gravity.

Let's call $\phi(w)$ the phase of the DFT computed at time 0 (associated to t_a^i) on two local periods and let's translate the signal with the result that its centre of gravity located in t_0 (unknown) corresponds to the time origin (in $t = 0$).
We obtain by this way : $\theta(w) = \phi(w) + w.t_0$ (2)
with $\theta(w)$ referring to the phase of the unknown centre of gravity.

As the centre of gravity is now located in $t_0 = 0$, we deduce by the first property (1) : $\theta^{(1)}(0) = 0$.
We deduce from (2) : $\theta^{(1)}(0) = \phi^{(1)}(0) + t_0 = 0$
From which : $t_0 = -\phi^{(1)}(0)$

We are looking now for the evaluation of the second member. By approximating the linear prediction residual error by an Dirac impulses train, we deduce from the DFT a linear phase. Actually the phase of an impulse train of Dirac is nil if it is estimated in $t = 0$; if we now evaluate it in t_0, its phase will be equal to $-w.t_0$, linear in w. But, we have previously considered this linear prediction residual error equivalent to the original one for the phase. We suppose so that the phase $\phi(w)$ is linear. We deduce the desired derivative (by considering a known point, e.g. w_0) : $\phi^{(1)}(0) = \frac{\phi(w_0) - \phi(0)}{w_0 - 0} = \frac{\phi(w_0)}{w_0}$.

We can rewrite (2) : $\theta(w) = \phi(w) + w.t_0$

$$\theta(w) = \phi(w) - w.\phi^{(1)}(0)$$

$$\theta(w) = \phi(w) - w.\frac{\phi(w_0)}{w_0}$$

$$\theta(k.w_0) = \phi(k.w_0) - k.\phi(w_0)$$

So, with a correction of each phase $\phi(k.w_0)$ (previously obtained during the analysis process) by a factor $-k.\phi(w_0)$, the centre of gravity (t_0) is translated at the origin (associated to t_a^i that corresponds to the centre of the analysis frame), independently from the initial position of the analysis window (of which the length is equal to 2 local fundamental periods and centred on t_a^i).

By applying this property on a frame, we obtain the following examples of results :

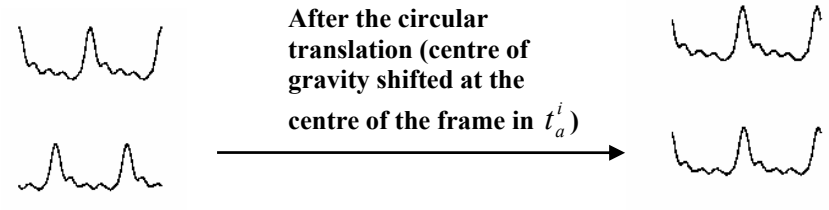


Figure 25: Circular translation of the samples of a window (analysis frame of centre t_a^i)

We proceed in the same way for each frame of an acoustic unit, what permits a rotation of these ones around their own centre of gravity. A global synchronization is then ensured and ideally no phase discontinuities will be sounded during the concatenation process (see synthesis). This synchronisation has been run without any estimation of the glottal closure instants.

Be careful that the rotation of the samples is applied on the synthesized signal (sum of harmonics part) and not on the original one. Note also that the translation doesn't modify the synthesized signal, so the quality remains similar.
Let's take again the previous two examples of figure 21 and let's show the correction effect :

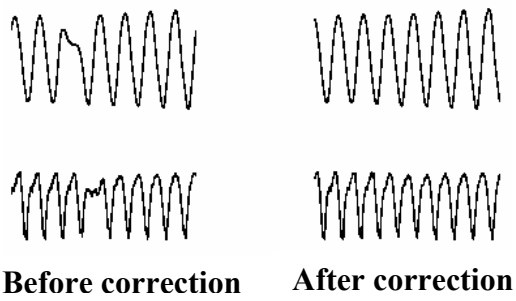


Figure 26 : Examples before and after the phase correction

This method based on the centre of gravity has another important advantage. Actually we observe for the voiced frames that the harmonic part has to be merged correctly with the noise part (see before). If this combining isn't respected, the timbre of the vowel will be perceived with a narrower "high frequencies" spectrum and two distinct sounds will be produced. In the other case, it's a larger "high frequencies" spectrum that will appear and that will give a more natural sound. It is so an important property for the vowel perception.

Thanks to the previous correction, the energy bursts are now located on the analysis time instants (and later on the synthesis time instants), so on known instants. Consequently, the boundaries of the triangular-like energy envelope can now be relative to the burst of the harmonic part.

This process appears so very interesting because beyond the initial purpose, i.e. the correction of the discontinuity between 2 consecutive acoustic units, it doesn't change the synthesized signal and permits us to easily do the combining between the noise and harmonic parts for voiced frames. Besides it's an off-line process that doesn't take resource during the synthesis.

IV.4.4 HNM Synthesis

We have seen in the description of this project that the concatenation of two acoustic units during waveform synthesis generally requires prosodic modifications (pitch and duration of phonemes).

From the HNM parameters issued from the analysis process, the synthesis is able to change some aspects such as the time scale and the pitch and to smooth the discontinuities between pitch and amplitudes at the concatenation point.
Hence, on the one hand the HNM synthesis brings desired prosodic modifications, and on the other hand, merges as naturally as possible two consecutive acoustic units (registered in different contexts).

The first step consists of loading for each used acoustic unit the corresponding parameters issued from the analysis process and saved in a database. This replaces the original speech data and consequently, no longer retain.

Next, we assume the pitch contour P(t) is continuous and known. Actually, this curve is provided by the previous module ("computation of the prosody"). Based on the initial prosodic characteristics of the two acoustic units to concatenate, we deduce the pitch modification factor $\alpha(t)$ and the time-scale modification factor, $\beta(t)$. These 3 parameters are provided by the "Prosody targets computation" unit.

IV.4.4.1 Estimation of the synthesis time instants and reestimation of the harmonic amplitudes

In this part pitch and time-scale modifications methods are developed.
The estimation of the synthesis time instants is necessary because they have to remain synchronous according to the target prosody. They will be so different from the analysis ones because a pitch modification is desired. Besides, if duration modifications are needed, duplication or removing of analysis time instants will be processed.
Secondly, if the pitch is modified, we have to reestimate the new amplitudes and phases of the new harmonics.
To have a good understanding, we shall separate at the beginning the time-scale modification and the pitch modification, and later combine them in an alone process.

A. Synthesis without any modification

No prosodic modification has to be applied, so it's the trivial case where synthesis time instants correspond exactly to the analysis ones.

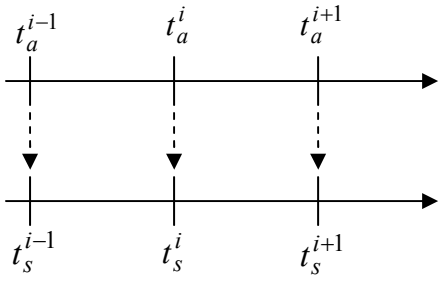


Figure 27 : Synthesis without any modification

B. Time-scale modification

By this modification, we would like to obtain a different articulation rate, but without any spectral modification (same harmonic components). The pitch contour will have to be stretched in the time domain, but remains identical.
Let's first define the time-scale warping function from the time-scale modification factor $\beta(t)$:

$$t' = D(t) = \int_0^t \beta(\tau).d\tau = \beta.t \quad \text{if } \beta = consant$$

We will see that this function is used to switch from the time-scale analysis times to the time-scale virtual times or from the time-scale virtual times to the time-scale synthesis times.
If β is lower than one, this means we carry out a compression of the time, so a speeding up of the articulation rate. On the other case, if β is higher than the unit, we proceed to an expansion of the time, so a slowing down of the articulation rate.

We associate a value of β to each analysis time instant t_a^i which we will assume constant for the whole interval $\left[t_a^i,...,t_a^{i+1}\right]$. The integral becomes so a linear function of the time :

$$t' = D(t) = D(t_a^i) + \beta_i (t - t_a^i) \qquad \text{for} \quad t_a^i \le t < t_a^{i+1}$$

So it's a matter of a recursive equation with the initial condition : $D(t_a^1) = 0$

Let's apply it to the 3 first instants with β_i = 1.5 :

$$D(t_a^1) = 0$$
$$D(t_a^2) = \beta_1 (t_a^2 - t_a^1)$$
$$D(t_a^3) = D(t_a^2) + \beta_2 (t_a^3 - t_a^2)$$
$$\dots$$

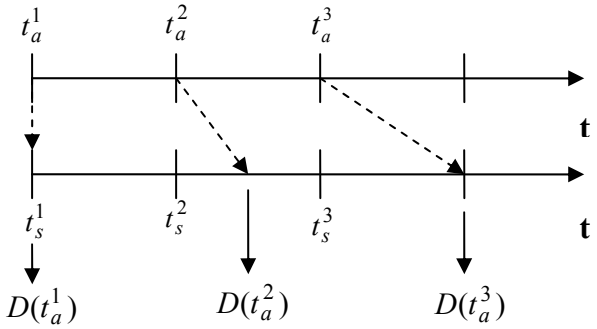


Figure 28 : Computation of $D(t_a^i)$, didactic figure explaining the function D(t)

We can see an expansion of the time. However these new times D(t_a^i) cannot correspond to the synthesis time instants because the fundamental frequency has to remain constant, so the space between the synthesis time instants and the analysis time instants must be equal. What varies here is actually the number of time instants. So we need to conserve the value of the pitch and in the same time modify the time-scale, that is to say : $T_0'(t') = T_0(D^{-1}(t')) = T_0(t)$, where t' corresponds to the synthesis time instant.

The synthesis time instants will so have to verify : $t_s^{i+1} = t_s^i + T_0'(t_s^i)$ to remain synchronous with the initial pitch. To achieve this task, we use virtual times instants between analysis and synthesis instants. We define the following relation :

$t_s^i = D(t_v^i)$ i.e. a ratio β between these two kinds of instants.

We also have between the analysis and virtual time instants a ratio of $\frac{1}{\beta}$, but that doesn't express itself by the mapping function D(t) because the number of virtual time instants (equal to the number of the synthesis ones) is different from the one of analysis time instants[3]. Once these virtual time instants have been determined, the synthesis time instants are found by :

$$t_s^{i+1} = t_s^i + \frac{1}{t_v^{i+1} - t_v^i} \int_{t_v^i}^{t_v^{i+1}} T_0(t'')dt'' \qquad \text{with } t_s^i = D(t_v^i) \text{ and } t_s^{i+1} = D(t_v^{i+1})$$

Therefore, the mean value of the pitch between the virtual time instants t_v^i and t_v^{i+1} lets us to synchronize the synthesis ones. The virtual time axis is denoted t'', its length is the same as the analysis one denoted t, but as previously announced, the number of virtual time instants is different of the analysis one. By considering $T_0(t'') = T_0(< t = t_a^i >) = cst$ for $t_v^i \le t'' < t_v^{i+1}$ - where the operator <.> means consider the time t equal to the nearest analysis time instant – we can simplify the last relation : $t_s^{i+1} = t_s^i + T_0(t_a^i)$

The goal is so achieved, the synchronism is respected in the synthesis process by keeping the initial values of pitches. The duration has been well modified since the number of synthesis time instants is different from the analysis ones.

Figure 29 expresses more clearly these changes of axes. To simplify, a ration β of 1.5 constant is chosen. The analysis time axe corresponds to the time t, the virtual one, to the time t'' and at last the one of synthesis, to the time t'.
We see the length of the analysis and virtual axes is identical but the defined time instants are different, the ratio 1/1.5 leads to a greater number of virtual time instants compared to the analysis one.
Next the function D(t) (ratio 1.5) maps the virtual time instants to the synthesis ones and a dilatation of the time axis is so observed.

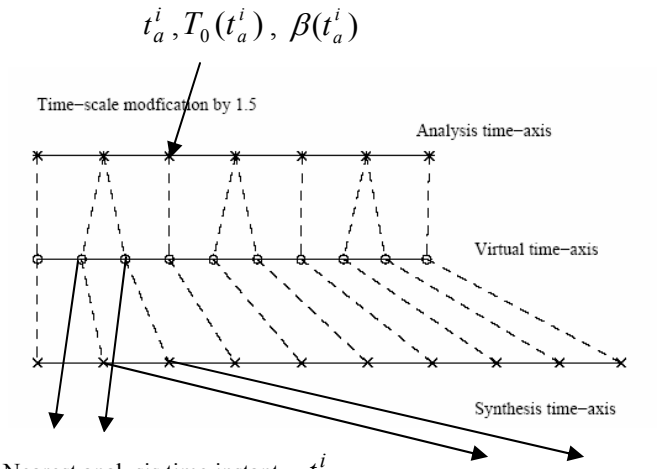


Figure 29 : Time-scale modification

[3] A geometric relation is so needed and will depend on the constancy of β in the time.

C. Pitch-scale modification

We wish here to modify the value of the pitch by a factor $\alpha(t)$ without changing the spectral envelope of the speech signal (the formants are so conserved). The time-scale will be conserved (length of the analysis axis equal to the virtual one, itself equal to the synthesis one) seeing as we do not want to modify the duration ; but the distance between the analysis time instants will be different from the synthesis ones because the fundamental period is modified.

We have : $t_a^{i+1} = t_a^i + T_0(t_a^i)$

We write so : $t_s^{i+1} = t_s^i + T_0'(t_s^i)$

As the virtual time instants are here identical to the synthesis ones, we can write :

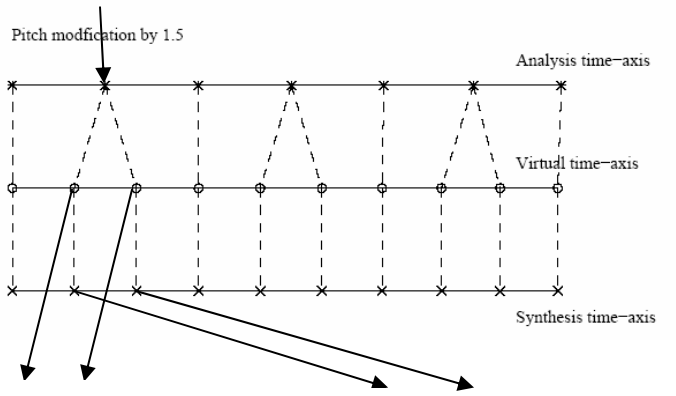
$$t_s^{i+1} = t_s^i + \frac{1}{t_s^{i+1} - t_s^i} \int_{t_s^i}^{t_s^{i+1}} T_0(t')dt'$$

Like the previous paragraph, we consider $T_0(t') = \frac{T_0(< t = t_a^i >)}{\alpha(t_a^i)} = cst$ for $t_s^i \le t' < t_s^{i+1}$

The integral amounts so to : $t_s^{i+1} = t_s^i + \frac{T_0(t_a^i)}{\alpha(t_a^i)}$ for $t_s^i \le t' < t_s^{i+1}$

Figure 30 shows this transformation, we take as a simplification $\alpha(t) = 1.5$ constant for all the frames (the pitch is so modified and multiplied by 1.5) :

t_a^i , $\frac{T_0(t_a^i)}{\alpha(t_a^i)}$



Nearest analysis time instant $= t_a^i$,
We consider so for these two virtual time instants : $\frac{T_0(t_a^i)}{\alpha(t_a^i)}$

We will have the following synchronization :

$t_s^{i+1} = t_s^i + \frac{T_0(t_a^i)}{\alpha(t_a^i)}$ and $t_s^{i+2} = t_s^{i+1} + \frac{T_0(t_a^i)}{\alpha(t_a^i)}$

Figure 30 : Pitch-scale modification

D. Time-scale and pitch-scale Modification

We introduce at the same time the time-scale modification factor $\beta(t)$ and the pich modification factor $\alpha(t)$; regarding to the two previous points, the following formula is developed :

$$t_s^{i+1} = t_s^i + \frac{1}{t_v^{i+1} - t_v^i} \int_{t_v^i}^{t_v^{i+1}} T_0(t'')dt''$$

The mean value of the pitch modified by the factor $\alpha(t)$ between the virtual time instants t_v^i and t_v^{i+1} enables us to synchronize the synthesis time instants. The virtual time axe t'' has the same length as the analysis one. As previously, if we consider $T_0(t'') = \frac{T_0(< t = t_a^i >)}{\alpha(t_a^i)} = cst$ for $t_v^i \le t'' < t_v^{i+1}$, we can simplify the latest relation : $t_s^{i+1} = t_s^i + \frac{T_0(t_a^i)}{\alpha(t_a^i)}$

Let's take the first example again which can be applied with $\alpha(t) = 1$ and $\beta(t)$=1.5 :

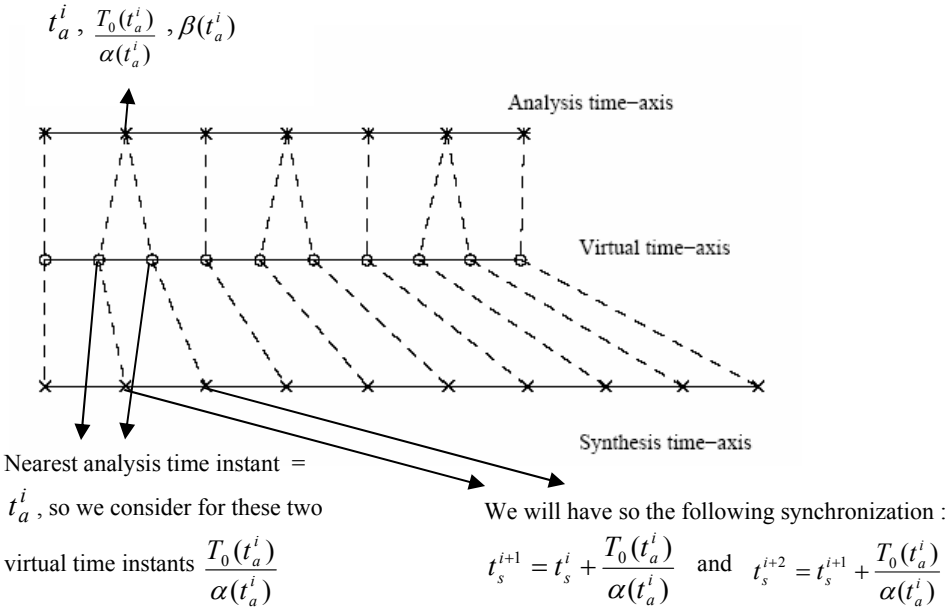


Figure 31 : Time-scaled and pitch scale modifications

Remember that the HNM parameters (voiced decision, pitch, amplitudes, phases, prosodic parameters,…) of the virtual time instants (mapping to the synthesis ones) are equal to the nearest analysis time instants.

Figure 32 presents the spectrograms (with narrow band) of an original signal and of the signal of which the time-scale and pitch have been modified. We clearly see the horizontal moving of the formants (due to the expansion of the time) and the vertical moving of the harmonics (due to the pitch modification).

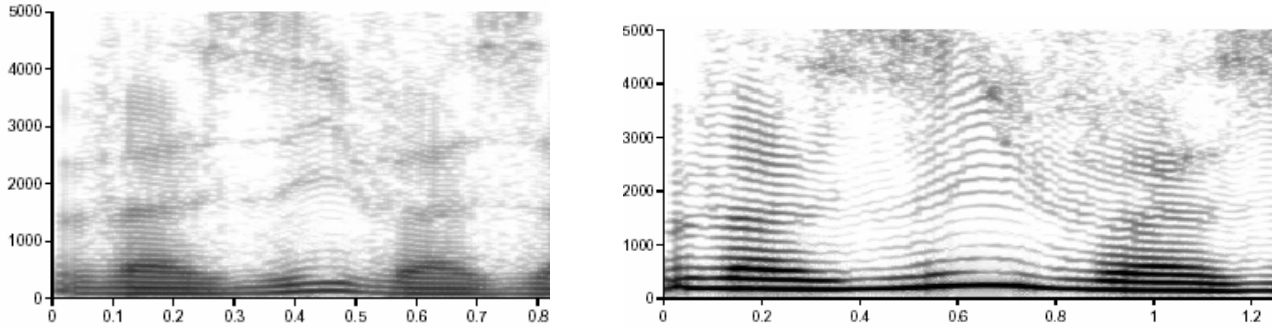


Figure 32: Spectrograms of an original signal and the corresponding time-dilated and pitch-increased signal

E. Reestimation of the amplitudes and phases of the new harmonics

If the fundamental frequency has been modified, we need to reestimate the amplitudes and phases of the new harmonics. Actually, the new harmonics are now : $2.\pi.k.\alpha(t_a^i).f_0(t_a^i)$ for the considered frame. So, if $\alpha(t_a^i) < 1$, $\hat{s}(t)$ will be composed of more harmonics, while if $\alpha(t_a^i) > 1$, $\hat{s}(t)$ will be composed of fewer ones, since the maximum voiced frequency is the same.
We don't have the original samples, but only the HNM parameters, therefore we cannot use the original Fourier Transform. We will use the complex envelope computed during the analysis process and resample this one (in amplitudes and phases) at the new harmonics.

To preserve the original energy of the signal, we can apply a scale factor on the resulting spectrum.
The figure 33 shows the original spectrum (above) and the resulting spectrum with a pitch modification of $\beta = 1.5$, we remark that the harmonics are more spaced.

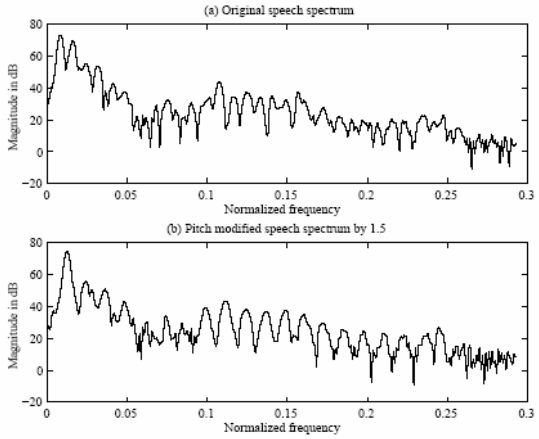


Figure 33 : Original and modified spectrum by $\beta = 1.5$

IV.4.4.2 Smoothing of the discontinuities

Once the synthesis time instants have been computed on the basis of the modifications required by the prosodic targets, we have to smooth the discontinuities of amplitudes and pitches for voiced frames before the synthesis of the waveform. Normally, the pitch discontinuity has to be removed by the pitch modification at the previous step. This smoothing is processed by a linear interpolation. Perceptually, the discontinuities of the noise parameters (AR filter coefficients and variances) have no influence, so they are not subjected to the smoothing process. The phase modification based on the centre of gravity shifting eliminated the problem of phase discontinuity, so only the pitch (and so the harmonics) and the amplitudes will be considered.

Let's remark at this stage that the concatenation point means a "virtual" point between the first synthesis time instant of the right acoustic unit and the last synthesis time instant of the left one.

At the beginning, the differences between the pitches and the amplitudes of the harmonics of the two acoustic units are evaluated at the concatenation point (so between the first synthesis time instant of the right acoustic unit and the last synthesis time instant of the left one). We call U_l and U_r the two acoustic units, on the left and on the right of the concatenation point, respectively. We also call w_0^i and A_k^i the pitch and the amplitudes of the k harmonics (of which the fundamental) of the last frame of U_l and w_0^{i+1} and A_k^{i+1} the ones of the first frame of U_r. The smoothing is

performing from the concatenation point and propagated on each side of this point with a different weight.
The smoothing of the pitch will be performed according to :

$$\Delta w_0 = \frac{w_0^{i+1} - w_0^i}{2}$$

$$\tilde{w}_0^i = w_0^i + \Delta w_0 . \frac{i}{L} \qquad \text{with} \quad i = L, L\text{-}1,\ldots,1$$

$$\tilde{w}_0^r = w_0^r - \Delta w_0 . \frac{n}{L} \qquad \text{with} \quad n = R, R\text{-}1,\ldots,1$$

with $r = i + R - n + 1$

The pitches at the concatenation point (w_0^i and w_0^{i+1}) are already modified by the pitch modification factor α during the prosodic modifications process, so they are quite close.
We can observe the added difference to the pitch value ($\Delta w_0 . \frac{i}{L}$ and $\Delta w_0 . \frac{n}{L}$) is decreasing as we reach the extremities.
Note that the harmonic frequency values are automatically modified because it corresponds to a multiplicative factor of the pitch during the synthesis process.

The smoothing of the amplitudes (fundamental (k = 1) and the k harmonics) are obtained in the same way :

$$\Delta A_k = \frac{A_k^{i+1} - A_k^i}{2}$$

$$\tilde{A}_k^i = A_k^i + \Delta A_k . \frac{i}{L} \qquad \text{with} \quad i = L, L\text{-}1,\ldots,1$$

$$\tilde{A}_k^r = A_k^r - \Delta A_k . \frac{n}{L} \qquad \text{with} \quad n = R, R\text{-}1,\ldots,1$$

with $r = i + R - n + 1$

A voiced frame followed by an unvoiced one (and vice versa) becomes then a problem. If a unvoiced frame i is followed by a voiced frame i+1, we apply the next operation :

$$\begin{cases} \text{Frame i} & \text{NV} \\ \text{Frame i+1} & \text{V} \end{cases} \longrightarrow \begin{cases} A_k^i = 0 \\ f_0^i = f_0^{i+1} \end{cases}$$

If now the voiced frame i is followed by an unvoiced frame i+1 :

$$\begin{cases} \text{Frame i} & \text{V} \\ \text{Frame i+1} & \text{NV} \end{cases} \longrightarrow \begin{cases} A_k^{i+1} = 0 \\ f_0^{i+1} = f_0^i \end{cases}$$

Let's indicate that this linear interpolation makes the formant discontinuities (on the left and on the right of the concatenation point) less perceptible. However, if these discontinuities are very significant, then the problem is not solved. If the synthesis is using a corpus database (unit selection-based synthesis), another acoustic unit will have to be chosen ; but if the diphones synthesis is performed – where only one representative per diphone is registered in the database – a degradation of the speech quality will be unavoidable.

IV.4.4.3 Speech waveform generation

We are now ready to synthesize our speech signal. Actually, we have the synthesis time instants computed on the basis of the analysis ones and the necessary prosodic modifications, associated to new pitches and new amplitudes and phases computed and smoothed.
This generation of waveform is based on the formula introduced in the description of the *Harmonics Plus Noise Model*, so for voiced frames:

$$\hat{S}(t) = S_h(t) + S_n(t) \quad \text{with} \quad \underset{-h}{S}(t) = \sum_{-L(t)}^{L(t)} \underset{-k}{A}(t).e^{jkw_0(t).t} \quad \text{and} \quad S_n(t) = e(t).[h(t) * b(t)]$$

and for unvoiced frames:

$$S(t) = [h(t) * b(t)]$$

The harmonic signal generation is the most computationally expensive operation. Some implementation strategies can reduce the complexity and are developed in the point A, [5]. Then we detail the noise synthesis (point B) and finally we will compute the synthetic discrete waveform (point C) which combines the noise part and/or the harmonic part.

A. Generation of the harmonic part

This point concerns only the voiced frame.
We synthesize the harmonic part according to the following equation :

$$S_h(t) = \sum_{k=1}^{L_M(t_S^i)} A_k(t_S^i).\cos(k.w_0.t + \phi_k(t_S^i))$$

with t varying from 0 to N (where N corresponds to the number of samples of a voiced frame) and $L_M(t_S^i) = \frac{F_M(t_S^i)}{f_0(t_S^i)}$ is the number of voiced frequencies (harmonics) at time (t_S^i)
Four methods lead to the synthesis of k harmonics :

- The first one consists of directly implementing the sinusoids in the temporal domain Straight-Forward synthesis, SF). The direct way is however the more computational expensive process.

- The second one is based on the Inverse Fourier Transform (IFFT). We simply compute the IFFT from the k necessary harmonics.

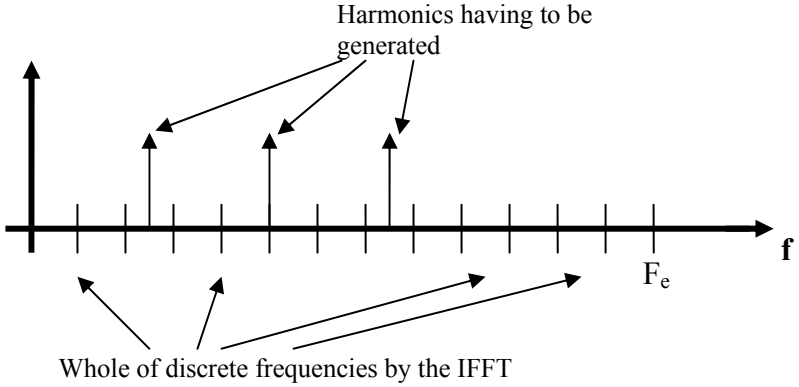


Figure 34 : Harmonic part generation by IFFT

Nevertheless to assure a fast operation, it is necessary to have a length of IFFT (frequencies where the FFT or IFFT is computed) equal to a power of 2. In associating the harmonics (having to be generated) to the available discrete frequencies (see figure34). We see that there are 2^x in number and are uniformly distributed between 0 and F_e Hz (where F_e is the

sampling frequency and x the number of bits of coding). This introduces so an error in the harmonic model. Evidently the greater the length of the IFFT, the smaller the error, but the synthesis will be slower. A minimum length of 1024 is required to reach an acceptable quality (with F_e of 8000 Hz).

- The third method runs recursive trigonometric relations (Recurrence Relations for Cosine functions, RR). Each iteration actually computes a new harmonic :

$$\cos(\theta + \delta_k) = \cos(\theta) - [\alpha_k . \cos(\theta) + \beta_k . \sin(\theta)]$$
$$\sin(\theta + \delta_k) = \sin(\theta) - [\alpha_k . \sin(\theta) - \beta_k . \cos(\theta)]$$

where the coefficients α_k and β_k are pre-computed by the following relations :

$$\alpha_k = 2.\sin^2(\frac{\delta_k}{2}) \quad \text{with } \delta_k = k.w_0 \text{, for each harmonic.}$$
$$\beta_k = \sin(\delta_k)$$

The initialization computes the fundamental sinusoid in the temporal domain (by method 1), with the associated phase ($\theta^{(1)} = w_0.t + \phi_1$).Then the harmonics are calculated by recurrence. Their phases are taken into account thanks to the initialization.

- At last, the fourth method consists in converting the phase into delay and down-sampling the initial sinusoid (Delayed Multi-Resampled Cosine functions, DMRC). The delay (at the origin) of the k[th] harmonic is defined by : $t_k = \frac{-\phi(k.w_0)}{k.w_0}$.

By this way the function that must be generated : $\tilde{s}(t) = \sum_{k=1}^{L_M(t)} A_k(t).\cos(k.w_0.t + \phi_k(t))$ where Φ_k is the delay at the origin, becomes : $\tilde{s}(t) = \sum_{k=1}^{L_M(t)} A_k(t).\cos[k.w_0.((t - t_k) \bmod T_0)]$ for t = [0,…T_0-1] (because periodic). "modT_0" allows the computation on only one period (T_0, the fundamental period). So this method consists of calculating (by the first method) the function $U(t) = \cos(w_0 t)$ on one period T_0 and then for each harmonic to translate the samples by a factor t_k and down-sample it by a factor k so as to satisfy the equation $U(k(t - t_k))$.

The down-sampling is sufficient since the synthesis is discrete and the step of quantisation (sampling period) is constant for each harmonic. For example, on a fundamental period containing two periods of the first harmonic, as the step is constant we will need two times fewer samples for the first harmonic than for the fundamental.

The comparison of the four methods can be done by the signal to noise ratio (SNR) defined by :

$$SNR = 10 \log_{10} \frac{\sigma^2_{s(t)}}{\sigma^2_{\tilde{s}(t) - s(t)}}$$

A study realised in the same paper [5], has collected around 500 frames (divided equally between genders), synthesized using the four methods to give the resulting table of performances for SNR and computation time. The IFFT method was performed with different length (512, 1024, 2048, 4096, 8192) and the computation time is relative to the SF method. See table 2.

As previously announced the computation times increases with the length of the IFFT until exceeding the SF method ; the signal to noise ratio also increases, but cannot exceed the one of the SF method. The RR method is highly effective on the two criteria, but the best implementation is the last, because for the same signal to noise ratio as the RR method, it allows to reduce the computation time by a factor of 3.

	SNR (dB)	Computation time
SF	31.21	1
IFFT (512)	5.04	0.206
IFFT (1024)	10.66	0.238
IFFT (2048)	16.65	0.444
IFFT (4096)	21.28	0.984
IFFT (8192)	27.52	2.507
RR	29.89	0.158
DMRC	30.62	0.047

Table 2 : Performances of the harmonic generation algorithms

B. Noise part generation

Recall that the noise part plays a part in voiced and unvoiced frames. We model this by a Gaussian white noise modulated by the variances envelope and then filtered by the AR filter. In the case of voiced frames the harmonic part and the noise part are separated by a high-pass filter with a time-varying cutoff frequency ($F_M(t_S^i)$). A shift of the noise samples has also to be considered for taking into account the phase modification (based on the centre of gravity) on the voiced frame. A triangular-like time-domain energy envelope is then introduced to modulate the signal and synchronize these two parts. As previously cited, if this last step is omitted, the harmonic and noise parts will perceptually appear distinct because they won't have been correctly fused.

We generate the white noise by a sum of sinusoids with random phases at the origin. For voiced frames the frequency of these harmonics will be greater than the maximum voiced frequency F_M. By this way we avoid the use of the high-pass filter.
In the case of unvoiced frames the frequencies of the sinusoids (whose sum approximates the white noise) cover the full spectrum.

C. Final synthesis

The artificial speech signal synthesis constitutes the outcome of all the analysis and synthesis processes. The voiced and unvoiced frames are synthesized by an overlap and add process (OLA) to give the resulting speech. The general graph used in practice is shown in "programming of the HNM" (point **V.2.2.1**).

IV.4.5 Results

Some MOS (Mean Opinion Score) tests have been detailed in the literature from the diphone-based HNM and TD-PSOLA synthesis [2]. The recordings were done by 6 different professional speakers and were on the one hand English sentences and on the other hand nonsense words. Both synthesizers used the same prosodic corrections.
41 adults forms the listeners, they don't have any knowledge in TTS synthesizer and haven't any hearing problem. These MOS tests consist in indicating on a scale (from 1 to 5) the listened speech quality, the highest value corresponds to the excellent quality (see table 3).

NOTE	DEGRADATION
5.0	**Imperceptible**
4.0	**Perceptible but not irritating**
3.0	**Lightly irritating**
2.0	**Irritating**
1.0	**Very irritating**

Table 3 : MOS scale

On the basis on this scale the listeners have classified the sentences and nonsense words according to their intelligibility, naturalness and pleasantness. Everyone has emitted 936 ratings, totalling 38376 observations.
It ensues from these observations (for both syntheses) that the intelligibility rating is higher than naturalness and pleasantness ratings, which are more or less equivalent. It emerges from this study that HNM synthesis is better than TD-PSOLA, as the following table illustrates.

	English sentences	**Nonsense words**	**Mean**
HNM	**3.05**	**2.95**	**3.00**
TD-PSOLA	**2.84**	**2.66**	**2.75**

Table 4 : Comparison between HNM et TD-PSOLA on the MOS scale

The two first columns correspond to means on all the ratings of intelligibility, naturalness and pleasantness. The third represents the mean of the English sentences and nonsense words. We thus see that the diphone-based HNM synthesis was by average rated about 0.25 points higher than TD-PSOLA synthesis on the MOS scale.

A second test has been realized by using the unit selection-based HNM synthesis. Four sentences especially appropriated to the prosodic modifications of *Festival* have been chosen and presented to 44 listeners. A quotation of 3.91 on the MOS scale has been resulted. The speech quality has been so judged to be close to the natural speech quality without any smoothing problem.

These tests demonstrate the high quality of speech that HNM can provide.

IV.5 Recapitulative of the advantages and drawbacks of the LPC, TD-PSOLA, MBROLA and HNM syntheses

This point recapitulates the different advantages and drawbacks of Linear Prediction Coding (LPC), *Time Domain Pitch Synchronous Overlap and Add* (TD-PSOLA), *Multi-Band Re-synthesis Pitch Synchronous OverLap and Add* (MBR-PSOLA) and *Harmonic Plus Noise Model* (HNM), in their standard forms (as described in this report).

	Advantages	**Drawbacks**
LPC	• Simple and easily run. • Very low bit rates can be reached in speech coding. • Possible Interpolations (formants, pitch).	• Cannot model the antiformants and mixed sounds. • Decouples the modelling of the excitation and the vocal tract. • Average quality of speech.
TD-PSOLA	• Low cost in computation load. • Prosodic modifications easily brought • High speech quality.	• Unparametrical (acts directly in temporal domain), hence phase, pitch and formants discontinuities during concatenation process. No possibility of smoothing.
MBR-PSOLA	• Combines the low computation load and the prosodic modifications of TD-PSOLA with a parametric which allows to re-synthesize the segments in order to avoid any discontinuities at the concatenation point. • High speech quality.	• Constant phases at the origin during the re-synthesis of the segments gives an unnatural speech.
HNM	• Decomposition into a harmonic and a noise parts give more natural sounds. • Coupling of the modelling of the excitation and of the vocal tract. • Parametrical structure allows easily bringing prosodic modifications and smoothing of the discontinuities during concatenation.	• High computation load related to its complexity.

Table 5 : Recapitulative of the LPC, TD-PSOLA, MBR-PSOLA and HNM syntheses

V. Integration into Festival and programming of the HNM

This fifth part is devoted to the practical realization of the *Harmonic Plus Noise Model*. As this waveform uses in this project the natural language processing modules of *Festival*, a brief description of its architecture is first presented. Then the implementation of the HNM and the strategic choices are developed. This part, which constitutes an essential point of my work shows the practical approach of the model with the encountered difficulties and their resolutions. It shows moreover the modifications brought on the theoretical approach. The obtained qualitative results of this implementation are presented in the third part. Finally we shall explain how we can integrate our HNM module into the *Festival* speech engine, since this part was not realized, through lack of time.

V.1 Architecture of Festival speech engine

Festival carries every functionalities of a speech synthesis engine [21,22]: from the natural language processing modules to waveform generation. The natural language processing converts the input text into parameters (*features*) necessary for the waveform synthesis. These parameters are stored and linked among themselves by an *architecture*, which mustn't be tied to any particular language in order to provide a maximum of generality. It must also be efficient and rapid so as to allow a real-time waveform synthesis. By default, the waveform generation is realized by RELP, so LPC where the simple excitation is replaced by the residual signal, which considerably increases its quality.

Festival brings a different architecture completing the two previous ones : the first, known under name of "string processing" consists of a linear architecture ("string") in which are represented all the linguistic informations. At the beginning the "string" contains the text embellished with a number of symbols called *items* (like the speed rate, the phonemes,…). The main problem of these structures is that they rapidly become complex and quite difficult to manipulate. The second structure separates the previous linear information by different *streams* to result in the *Multi-level data structure* (MLDS). So for example words, phonemes and syllables *streams* will be separated. These *streams* aren't of course independent and are aligned by the edges of *items* (relations between streams), in order to retrieval the phoneme of a word for example. The greatest drawback of the *architecture* is that it conserves linear *streams* of informations, whose number can become considerable, which increases the complexity of the *relations*. Sometimes a particular attribute will be absent in a relation : for example a pause will be represented in the phoneme *stream* but not in the syllable and word *streams* ; so a "hole" will be necessary in order to ensure the synchronization between the different *streams*. With the considerable number of *streams*, the number of "holes" can also become enormous what can lead to difficulties in the algorithms that use these *streams*. Figure 35 gives an example of the MLDS representation (whose only a part is represented).

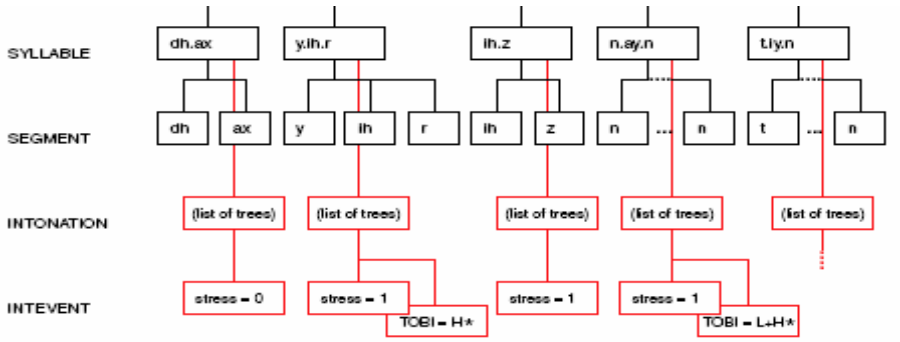


Figure 35 : Example of MLDS architecture

The current *architecture* of *Festival* distinguishes itself from the MLDS one by allowing all possible structures (trees, linked lists,…) which are so not constrained to remain linear. Additionally the items (words, syllables, phonemes,…) can now belong to many different structures. So this architecture offers greater flexibility. Each *item* has a number of associated *features* which describe its local properties and the different links into its different structures. Any number of features are possible (even equal to zero for syllables for example) and can be made up of functions as values. These allow a reduction of storage of the redundant information in the data structures. For example, instead of storing the start time and the duration of each phoneme, a function can return the end time of the previous phoneme. Another function can then take these values as an argument for computing the phoneme durations.
The *items* of the same class are bound together by *relations*. These permit organizing the *items* into structures. For example we shall use doubled linked list for the words and trees for the syntactic relations. The "word" nodes will have a previous and a next node, while the "syntactic" nodes will have a father node and a child node. The following example illustrates these things (figure 36) and shows the "word" items belonging to two different structures : word and syntactic structures.

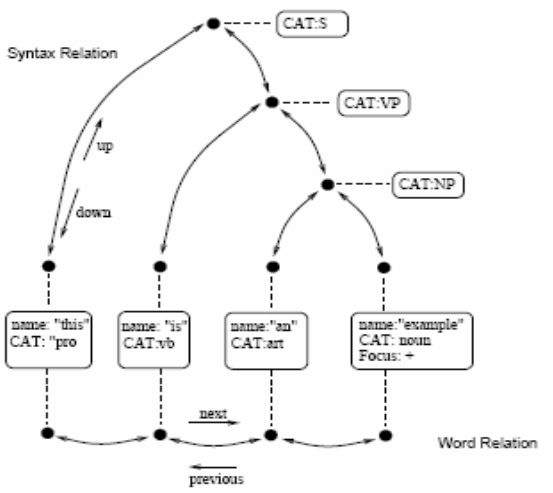


Figure 36 : Architecture of Festival

V.2. Programming of the HNM

I realized the implementation of the *Harmonic Plus Noise Model* in the C/C++ language, but mainly in C. The C++ was only useful for in/out manipulations (files) since the algorithms are based on computation and not on data processing, so any important class emerges and the low level C langage is then preferred. It offers, moreover, greater performance – because it is closer to the machine language – which has to be taken into account, since we need real-time synthesis of the speech.

The voice diphones database I have used is the British voice 'RAB'. It is freely provided with *Festival* (*http://festvox.org/index.html*).

This part describes the implementation of the HNM analysis and synthesis process. It recovers the different points discussed in theory making necessary modifications according to the constraints fixed by Festival and the different choices tied to the implementation. In addition it brings some corrections and modifications compared with the theoretical approach.

V.2.1 HNM analysis

V.2.1.1 Evaluation of the pitch

Festival provides in its database, as well as the diphones database, the associated pitch-marks, that is to say the instants corresponding to closure of the vocal cords and so separated by a local fundamental period. The file is presented like this :

```
#
0.010826 125 1.000000
0.021653 125 1.000000
0.032479 125 1.000000
0.043306 125 1.000000
…
```

The first column corresponds to the pitch-marks. We easily compute for the two first instants :

$$\text{Instant 1 : } 0.010826 \text{ : pitch = } \frac{1}{0.010826} = 92.3702 \text{ Hz}$$

$$\text{Instant 2 : } 0.021653 \text{ : pitch = } \frac{1}{0.021653 - 0.010826} = 92.3617 \text{ Hz}$$

These pitchmarks have been generated by a laryngograph and are so of high accuracy. The unvoiced frames also have an associated pitch (of arbitrary value). It will be up to the following algorithm to correctly distinguish voiced frames from unvoiced ones.

V.2.1.2 Voiced/Unvoiced decision

The criterion set out in the theory (point **IV.4.3.2**) is applied in the discrete time.

$$E = \frac{\int_{0.7\hat{f}_0}^{4.3\hat{f}_0} \left(|S(f)| - \left|\hat{S}(f)\right| \right)^2 df}{\int_{0.7\hat{f}_0}^{4.3\hat{f}_0} \left(|S(f)|\right)^2 df}$$

We only need to convert 0.7 w_0 ; 4.3 w_0 and also the four first harmonics w_0, 2 w_0, 3 w_0 et 4w_0 into the correct index of the corresponding spectrum ray. $\hat{S}(t)$ that correspond to the sum of harmonics, is increased by amplitude values from one hand to other of each one in order to more decreasing the numerator. Besides, the threshold has been increased to -10 dB for decreasing the strong criterion.

Note : we are working with a sampling frequency of 16kHz (used for speech synthesis), so with N_{FFT}=4096, we obtain a resolution of f_e/N_{TFD} = 3.9 Hz and the accuracy will be sufficient even if $k.f_e/N_{FFT}$ is not a multiple of $k.f_0$.

Three smoothing median filters (a first of three points during the computation of the decisions, a second of three points and a third of five points after the decision process) are applied in order to provide uniform decision groups. It is true that “gaps” in the succession of decisions (a voiced decision in a unvoiced group or the contrary) can lead to decrease in the resulting speech quality, hence an additional precaution has so been made.

The filtering consists in moving a 3 cases window on the binary vector (by step of one) and in putting in the central cell the mean value (or here the most represented) of the three considered.

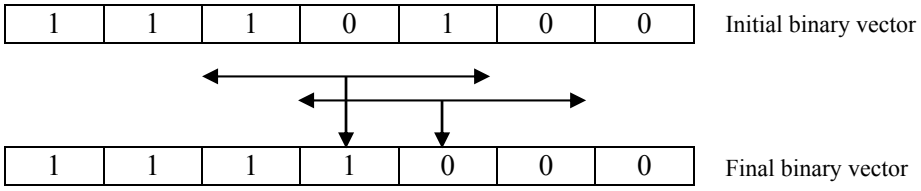


Figure 37 : Three points median smoothing filter

At last it appears that the criterion is more effective by choosing for the boundaries of the working frame one local fundamental period before the computation instant and three local fundamental periods after the computation instant, respectively.

```
lowboundary=TIME[i]-T0[i];
if(lowboundary<0){lowboundary=0;}
upboundary=TIME[i]+3*T0[i];
if(upboundary>=numbersamples-1){upboundary=numbersamples-1;}
```

But by this way the ends of the voiced groups are missed. We complete thus manually these ends of groups by bringing a number of additional voiced decisions (based on the number of unvoiced decisions of the following group). So we put to "1" some unvoiced decisions at the end of voiced groups.

V.2.1.3 Estimation of the maximum voiced frequency

The criterion (of point **IV.4.3.3**) is applied to the letter. Only the conditions were relaxed :

```
if( (AM/AMC)>1.2 &&(maximumampldB-maximumpeaksampldB)>0.0  &&  (fabs((Fc-l*f0)/(l*f0))<0.2))
```

The number of resulting harmonics varies between 30 and 70 in accordance to the speech spectrum, which is expressed by a maximum voiced frequency of about 3000 – 7000 Hz. This maximum voiced frequency is generally equal to 5000 Hz.
So as to reduce the discontinuities of maximum voiced frequency between successive frames, a 7-points median smoothing filter is applied two times.

Let's finally notice that a mistake of 30%, indeed 50%, of the estimated number of harmonics has not a great influence on the resulting speech quality (providing that this number remains sufficiently high (>20)).

V.2.1.4 Reestimation of the fundamental frequency

The pitch-marks provided by the database are enough accuracy (manual verification) and this part is so removed from the analysis process.

V.2.1.5 Position of the analysis time instants

The position of the analysis time instants is obtained in a recursive way. Recall that the instants that we have at this time are separated by a local fundamental period issued from the Festival database (pitchmarks). We also have for these times a voiced decision V/NV. So the analysis time instants are calculated by :

→ Initialization : 1st analysis time instant equal to the 1st instant from the database.

→ Next analysis time instant : $t_a^{i+1} = t_a^i + T_0^i$ (voiced)

$$t_a^{i+1} = t_a^i + 10ms \text{ (unvoiced)}$$

which is associated to the nearest database instant (k) :

```
do
```

```
{
        comp1=abs(antimeinst[i+1].time-(TIME[k]));
        k++;
        comp2=abs(antimeinst[i+1].time-(TIME[k]));

}while(comp2<comp1);
k--;
```

→ End when the last database instant has been associated.

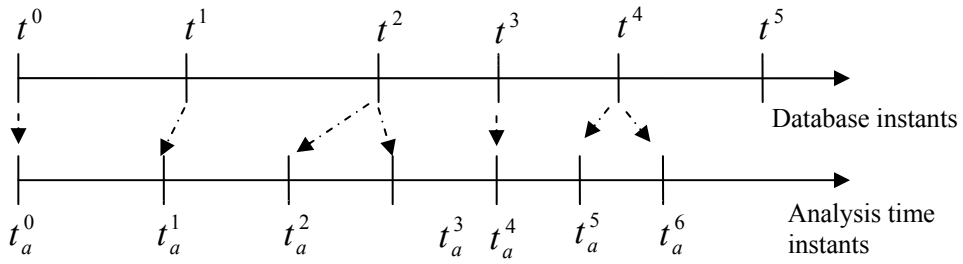


Figure 38 : Association of the database instants to the analysis ones

The associated parameters at each analysis time instant will be considered as constant on the whole analysis frame centred on the analysis instant in question (HNM1 model).

V.2.1.6 Estimation of the amplitudes and phases of the harmonic part

The matrix [R] introduced in the theory can be computed by the following formula[4] :

$$r_{i,k} = \sum_{t=t_a^i-T_0}^{t_a^i+T_0} w^2(t).e^{j.(k-L-1).w_0.t-j(i-L-1)w_0.t} = \sum_{t=t_a^i-T_0}^{t_a^i+T_0} w^2(t).e^{j.(k-i).w_0 t}$$

and [b] with :

$$b_k = \sum_{t=t_a^i-T_0}^{t_a^i+T_0} w^2(t).S(t).e^{-j(k-L-1)w_0.t}$$

The matrix [R] is a Toeplitz matrix because :

$$r_{i+p,k+p} = \sum_{t=t_a^i-T_0}^{t_a^i+T_0} w^2(t).e^{j.((k+p)-(i+p)).w_0 t} = \sum_{t=t_a^i-T_0}^{t_a^i+T_0} w^2(t).e^{j.(k-i).w_0 t} = r_{i,k}$$

However this matrix and the independent term are complex. The C language doesn't deal with complex numbers, hence, we need so work again on our matrix in order to separate it into a real part and an imaginative part.

$$[R][x] = [b]$$

$$[R_r + j.R_i][x_r + j.x_i] = [b_r + j.b_i]$$

$$[R_r x_r - R_i x_i] + j.[R_i x_r + R_r x_i] = [b_r + j.b_i]$$

hence :

$$\mathrm{Re} -> R_r x_r - R_i x_i = b_r$$

$$\mathrm{Im} -> R_i x_r + R_r x_i = b_i$$

We define so the following system : $\begin{pmatrix} R_r & -R_i \\ R_i & R_r \end{pmatrix}\begin{pmatrix} x_r \\ x_i \end{pmatrix} = \begin{pmatrix} b_r \\ b_i \end{pmatrix}$

The reorganized matrix [R] is no longer of Toeplitz. So we must resolve the linear system by a Gauss-Jordan matrix inversion (for example). However it exists other more efficient solutions.

[4] The formula given in the literature has a sign inversion in the exponent argument which has been corrected.

The solution [x] is also composed of the negative part of the spectrum. This is simply the complex conjugate of the positive part. In the temporal domain, the following manipulation is thus required for each harmonic :

$$\underline{A}_{-L}.e^{-j.w_0.t} + \underline{A}_L.e^{j.w_0.t} = \underline{A}_L^*.e^{-j.w_0.t} + \underline{A}_L.e^{j.w_0.t} = A_L.e^{-j.\varphi_L}.e^{-j.w_0.t} + A_L.e^{j.\varphi_L}.e^{j.w_0.t} = 2.A_L.\cos(L.w_0.t + \varphi_L)$$

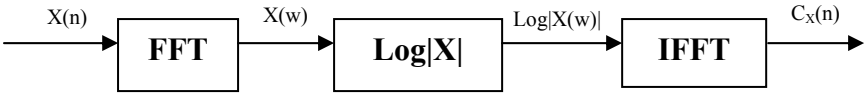
Note that the Gauss-Jordan matrix inversion has a cubed complexity. As the dimensionality of the matrix R can reach 70, this calculation is the most computationally expensive of the analysis process.

V.2.1.7 Estimation of the complex envelope

The computation of the complex envelope will permit us to reestimate during synthesis the new harmonics and phases at the new harmonic frequencies imposed by the prosodic modifications. However, during this synthesis process, we no longer have the original samples, only the HNM parameters. We distinguish 3 methods for obtaining the spectral envelope : by the LP (Linear Prediction) algorithm, by the Cepstral coefficients and obtained by a simple linear interpolation between the amplitudes and phases previously computed.

The calculation of the LP envelope consists in determining the a_i coefficients of an AR filter. These are determined by the Yule-Walker equations developed at point **IV.1.1**. Once these coefficients have been computed we have the Z transform of our filter. By applying the transformation $Z = e^{j\varphi}$, with φ representing the normalized frequency ($\varphi = 2.\pi.F = 2.\pi.\frac{f}{F_e}$), we finally have the isochronous response of the filter of which we can deduce the amplitudes and phases at the new harmonic frequencies. However, the minimum phase system gives a quite metallic sound.

This envelope can also be computed by a cepstral analysis. We don't go into detail, through lack of space and give only the principle of the calculation of the amplitude envelope. The real cepstral coefficients are obtained by the following way :



The resulting vector $C_X(n)$ is composed of the same number of samples as the starting vector. The envelope is obtained by computing a Fourier Transform (FFT) on the first (truncated) cepstral coefficients (the others are set to 0) and by paying attention to organize the vector as symmetrical (without considering the first coefficient). It allows to retrieval a real Fourier Transform, that is the amplitude spectrum (what consists of our starting point since we have considered the absolute value of the FFT).
We find again in the frequency domain the signal envelope because only the first coefficients (in the temporal domain) were considered (what ensues of the definition of the Fourier Transform where the first samples are associated to the low frequencies).

At last, the third method consists in building the envelope by a linear interpolation from the computed amplitudes and phases (during analysis) at the initial harmonic frequencies. Only a phase unwrapping is necessary. The resulting phase is bounded between $-\pi$ and $+\pi$, a phase unwrapping is so needed so as to ensure a coherent re-sampling during prosodic modifications in the synthesis part. This unwrapping is carried out by modifying the phase i of +/- $2*\pi$ such that the difference with the previous phase (i-1) is less than PI in absolute value. Figure 39 illustrates this principle (continuous and dashed lines correspond to the original and new harmonics, respectively).

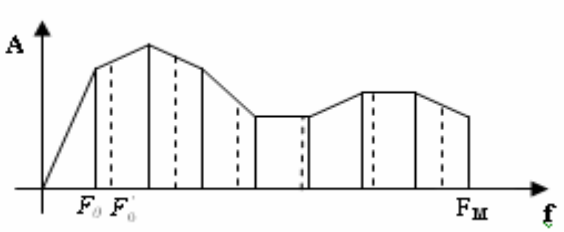


Figure 39 : Envelope estimated by linear interpolation of the amplitudes

I have used the third method because it doesn't increase the size of the database and provides good resulting speech quality. The synthesis will re-sample these envelopes at the new frequencies. No stored coefficient is so needed at this step.

V.2.1.8 Estimation of the noise parameters

In order to understand the different operations during the noise estimation, we can summarize the steps through figure 40, without forgetting the high-pass filter with cutoff frequency equal to the maximum voiced frequency F_M for voiced frames.

This graph is here given to better explain the steps but will be revised during synthesis.

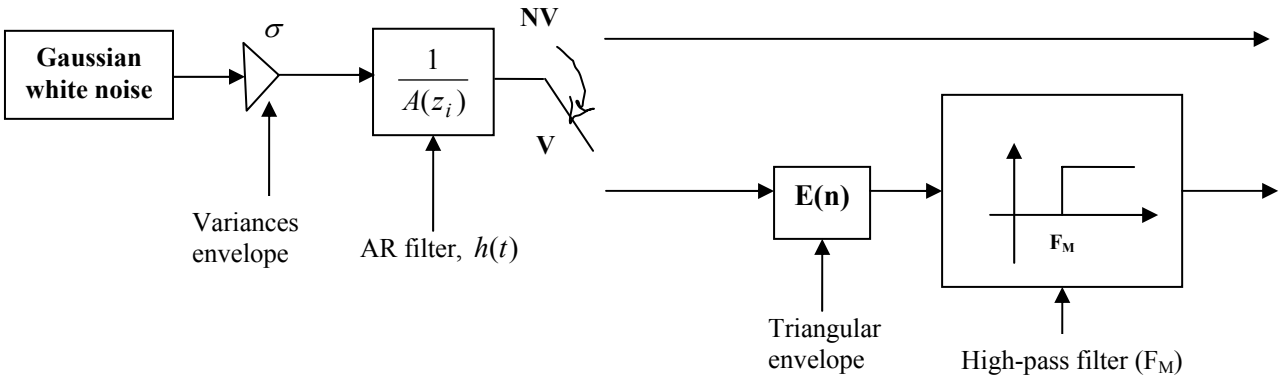


Figure 40 : Graph of the noise part generation for voiced and unvoiced frames

The variances envelope first has to be estimated every 2 ms (32 samples). The voiced frames have $2*T_0$ samples. So it this case the frame is divided into a integer number of 32-samples groups ; it remains thus a number of samples less than 32 at the end of the voiced frame. We apply then the following criterion : if this number is less than 10, the variance of the previous group will be valuable for these samples, in contrary, if this number is greater than 10, then a new variance will be computed for this last group.

A great difference (compared with the theory) is here built. It appears during synthesis that the variances envelope causes some discontinuities in the resulting speech. On advice of my supervisor this envelope is then computed on the prediction residual signal (obtained by the inverse AR filter) and not on the speech samples. The purpose of this envelope is mainly to model the transitive sounds of the speech which are still present in the residual signal.

The AR coefficients of the filter and the associated gain (standard deviation of the prediction residual signal) are also computed. We use the Shür algorithm with an order of 18. This needs the calculation of the biased autocorrelation function on the frame in question ; as the use of the unbiased autocorrelation results in an unstable filter (from the Yule-Walker equations). The Shür algorithm begins with the computation of the Parcor coefficients, so an extra operation is needed to reach the a_i coefficient of the filter. Recall that the AR filter is stable if the Parcor coefficients are bounded by -1

and 1, if the roots of the polynomial (constituting the denominator) are including into the unit complex circle (minimum phase).

At last the l_1 and l_2 parameters of the triangular envelope will be computed during the synthesis process because their calculations are very easy and storing them will needlessly increase the database.

V.2.1.9 Modification of the phase

We add to the theory an additional justification. This operation amounts to doing a "circular" translation of the samples. In fact, an interpretation of the Discrete Fourier Transform (DFT) of a succession of N samples is that it corresponds to Discrete Time Fourier Transform (DTFT) of the periodic signal of which the period is the N samples, except a multiplicative factor (T_0).
An illustration is presented in figure 42. The moving of temporal window over the periodic temporal signal (equivalent to bring a phase correction to each harmonic with a proportional factor of the number of the harmonic) amounts so to apply a “circular” translation to the samples.

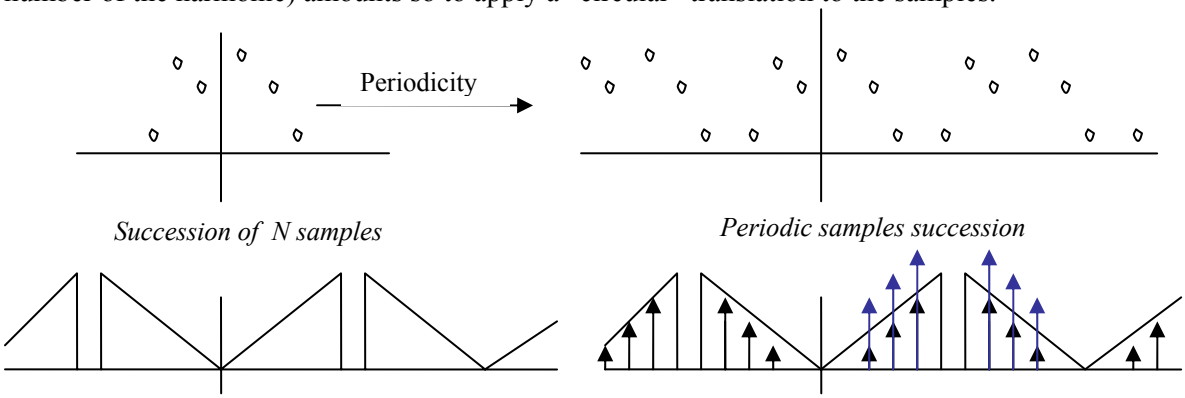


Figure 42 : Illustration of the circular property of the DFT

This phase modification will be run during synthesis since we need original unwrapped phases to build the envelope necessary for the prosodic modifications. This is run after pitch modification and during the harmonic generation process.

V.2.1.10 Computation of the gains of the noise parts

At last, we need add the calculation of the gains of the noise frames which isn’t discussed in the literature.
The gain of the unvoiced frames is easily computed by the measurement of the standard deviation of the temporal samples.
The gain of the voiced frames is nevertheless more difficult to estimate. On advice of my supervisor we start with a frequency approach. This decision is justified by the knowledge of the maximum voiced frequency and by the energy conservation of temporal signal in the spectral domain. Parceval theorem indeed tells us : $\sum_{n=0}^{N-1}|f(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1}|F(k)|^2$

So the estimation of the stochastic energy can be realized by summing the square of the amplitudes corresponding to the frequencies upper F_M than and by dividing this sum by the number of samples.

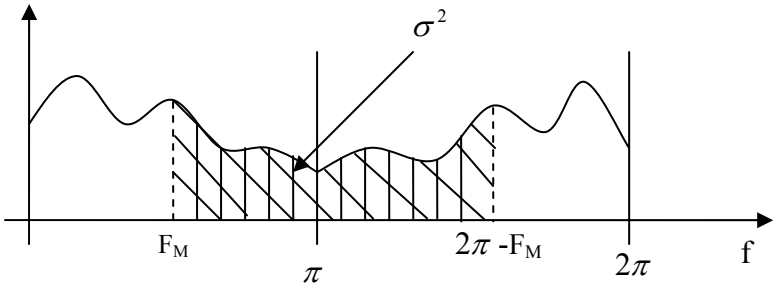


Figure 43 : Estimation of the noise energy for voiced frames

However it turns out that this method is lacking of accuracy since a median smoothing filter was used for the maximum voiced frequency. So this does not correspond anymore to the initial computed value, hence, this method provides a standard deviation sometimes too important compared with the harmonic level (especially on the ends of voiced groups) and so cannot be used alone for the gain adjusting. But we will even take it into account like an additional modulation. We will see during the synthesis part that this gain is finally adjusted by a heuristic way.

For the analysis process, the Fourier Transform module, the Shür algorithm (giving the AR filter coefficients) and the Gaussian matrix inversion could be obtained from existing C source code. Each part was programmed and tested separately. For these test phases, Matlab provides a efficient and fast checking tool. The different modules were then grouped and tested together on simple signals (sum of sinusoids) and later on the words of the database.

V.2.2 HNM synthesis

The synthesis process was divided into two parts. Firstly only a purely re-synthesis (no concatenation or prosodic modification) of the words of the database is carried out. The process "HNM analysis + purely HNM Re-synthesis" so corresponds to an HNM coding. This step justifies itself as an intermediate point between the analysis process and the delicate concatenation synthesis process. The results are excellent and only slight difference can be heard between the original database words and the HNM re-synthesized words. This validates all the analysis and the purely re-synthesis.

Secondly two test sentences have been generated by a manual concatenation. So the correct database nonsense words, the segmentation of the diphones in these words and the pitch and duration modifications are provided by Festival and written out to a file. In addition to the pure re-synthesis the concatenation synthesis brings a diphone concatenation, duration and pitch modifications and finally a smoothing of the discontinuities.

V.2.2.1 HNM re-synthesis

At the beginning, the stored HNM parameters of the word to be synthesized are retrieved from a database. We recall here the major steps. Note that the synthesis time instants are equal to the analysis ones since no prosodic modification is performed.

Figure 45 shows a complete bloc diagram of a synthesis realized at time t_s^i. The pitch modification and the discontinuities smoothing are represented but are not described here.

- Firstly the voiced/unvoiced decision is made.

- If the frame to be synthesized is unvoiced, only the noise part must be generated. The Gaussian white noise constitutes our starting signal. The analysis provides the variances σ_i^2 and the AR-

filter coefficients. The signal is modulated to begin with by the variances envelope and then by the AR filter.

● If the frame to be synthesized is voiced, the harmonic signal generation is done in time domain according to the $L_M(t_S^i)$, $A_k(t_S^i)$ et $\phi_k(t_S^i)$ parameters. $L_M(t_S^i)$ indicates the number of harmonics to be generated of which phases and amplitudes are known. We shall synthesize the following signal : $\sum_{k=1}^{L_M(t_S^i)} A_k(t_S^i).\cos(k.w_0(t_S^i).t + \theta_k(t_S^i))$. The phase modification based on the centre of gravity is added at this stage.

At the same time the noise signal is generated by the same way as for the unvoiced ones (high pass filtered white noise (with $f>F_M$)). To take into account the phase modification of the samples of the harmonic part, we add a shift of the noise samples.
At last the modulation by the triangular-like energy envelope is considered. This envelope is adjusted by taking into account the location of the harmonic energy burst which is now on the synthesis time instant (thanks to the phase modification). The l_1 and l_2 parameters are easily computed. On figure 44 we consider that the harmonic and noise bursts are synchronous, which is verified to be.

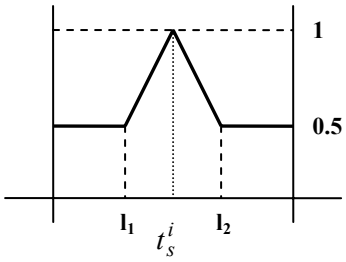


Figure 44 : Triangular window E(n) for synthesis

The noise and harmonic components of the voiced signal are finally combined.

● At last, the synthesis is run by an overlap and add process to give the resulting speech signal. A Hanning window is considered for its zero-value and its zero-value first derivative at its boundaries.

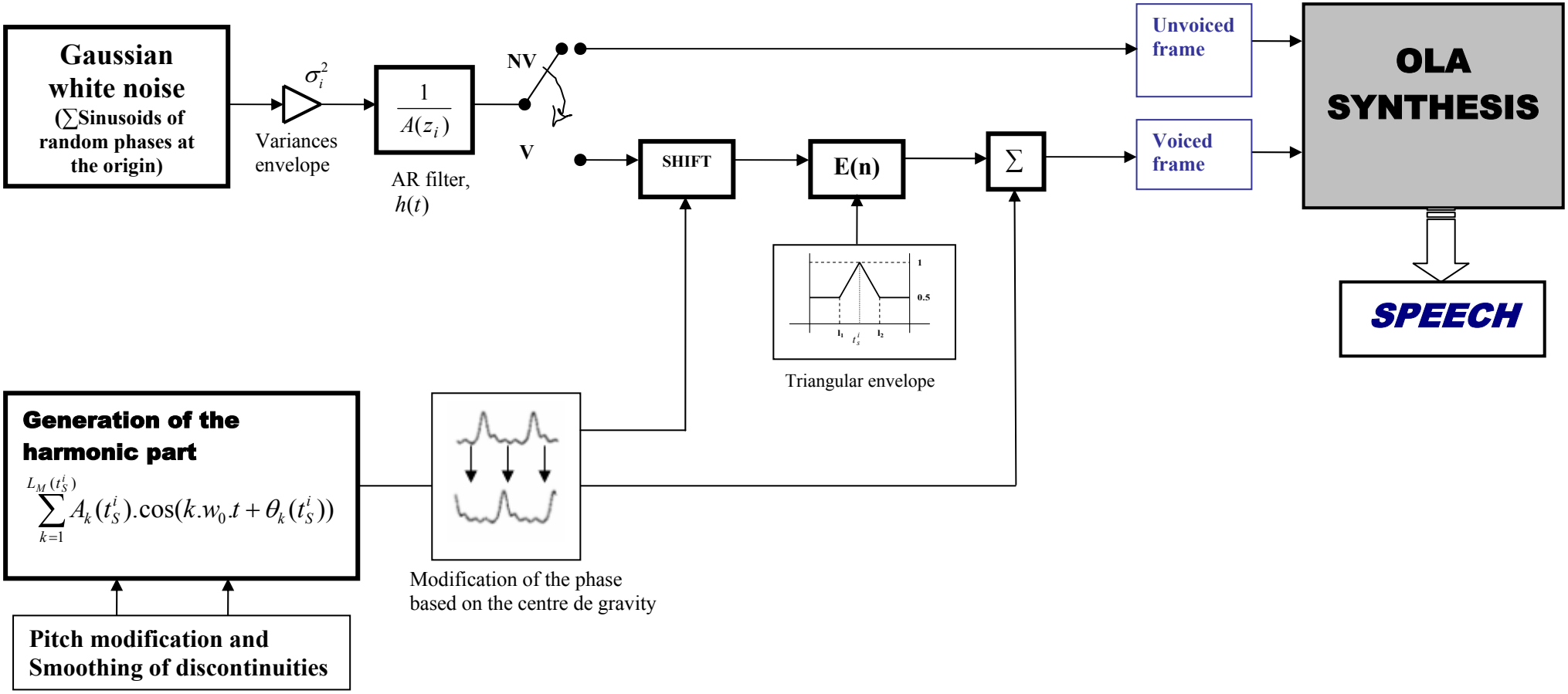


Figure 45 : Bloc diagram of the synthesis process

The programming is run like this : the synthesis of the harmonic and noise parts are separated. The harmonic one is firstly considered. It is stored in a table of structures, each them containing the harmonic samples, the length of the frame ($2*T_0$) and the position of the maximum sample (that will be useful to adjust the triangular envelope in the noise part).

The harmonic part is generated by applying the first direct method (see SF, **IV.4.4.1)** of generation of harmonics since the synthesis is running at real time on a P4 computer. Note that the initial time *t* (from which the sum of harmonics is computed) is equal to zero since the phase at the origin takes into account this reset (see HNM Analysis). The calculation is of course discrete and the time-increment corresponds to a sampling period T_S. Figure 46.a gives the superposition of an original signal (blue) and the corresponding synthesized signal (red). We notice the perfect correspondence between the two signals at the centre of the frame, what is interesting for us because of the Hanning window applied during the OLA process.

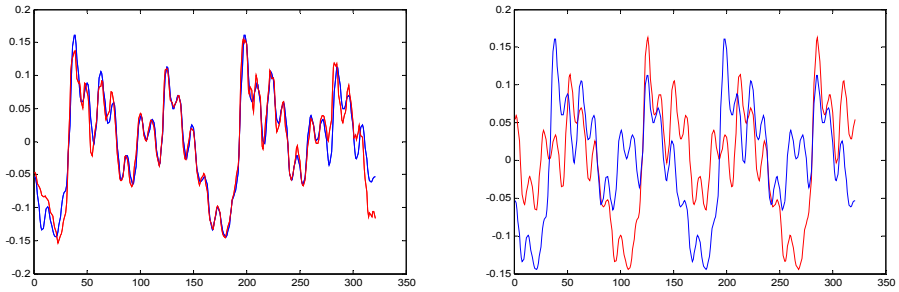


Figure 46 : Superposition of an original and synthesized signal (left) and phase modification (right) ; number of samples for X-axis and amplitude forY-axis

At last the phase modification based on the centre of gravity is applied. This modification proceeds a shift of the harmonics-generated window as illustrated by figure 46(b).

The noise part is then synthesized. By the same way it is stored in a table of structures.
A different Gaussian white noise for each frame is firstly generated. Of course if it was identical it would lead to a periodicity and so the introduction of low frequencies in the resulting noise signal. In the case of unvoiced frame the noise spectrum is ideally flat on the whole spectrum (the noise has to contain every frequency). On the other hand in the case of voiced frames, we avoid the using of the high-pass filter if the noise frequencies are upper to the maximum voiced frequency. We generate this part by a sum of harmonics (of which $f > F_M$) with unit-amplitudes and random phases at the origin [5]. We can observe that this noise is approximately Gaussian without any histogram modification, as shown by the following example :

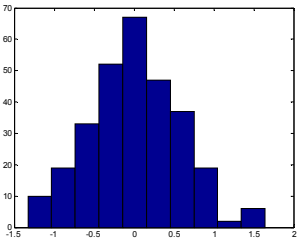


Figure 47 : Histogram of white noise generated by a sum of harmonics with random phases at the origin and unit-amplitudes

The signal is then normalized to have a unit-variances.

Then the variances envelope is applied by a simple multiplication in the temporal domain. As has been previously described, it has been computed on the residual signal since its computation on the original signal has produced some crackling in the resulting speech.

After it comes the AR filter (recursive). This modulation is realized in the temporal domain by the following equation :

$$y(n) = x(n) - a_1.y(n-1) - a_2.y(n-2) - ... - a_{18}.y(n-18)$$

Only the initialization of the filter causes trouble. The classical initialization (so the terms y(n-i) that don't exist are put to zero) and the initialization taking into account of terms of the previous frame has been tested to give the same result.

The voiced frames have yet to be modulated by a temporal triangular envelope. However in order to correctly adjust it, we first need to verify where the noise burst is located compared with the harmonic one. After some experiments on speech frames we see that these two bursts are synchronous (figure 48). At least we can conclude that the maxima fit, more or less. The maximum of the triangular envelope will not be exactly placed on the synthesis time instant (as previously discussed) but on the maximum of the harmonic frame.

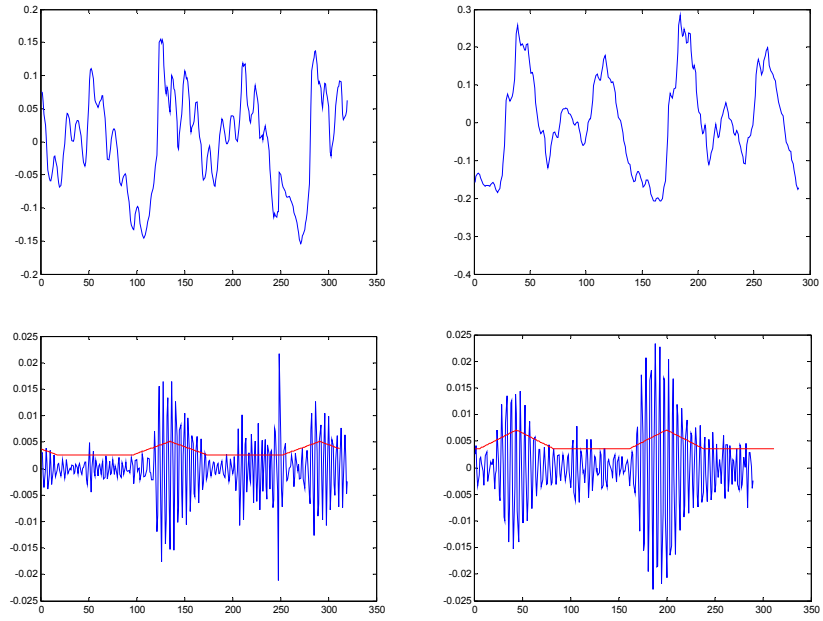


Figure 48 : Adjustment of the triangular envelope on the harmonic frame ; number of samples for X-axis and amplitude forY-axis

Some experiments have shown nevertheless that the resulting speech quality was not very sensitive to the position of this envelope and to the relative amplitude of its maximum to its flat part.

The noise frames are finally multiplied by a variable gain (so as to obtain a comparable variance to the original signal). The unvoiced gain allows us to exactly adjust the level of the output stochastic signal. On the other hand for the voiced frame, it is heuristically obtained.
We observe that following to the variances envelope, the AR filter and the triangular envelope, the gain computed as the product of the standard deviation of the residual signal (AR filter) by the one representing the energy of the high frequencies (see **V.2.1.10**) allows us to reach a sufficiently accurate level of output noise.
Figure 49 compares the temporal signals real "high frequencies" (right) and synthesized (left) of a voiced frame. We notice the good application of the triangular envelope (the correct gain is not already adjusted).

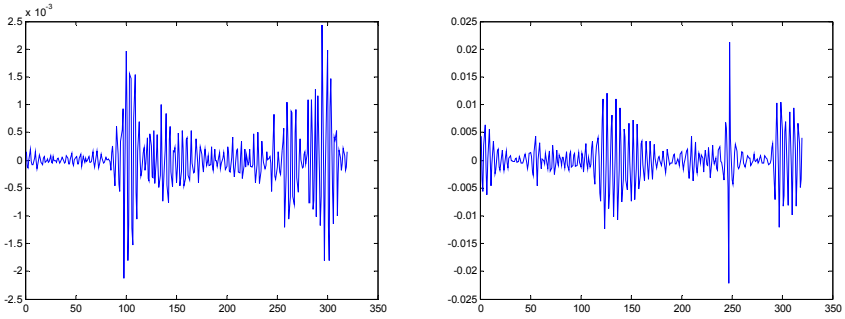


Figure 49: Comparison of the temporal signals "high frequencies" *real (right) and synthesized (left) of a voiced frame ; number of samples for X-axis and amplitude forY-axis*

The obtained purely re-synthesis is of excellent quality. Figure 50 illustrates the correlation between an original signal issued from the database (original) and the HNM re-synthesized one.

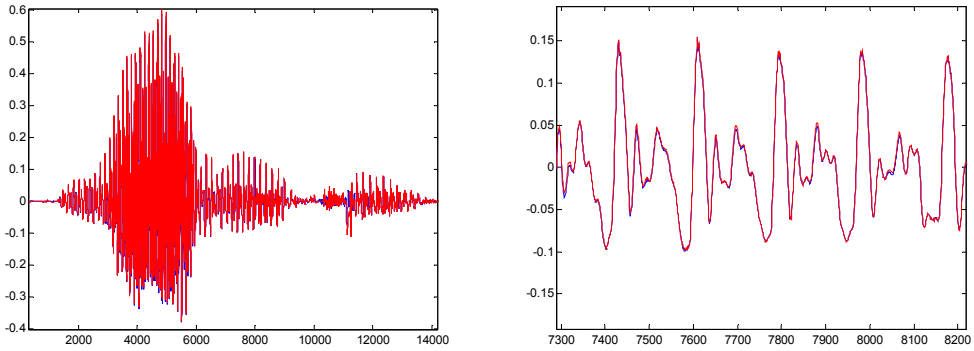


Figure 50 : Superposition of an original word (blue) and the same word synthesized by HNM (red) (left) ; zoom on the harmonic parts (right); number of samples for X-axis and amplitude forY-axis

In the same way about fifteen nonsense words from the database have been re-synthesized and have given a result comparable with the original. This permits us to validate the purely re-synthesis process and also all the analysis process. Additionally this permits us to use the whole system "Analysis + Re-Synthesis" as a high-quality speech coding. Let's appreciate in the next point the bit rate performance of our HNM coding.

V.2.2.2 Concatenation and prosodic modifications

We enter a more difficult part that consists in concatenative speech synthesis and the prosodic modifications.

Firstly *Festival* gives us the different diphones to be concatenated and the corresponding segmentation times. The first step consists of associating the segmentation instants to the closest analysis time instants. The transition instants between phonemes into each diphone are also computed. The necessary structures for reading files and for the segmentation are rapidly transformed in a single structure (containing only the useful analysis instants) so as to remain compatible with the purely re-synthesis programming.

Secondly the prosodic modifications are brought by geometrical relations between time axes. Firstly, the duration modification is computed, because the pitch modification given by *Festival* has already taken into account this duration modification. But these modifications are relative to the phonemes and not to the diphones. So two "boundaries" vectors are defined : the first for the phonemes and the second for the diphones. The first allows the duration modification and the second will be useful

during the discontinuities smoothing at the concatenation points, which is an operation computed between diphones.

A first virtual time axis (1) is defined to which the phoneme boundary instants correspond. Between these instants the number of virtual instants is equal to β_i (duration modification factor) times the number of analysis time instants ; which permits us to define a constant step (STEP) between the virtual instants for each phoneme. Then these first virtual time instants are associated to the analysis ones by the closest instant method. Lastly, since their fundamental periods haven't been modified, the virtual instants are pitch synchronously mapped to a new virtual axis (2). Let's note that the number of the first virtual time instants is equal to the second one and that the duration has been well modified by deleting or duplicating analysis time instants.

Secondly, it is the pitch modifications are carried out. The pitch curve and the second virtual time axis have the same length, so it becomes easy to sample this curve[5] at the different virtual time instants 2. Each pitch value of the virtual instants is mapped to a new value. The evaluation of the new harmonics and phases is achieved by re-sampling the envelopes by linear interpolation.

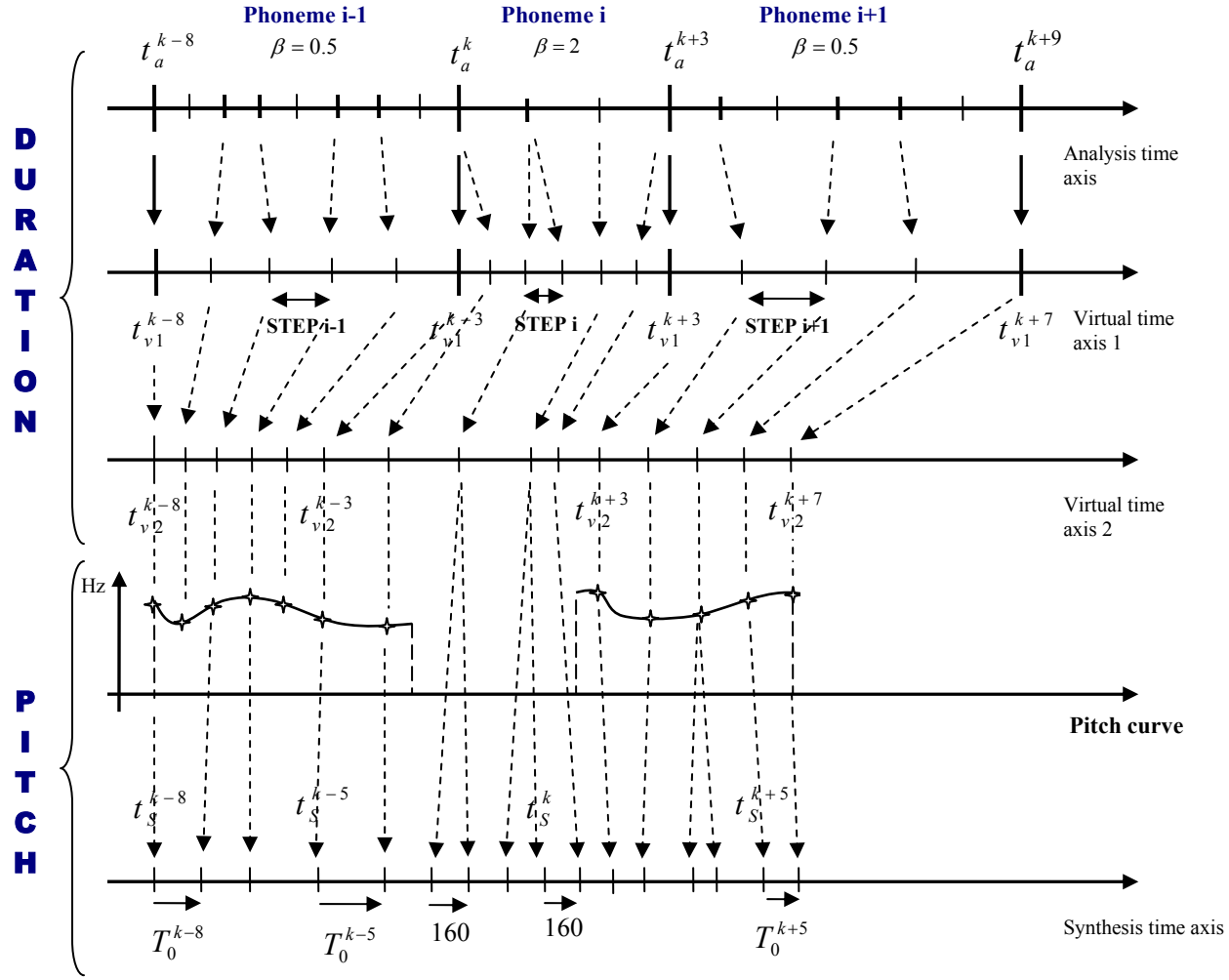


Figure 51 : Duration and pitch modifications

Let's notice that the maximum voiced frequency remains identical, so the number of harmonics can be less than the previous one (if $f_0^2 > f_0^1$) or greater than (if $f_0^2 < f_0^1$). In addition, an energy correction

[5] Actually this pitch curve is also discrete and sampled at every 10 ms. The method of the closest instant is a new time applied for the correspondence with the second virtual time instants.

factor is applied so as to keep constant the sum of the square amplitudes ($\sum_i a_i^2$ = cst), that is to say the same energy. At last, the synthesis time instants are computed according to the new fundamental frequency by the recursive relation : $t_S^{i+1} = t_S^i + T_0^i$ (where T_0 = 160 if the frame is unvoiced) where t_S^{i+1} is associated to the closest second virtual time instant.

We finally observe that the time axis has been properly modified by respecting the pitch modification. The prosodic modifications, every geometrical relation and instant associations are illustrated in figure 51.

As the last stage, a smoothing of the discontinuities is performed at the concatenation points.

Let's look again the details from the theory and let's draw a graph. U_l and U_r are the two acoustic units, on the left and on the right of the concatenation (interpolation) point ; w_0^i and A_k^i are the pitch and the k amplitudes of the harmonics of the last frame of U_l and w_0^{i+1} and A_k^{i+1} the ones of the first frame of U_r. The smoothing is performing from the concatenation point and propagated on each side of this point.

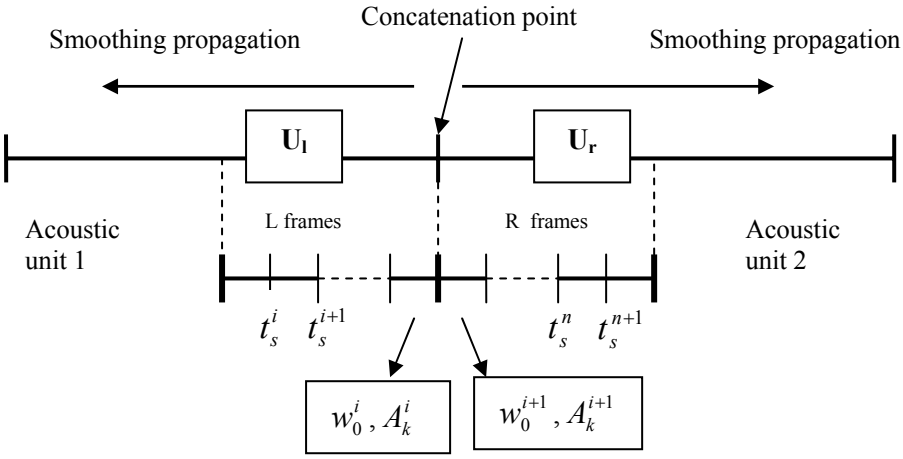


Figure 52 : Smoothing of the discontinuities

Recall that the phase discontinuities are already solved by the phase modification based on the centre of gravity. Only the fundamental and the formants will be so considered. The noise discontinuities are also removed from this process since a noise parameter discontinuity remains imperceptible.

The boundaries of the diphones are known, so we evaluate the pitch difference (very low since it is just modified by the pitch curve which has a low first derivative) and the amplitude differences of each harmonic. Note that if the pitches were very different at this level, we should estimate the amplitude differences not between the same harmonics but between harmonics of which frequencies are the closest, because it's the formant discontinuities that are being smoothed. This difference is evaluated until the lower maximum voiced frequency of the two instants around the concatenation point and is propagated – also for the pitch difference – on each side of this point with a different weight until the middle of each diphone.

```
analtimeinst[BEGIN-i].pitch=analtimeinst[BEGIN-i].pitch+Delta_W0*(L-i)/L;
analtimeinst[BEGIN+1+i].pitch=analtimeinst[BEGIN+1+i].pitch-Delta_W0*(R-i)/R;
```

So, we see that, thanks to its parametrical structure, the *Harmonic Plus Noise Model* allows us to reduce the formant discontinuities at the concatenation points.

The final structure containing the synthesis time instants is given to the re-synthesis module which computes the harmonic and noise parts and synthesizes the signal by OLA, exactly in the same way as explained earlier.

V.3. Results obtained with this HNM implementation

The quality of a speech synthesizer depends on some parameters :

- The database from which are extracted the different diphones.
- The corpus segmentation quality.
- The quality of the natural language processing modules.
- The quality of the prosody target computation (pitch and duration).
- The quality of the speech waveform synthesizer (which interests us in this work).

In order to correctly evaluate the qualities and defects tied to this implementation, we compare the results with those obtained by RELP synthesis (Residual Excited Linear Prediction, by default in *Festival*). The LPC excitation is then realized by the residual signal, which endows it with a high intelligibility quality ; it remains however quite "metallic" and expensive to store in memory. This comparison is done only between speech waveform synthesizers since the language processing modules are the same for both syntheses.

These tests were realized on 6 English native listeners and 6 non-native ones. The test consists of listening to 10 sentences generated with RELP and HNM and a preference decision is made (ABX test) on 3 criteria : the intelligibility, the naturalness and the pleasantness. The intelligibility aims to measure the understanding of the voice, the naturalness, its human or robotic feature, and lastly the pleasantness, aims to the choice of the most pleasant voice to listen to. These 10 sentences were distributed among 5 groups of 1,2 or 3 sentences so as to sometimes judge the pleasantness on a longer duration. These tests, realized at IDIAP, are of course very informal and are useful only to reveal the features of the HNM voice. The results are presented in table 7.

Native English	Pleasantness	Naturalness	Intelligibility
HNM	12	11	12
RELP	13	10	15
Equal	5	9	3

Non native English	Pleasantness	Naturalness	Intelligibilitity
HNM	10	10	11
RELP	15	10	18
Equal	5	10	1

Table 6 : Test comparison results between RELP and HNM with pitch modification

At this stage it appears that the RELP synthesis remains still better than HNM. Nevertheless these results are explicable : the main drawback of this HNM implementation is the integration of the noise part into the harmonic part after the pitch modifications. This operation gives rise to a "resonance" in the synthetic speech, which decreases its ratings on the 3 criteria and waters down the advantages revealed during its work. Indeed, behind this troublesome resonance, a more natural voice confirms the advantage of the HNM coupling of the excitation and vocal tract modelling.

So as to reveal this greater naturalness, a second test was proposed, taking sentences whose pitch issued from the database already corresponds (more or less) to the natural one (such as it would have been modified by the pitch curve). Then no pitch modification is needed for HNM synthesis and the resonance isn't present anymore. In contrary, the RELP synthesis is done with the pitch modifications. Results are presented in table 8.

Native English	Pleasantness	Naturalness	Intelligibility
HNM	12	15	12
RELP	12	10	13
Equal	6	5	5

Non native English	Pleasantness	Naturalness	Intelligibilitity
HNM	11	16	12
RELP	14	9	13
Equal	5	5	5

Table 7 : Test comparison results between RELP and HNM without pitch modification

The superiority of the HNM voice on the natural criterion is well perceived among the listeners. The intelligibility becomes equivalent since no resonance is present. However, the pleasantness remains depreciated, mainly owing to some pitch discontinuities in the HNM synthesis. It is indeed impossible to find a sentence whose stored pitch exactly corresponds to the real pitch. These results demonstrate all the potential of the HNM voice, which, for an identical intelligibility compared with the RELP voice, shows higher naturalness and pleasantness potentialities.

In addition, the size of the HNM database is 22 Mbytes memory, while the RELP requires double storage space.

It emerges from this study that the diphone segmentation is correct. The duration modifications are also very accurate and correspond to the RELP voice. At last, the intonation (pitch) is identical to the one of RELP. This validates the geometrical relations and time-scale operations (see figure 51). Additionally, the smoothing process removes effectively the formant discontinuities, whic remains less perceptible, but still observable on a spectrogram.
The combination of the pitch modification with the integration of the noise part has however to be improved in order to eliminate this "resonance" phenomenon. An energy correction factor was introduced in the spectral domain during pitch modification (see point **V.2.2.3**) and a great part of work was also devoted to the estimation of the noise level (by estimating the standard deviation, see point **V.2.1.10**) and to parameter tuning of the algorithm. The results were slightly improved but remains however inadequate.

V.4 Integration into Festival

The generation of sentences by the HNM synthesizer is based on the natural language processing modules of *Festival*, but its integration has not been done, through lack of time. The diphones, the time segmentation, the duration and pitch modifications are written out to file. The HNM synthesizer reads in accesses this file and reads in the database the correct diphones, before their segmentations for processing the waveform synthesis.

The integration into *Festival* needs the creation of a new sub-directory containing the HNM programming and the creation of a "Makefile" consistent with the architecture of *Festival*. Then the indication of the presence of this new module is done in the configuration of *Festival* (directory ‘festival/config/config’), and the compilation of the whole allows ensuring their compatibility.
Once this step has been done, the HNM programming has to be modified so as to take into account the classes of *Festival* which are needed to access the *features* of the natural language processing and so find the diphones and prosodic modifications. This integration needs around 3 weeks of work.

VI. Application to speech coding

Let's apply for information only our *Harmonic Plus Noise Model* to speech coding and let's calculate the obtained bit rate with in superficial way. Our HNM database contains a number of parameters tied to this implementation. Let's count by average 50 phases and amplitudes for voiced frames and 10 variances, plus a voiced decision V, the pitch value, the 18 AR-filter coefficients and standard deviation of the residual signal and finally the standard deviation of the unvoiced energy. For unvoiced frames, only the 18 AR-filter coefficients, 10 variances, standard deviation of the signal and the unvoiced decision NV have to be taken into account. The structure associated to an analysis time instant is shown below :

```
typedef struct
{
        float pitch;
        int voiced_decision;
        int LM;
        float *amplitudes;
        float *phases;
        float ai[19];
        float sigma_pred;
        float std_signal;
        int numberstd;
        float *std;

} ANALYSISTIMEINSTANTS;
```

In accordance to the voiced decision some parameters won't be needed. Let's note that the *LM* (number of harmonics) and *numberstd* (number of standard deviations or variances) parameters can be easily computed during the decoding process, so it is useless to encode these.

By assuming that the number of voiced frames is equivalent to the unvoiced one, the average number of coefficients to encode per frame will be :

$$\frac{(50+50+10+18+1+1+1+1)+(18+1+10+1)}{2}=81 \text{ (coefficients).}$$

The previous analysis is pitch synchronous for voiced frames and 10 ms synchronous for the unvoiced ones. So according to the pitch value our bit rate coding will be different. Let's consider 2 cases, a female voice of 250Hz and a male voice of 100Hz. In this last case for voiced and unvoiced decisions, the parameters are estimated every 10 ms. In the first case nevertheless we need taking an average value (by considering a second time the number of unvoiced frames (4 ms) equal to the voiced one (10 ms) : $\frac{10+4}{2}=7ms$. The fair sex will so use more resources… since by average 11570 coefficients / sec will be necessary when 8100 / sec will be sufficient for masculine voice coding.
Let's analyse these performances according to the bit-coding number per coefficient and let's begin by a highly expensive but highly-accuracy coding :

- 1 bit for the V/NV decision.
- 8 bits for the pitch, which must be accurately encoded, so here with 256 values distributed on a about 300Hz-scale, it represents a quantization error of about the half Hz, which is highly accurate.
- 8 bits for the first 10 amplitudes and phases and 4 bits for the others. Actually it is the first harmonics which are the most important since they give the low frequency energy.
- 4 bits by average for the AR-filter coefficients. Actually it is the Parcor coefficients that are encoded and the first coefficients are encoded on a greater number of bits

because their importance in the temporal signal correlation is bigger. This coding corresponds to the LPC coding.

- 4 bits for the standard deviations and variances.

Which gives :

$$\frac{(10*2*8+40*2*4+10*4+18*4+1*8+1*1+1*4+1*4)+(18*4+1*4+1*1+10*4)}{2}=363bits$$

by average per frame.

By this coding the quality should be close to the one we obtain, but it requires significant resources. By using a more complex quantization scheme, we can reduce a lot our bit rate but after a decreasing of quality. Let's consider only 10 harmonics and phases and let's eliminate the variances envelope and the standard deviation image of the unvoiced energy (for voiced frame) which ones don't bring a important added quality. Let's begin again our analysis :

- 4 bits for the pitch
- 4 bits for the amplitudes and phases
- 3 bits by average for the AR-filter coefficients
- 1 bit for the V/NV decision
- 4 bits for the standard deviations

Which gives :

$$\frac{(10*2*4+18*3+1*4+1*1+1*4)+(18*3+1*4+1*1)}{2}=101bits$$

The resulting bit rates are summarized in table 6. We should af course validate these results by a qualitative listening but it is not the subject of my work. The results are here given for information only in order to provide a rough estimate of the performances obtained with the coding tied to this HNM implementation. We notice however that an average bit rate of about 12 kbit/s can be reached by the second method. This bit rate could yet be decreased if the LPC envelope is accurate enough to re-compute the harmonics. This bit rate could largely be decreased if only the LPC coefficients are enough for the generation of an enough accurate envelope for the computation of the harmonics. But we have to keep in mind the analysis and synthesis computation complexities are so that a *Harmonic Plus Noise* coding has to be justified.

	Method 1	Method 2
Male voice	36.3 kbit/s	10.1 kbit/s
Female voice	51.9 kbit/s	14,4 kbit/s
Average	44.1 kbit/s	12.25 kbit/s

Table 8 : Bit rates obtained by a HNM coding

VII. Hidden Markov Models based speech synthesis

VII.1 Principles of the HMM-based speech synthesis

The current synthesizers permit producing a high quality of speech if a large database is used (unit selection-based synthesis). However they realize a neutral voice according to the recorded database. The Hidden Markov Model based speech synthesis (HMM-Based Speech Synthesis, HTS) tries to solve it [15,16,17,18,19,20].

The Hidden Markov models are reputable to offer a good modelling of temporal succession of events. They can taken into account emotions, individualities and styles in the synthesized voice. The basis principle firstly consists in training the Hidden Markov Models (HMMs) by taking into account contextual factors. Then, during the synthesis process, the speech is generated from the parameters of the HMMs, by trying to maximizing their likelihood.

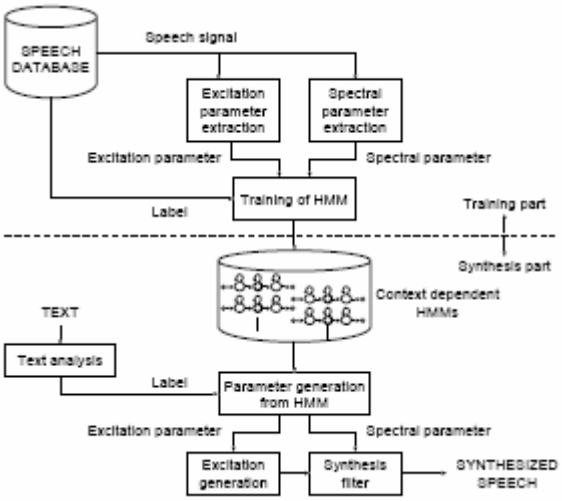


Figure 53 : HMM-based speech synthesis system

The training of the HMMs needs the extraction of the excitation and spectrum parameters and also a set of contextual factors (as the number of stressed syllables in the word, the position of the word in the sentence,…). In its current version, HTS uses the MEL-Cepstrum coefficients for the spectrum (and its delta and delta-delta coefficients) and the logarithm of the pitch (and its delta and delta-delta coefficients) and voiced/unvoiced decisions. These coefficients will be modelled by the context dependent HMMs. With this set of parameters, the waveform synthesis can be done using a MLSA (Mel Log Spectrum Approximation) filter.

Let's note that the contextual factors are extracted by a text analysis engine of *Festival*.

First, the synthesis extracts form the text to synthesize a context dependent factors sequence. Then according to this sequence, the chains of Hidden Markov Model are selected are concatenated. Lastly the f_0 and Mel-Cepstrum parameters (and their delta coefficients) are obtained from these HMMs by the maximization of the likelihood of these data's. These coefficients are then presented to the MLSA filter which one generates the speech signal. A summary graph is shown in figure 53.

By the transformation of the HMMs parameters (modelling of the coefficients), this approach allows modifying the voice and bringing emotion and other particular features. These HMMs parameters transformation can be run by a speaker adaptation technique or a speaker interpolation technique.

VII.1.1 Training part

The training of the HMMs consists in adjusting the probability distributions of the different parameters : cepstrum, pitch coefficients and the duration of the states.

For the modelling of the spectrum, a Multi-Gaussian probability distribution is used as emission probability (1 single Gaussian per dimensionality). The number of dimensionalities corresponds to the number of cepstrum coefficients.

For the modelling of the excitation, a multi-space probability distribution is used. This distribution has been modified for taking into account the symbols used for unvoiced decision (Multi-Space Probability Distribution HMM, MSD-HMM). The pitch curve indeed represents sometimes the pitch values in the case of voiced frames (dimensionality 1) and sometimes the symbol corresponding to the unvoiced decision (dimensionality 0). This model takes into account the continuous and discrete HMMs.

The probability density of the state duration is supposed as Gaussian.

Some contextual factors influence the spectrum, the pitch and the duration of the phoneme. They are considered in the modelling of the probability distributions to take into account the acoustical variations associated to these factors. However the number of contextual factors can be important to ensure a correct modelling. Their combinations become then enormous and no database can take into account every possible case. Even assuming such a database available and segmented, the EM (Expectation-Maximization) algorithm which computes the probability distributions should be difficult to run. Nevertheless this problem is overcome by applying a previous clustering of the training data's in order to estimate the mean and the variance of each data class. The EM algorithm can then be applied with a much more limited data and an equivalent accuracy.

This clustering is realized from a decision tree and is independently applied to the spectrum, pitch and duration since these three parameters are influenced by their own contextual parameters.

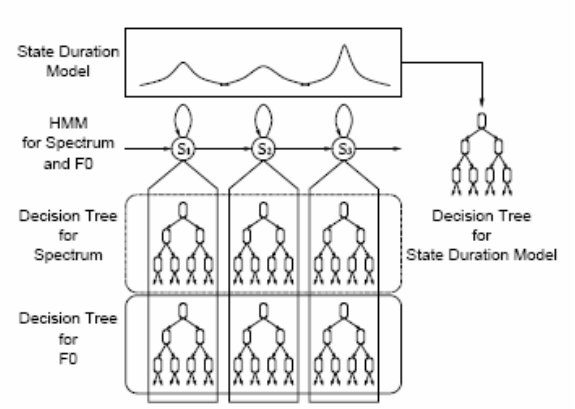


Figure 54 : Decision trees for context clustering

VII.1.2 Modelling of the pitch by the HMMs

The notion of multi-spaces probability distribution has been extended so as to take into account the discrete and continuous distributions (pitch values and unvoiced decisions) [16]. This kind of Hidden Markov Model is known as Multi-Space Probability Distribution HMM (MSD-HMM). It amounts to a discrete distribution if the space dimensionality is zero and to a continuous multi-spaces distribution if the dimensionality is greater than zero. It is based on an union of G spaces : $\Omega = \bigcup_{g=1}^{G} \Omega_g$, where Ω_g

is a n_g dimensional real space. Each space has got its own probability $P(\Omega_g) = w_g$; and so $\sum_{g=1}^{G} w_g = 1$. Besides if the dimensionality is greater than 0, this one has its probability distribution $N_g(x)$, where x is a n_g dimensional vector ; so we have also $\int_{-\infty}^{\infty} N_g(x).d(x) = 1$.

We suppose that if $n_g = 0$, then the space Ω_g contains only one point (discrete symbol representing the unvoiced decision).

The application to the modelling of the pitch leads to a one-dimensionality spaces representing the voiced frames (where a pitch is defined) and a zero-dimensionality for the unvoiced space. We have so $n_1 = 0$ and $n_g = 1$ (g=2,3,…,G).

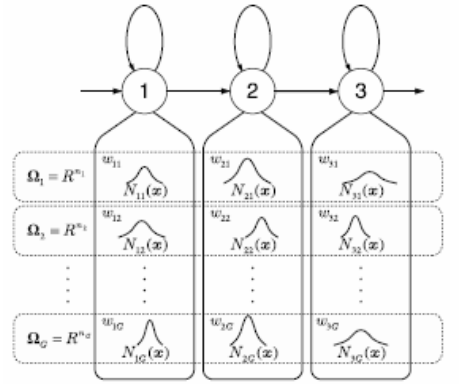


Figure 55 : HMM based on multi-space probability distribution

The emission probability of each state is then defined as : $b(o) = \sum_g w_g.N_g(x)$. If the dimensionality of the space n_g is0 then $N_g(x)$ is 1. The space indeed contains only one point (the unvoiced sylmbol). Figure 55 shows a G-dimensional space for the modelling of the pitch. We have in this case the following Gaussian probability distributions: $N_{i1}(x)$, $N_{i2}(x)$,…$N_{iG}(x)$ and their respective weights : w_{i1}, w_{i2},…, w_{iG} with $\sum_{g=1}^{G} w_{ig} = 1$. The weights and probability distributions of the λ model are estimated during the training algorithm so a to maximize the likelihood $P(O \mid \lambda)$, for a given observation sequence O.

VII.1.3 Generation of the HMM parameters

The generation of the HMMs parameters is based on the maximum likelihood criterion [15]. It is indeed aimed to maximize :

$$P(O \mid \lambda) = \sum_Q P(O, Q \mid \lambda)$$

where λ represents the HMM model, O the parameters sequence to determine and Q the visited state sequence. The problem is simplified (as the Viterbi algorithm) by maximizing the following equation :

$$P(O \mid \lambda) \approx \max_Q \sum_Q P(O \mid Q, \lambda).P(Q \mid \lambda)$$

In addition this problem is decomposed into 2 sub-problems :

1) $Q_{\max} = \arg.\max_Q P(Q \mid \lambda)$
2) $O_{\max} = \arg.\max_O P(O \mid Q_{\max}, \lambda)$

The first problem is solved by the Gaussian state duration hypothesis. The HMMs sequence is indeed selected according to contextual factors of the input text. This sequence is supposed as linear ("left-to-right HMMs"), what means that a transition can be done to the next state or the same occupied one. There from the probability of the state sequence $P(Q \mid \lambda)$ can be re-written as the product of the state duration probabilities : $P(Q \mid \lambda) = \prod_{k=1}^{K} p_{qk}(d_{qk})$, where $p_q(d)$ represents the probability of d consecutive observations in the same state q, and K, the number of visited states. So the maximization of this equation gives us the most probable sequence of HMM states.

The second problem uses the knowledge of this most probable state sequence Q_{max} to generate the most probable coefficients sequence (cepstrum and pitch) [15] ; the introduction of dynamic coefficients allows avoiding an output corresponding to the mean values and giving more continuity in the parameter generation. Figure 56 gives an example of this parameter generation.

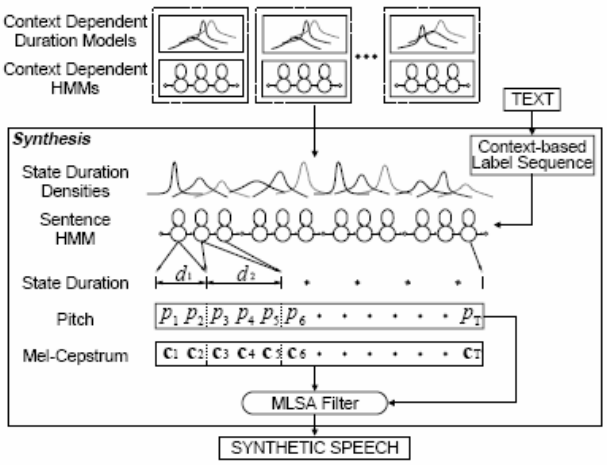


Figure 56 : Synthesis part of the HMM-based synthesis

VII.2 Advantages and drawbacks of the HMM-based synthesis and comparison with the concatenation synthesis

The advantages of the HMM-based synthesis [18,23] compared with the concatenative one can be summarized as followed :

- By the modification of the HMMs parameters, we easily introduce personal features in the voice (emotions, styles,…). These modifications can be brought by an speaker adaptation or speaker interpolation technique. In contrary, the concatenative synthesis only allows to product a neutral voice (characteristic of the databases).
  In addition, the HMMs and the introduction of dynamic coefficients allow avoiding discontinuity problems observed in the concatenative synthesis
- It is very more stable and "smooth" than its rival of which the quality decreases because of a bad unit selection.
- Thanks to its clustering, this synthesis permits more fast development of new voices compared with this one based on unit selection.
- It requires less memory storing than the unit selection-based synthesis.
- At last this synthesis can allow reaching low bit rates in speech coding and also the development of new techniques for people identification.

The main drawback of this synthesis nevertheless remains its quality quite average. It is indeed based on filter and so permits only to generate a quite metallic voice. In contrary the concatenative synthesis ensures a more natural quality, closest to the human voice.

VII.3 Principles of the integration of the HNM module into the HMM-based synthesis

The *Harmonic Plus Noise* synthesis developed in this project has the potential to allow more naturalness to the speech generated by the HMMs and so solve its main drawback. The main quality of the HNM synthesis shown in point **V.3** indeed demonstrates that it permits us to avoid the metallic feature of the filter-based syntheses.

This HNM integration constitutes a research domain in the IDIAP institute and following this project. We can already say that the HNM analysis will replace the mel-cepstrum and pitch analysis. In the same way the MLSA synthesis will be replaced by the HNM synthesis.

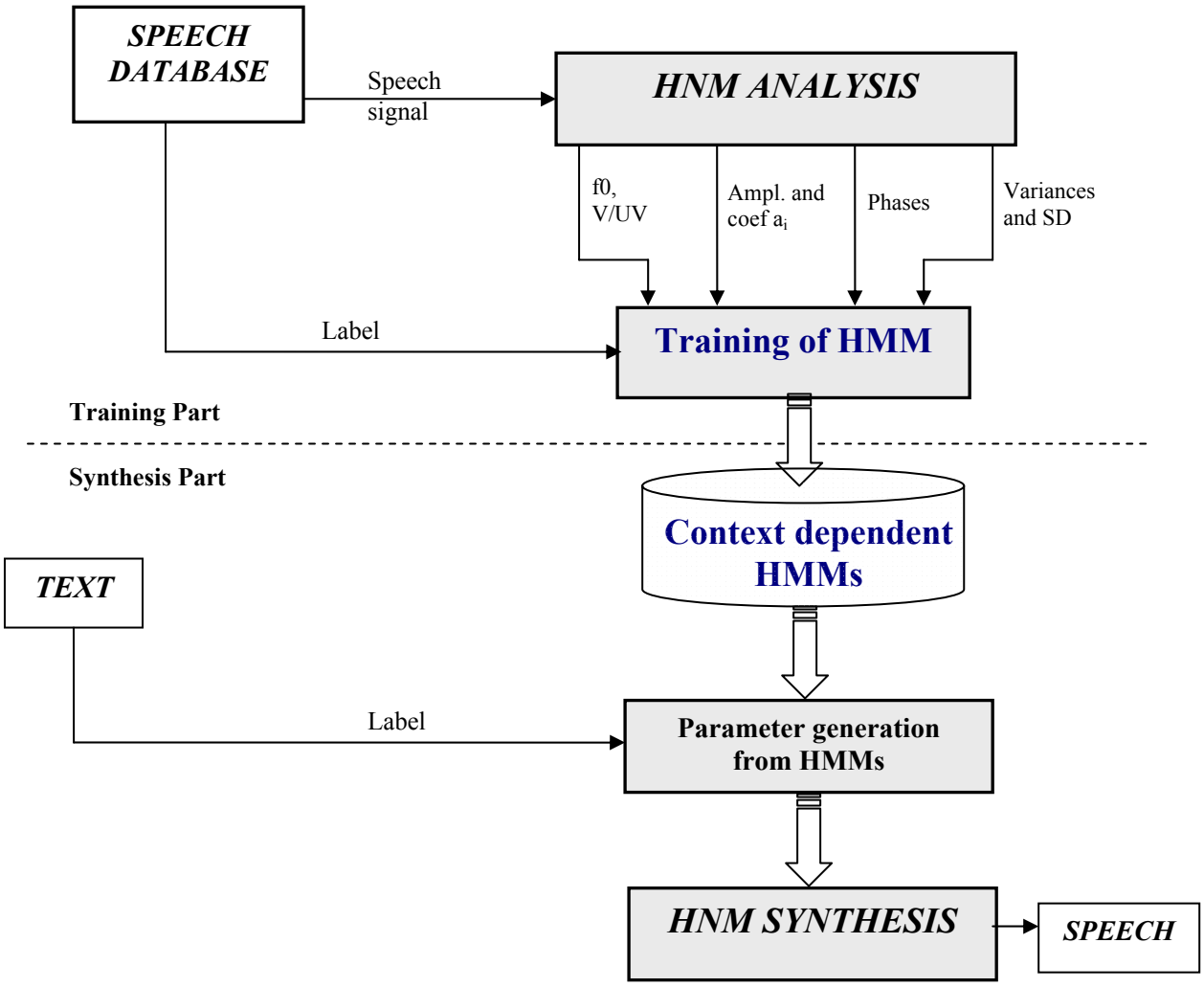


Figure 57 : HNM integrated into the HMM-based speech synthesis

The difficulty rests on the training of the HMM parameters which correspond now to the HNM parameters and no more to the cepstum coefficients necessary to the previous synthesis. These HNM parameters have thus to be modelled by suitable representations. The dynamic coefficients have also to be considered.

Let's firstly remind the parameters useful to the HNM synthesis :

VOICED FRAME :

- Pitch and V decision
- Number of harmonics L_M
- Amplitudes and phases of the harmonics
- 18 coefficients of the AR filter
- 10 variances
- 2 standard deviations

UNVOICED FRAME :

- Decision NV
- 18 coefficients of the AR filter
- 10 variances
- 1 standard deviation

Some parameters are only useful for voiced frames (pitch, L_M) ; on the other hand, the harmonics and phases need another representation.

The pitch and V/UV decisions may be modelled by the same way, that is the Multi-Space Probability Distribution HMM (MSD-HMM) that allows us to group pitch values and unvoiced decision in an unique framework. This is previously described in the literature.

The number of harmonics can be replaced by the maximum voiced frequency F_M, so as to group together the modelling of the pitch and F_M in a single MSD-HMM system, since these 2 parameters are (more or less) independent, while the number of harmonics is correlated with the pitch value. It is more accurate to model the pitch and F_M, from which it is easy to deduce the number of harmonics. In order to assure an efficient clustering for context dependent modelling, the pitch and F_M streams will be separated. So, each stream will be modelled by a sum of one-dimensional Gaussians.

The amplitudes, phases and AR coefficient are more difficult to model. A solution consists in grouping together the amplitudes values and AR coefficients in an unique system : the amplitude spectrum. Indeed, the amplitudes correspond to the spectrum until the maximum voiced frequency, while the AR coefficients represent the signal envelope for the whole spectrum. A Multi-Gaussians distribution (GMMs) could model a limited number of coefficients representing the envelope of amplitudes whose resampling will be then necessary at the harmonic frequencies during the synthesis process. However, this envelope will have to be accurate enough to ensure a sufficient quality of the HNM synthesis.
The generation of the LPC coefficients (AR filter) necessary to the synthesis can be (re-) obtained by approximating a new LPC envelope on this one generated by the HMMs. However this method can suffer of a lack of accuracy and a multiplication operation directly in the spectral domain (between the white noise and the amplitude envelope obtained by the HMMs) should perhaps be preferable. Indeed phases don't matter in the noise part generation. This should require modifying the current HNM synthesis since a Fourier Transform should be applied to the white noise (modulated by the variance envelope) and also an Inverse Fourier Transform after the modulation by the spectral envelope.

The variances envelope and the standard deviations may be modelled by a mixture of Gaussians (GMMs). In order to have a same number of variances per frame, we can consider a number of 10 for the voiced frames. In the same uniformity way, the standard deviation will be determined on the entire speech frame for the voiced and unvoiced frames during the analysis process ; and we will compute again this corresponding to the unvoiced energy for the voiced frames during the synthesis process. We have indeed the maximum voiced frequency and the complete energy of the absolute spectrum (variance of the signal).

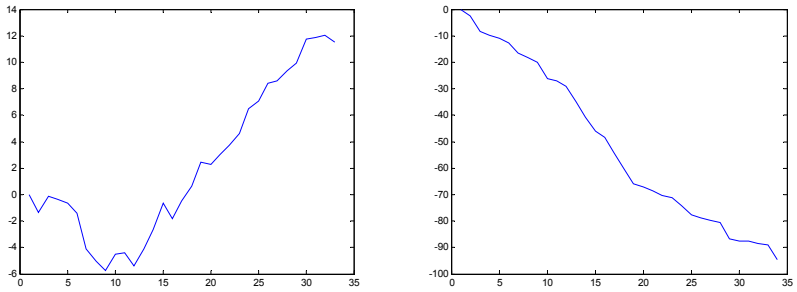


Figure 58 : HNM phase unwrapping (left) and linear (right)

The phase constitutes a last quite delicate point since the processed unwrapping (point **V.2.1.7**) gives us a completely random phase envelope. However, by unwrapping this phase in a continuously decreasing way, we can reach a more or less linear and negative sloping envelope. This unwrapping has been tested during the HNM programming and has given an equivalent quality of resulting speech. So the modelling of this phase could be summarized to the angular coefficient of the straight line (moving by this origin since signal is real) that approximates the phase envelope. Some tests will have to be run so as to establish the probability distribution of this angular coefficient. A more complex model could also model some coefficients allowing a more accurate approximation of this unwrapping.

The different energies and the absolute spectrum may be model by a single GMM system. The phases may be integrated to the first MSD-HMM model since it concerns only the voiced frames and a voicing decision is modelled at this stage. Then, the emission probabilities can be summarized by (considering the state j) :

$$b_j^{MSD-HMM}(f_0, F_M \mid \lambda)$$
$$b_j^{GMM}(spectre, phase, énergie \mid \lambda)$$

These 2 emission probabilities are considered as independent, so their product will constitute the output emission probability of the state j.

VIII. Conclusion

This project was devoted to the development of a software *Harmonic Plus Noise* (*HNM*) for the acoustic unit concatenative speech synthesis. It aims on the one hand the development of a new diphone-based voice based on the Festival speech engine and on the other hand, the improvement of the Hidden Markov Model-based speech synthesis that allows introducing emotions and personal features in the synthetic speech.

The new model *Harmonic Plus Noise* groups together some advantages conferring on it a superiority compared with previous methods, and enabling us to produce a very high quality of artificial speech. We refer to it as a hybrid model since it decomposes speech frames into a harmonic part and a noise part. This decomposition allows more natural sounds by ensuring the simultaneous conservation of the low and high frequency energies. Moreover, the modelling of the excitation and the vocal tract are here coupled in a unique system, in contrary to filter-based models. At last the parametric structure of the model permits us to easily carrying out prosodic modifications on the signal, smoothing of discontinuities at the concatenation points ; it also allows a speech coding. The main drawback of this method remains however its significant computation load tied to its complexity.

The programming of this model was based on the natural language processing modules of the *Festival* speech engine. It separates an Analysis part which parametrizes the speech and a Synthesis part which brings the prosodic modifications and the smoothing of the discontinuities, before the generation of the speech waveform. Informal tests have validated the different steps of the development of this model : the high quality re-synthesis, the diphones segmentation, the prosodic modifications and time-scales modifications and lastly the smoothing of the discontinuities are justified and prove their efficiency in the obtained results. These tests have also shown the naturalness gain of this synthesis compared with the RELP one. An improvement at this stage remains to develop : the integration of the noise part into the harmonic one after the pitch modifications introduces a "resonance" which decreases the real quality of the voice. The correction of this defect should also allow more intelligibility and pleasantness in the synthetic voice.

At the end the development of this model, its introduction in the Hidden Markov Model-based synthesis may constitute a future research work in the IDIAP institute. It seems indeed that the recognition and synthesis speech models tend today towards convergence, and IDIAP, specialized in speech recognition, tries consequently to develop its research in the second domain. The speech synthesizers allows today to realize only a neutral voice, but of high quality. The HMM-based synthesis can remove this neutral feature by particularizing the voice, but remains of average quality since it is based on filter. The *Harmonic Plus Noise* synthesis could consequently constitute a interesting advance by combining the high natural voice with the personalization of the speaker introduced by the HMMs. Some methods for the integration of the HNM waveform synthesis into the HMM-based speech synthesis were developed at the end of this report. They will constitute, I hope, an interesting starting point in the elaboration of the HMM-based synthesis.

Acknowledgment

First I would like thank my parents who have paid all my 5 years of study. They would be happy the outcome in this way.

I thank Prof. Hervé Bourlard who had the kindness of accepting me in his institute. Beyond a technical background, I found a rewarding cultural experience and an additional step towards the mobility way.

I am thankful to Prof Thierry Dutoit, not only for his help in the speech synthesis domain, but also for giving me the opportunity to realize this diploma work abroad.

My local supervisors, Jithendra Vepa and John Dines, take also an important place in the realization of this work since they have kept me up during all the stage and had the patience to bring me their help and advice with constancy.

Finally, I thank Carine Labie, language professor, who helped me in the translation of this work.

Bibliography :

[1] Y. Stylianou, Thèse de Doctorat "Harmonics Plus Noise Models for Speech Combined with statistical methods, for speech and speaker modification", Ph.D. diss.,Ecole Nationale Supèrieure des Télécommunications, Paris, France, Jan.1996.

[2] Y. Stylianou, "Applying the harmonics plus noise model in concatenative speech synthesis", IEEE, TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 9, NO. 1, JANUARY 2001

[3] Y. Stylianou, "A Pitch and Maximum Voiced Frequency Estimation Technique adapted to Harmonic Models of Speech", IEEE Nordic Signal Processing Symposium, Helsinki, Finland, Sept. 1996.

[4] Y. Stylianou, "Removing phase mismatches in concatenative speech synthesis", IEEE, in Proc. 3rd ESCA Speech Synthesis Workshop, Nov. 1998, pp. 267–272.

[5] Y. Stylianou, "On the implementation of the harmonic plus noise model for concatenative speech synthesis", IEEE, International Conference on Acoustics, Speech and Signal Processing 2000, Istanbul, Turkey.

[6] Y. Stylianou, T. Dutoit, and J. Schroeter, “Diphone concatenation using a harmonic plus noise model of speech,” in *Proc. Eurospeech*, 1997, pp.613–616.

[7] Y. Stylianou, J. Laroche, and E. Moulines, "High-Quality Speech Modication based on a Harmonic +Noise Model", Proc. EUROSPEECH, 1995.

[8] Th. Dutoit and H. Leich, \Text-To-Speech synthesis based on a MBE re-synthesis of the segments database," Speech Communication, vol. 13, pp. 435{440, 1993.

[9] Th. Dutoit, “High quality tex-to-speech synthesis : a comparison of four candidate algorithms”, Proceeding of IEEE, ICASSP 94, Adélaïde, pp. I. 565-568.

[10] Th. Dutoit, "An Introduction To Text-To-Speech Synthesis", Kluwer Academic Publishers, 1997

[11] Th. Dutoit, "Introduction au traitement de la parole", Faculté Polytechnique de Mons, notes de cours

[12] Th. Dutoit, "Introduction au traitement de la parole - Compléments", Faculté Polytechnique de Mons, notes de cours

[13] Th. Dutoit, "Traitement du Signal", Faculté Polytechnique de Mons, notes de cours

[14] Th. E. Quatieri, "Discrete-Time Speech Signal Processing", Prentice Hall Signal Processing Series, 2001

[15] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, ``Speech parameter generation algorithms for HMM-based speech synthesis," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, vol.3, pp.1315-1318, June 2000.

[16] Keiichi Tokuda, Takashi Mausko, Noboru Miyazaki, Takao Kobayashi, ``Multi-space probability distribution HMM," IEICE Trans. Information and Systems, vol.E85-D, no.3, pp.455-464, Mar. 2002.

[17] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi and Tadashi Kitamura, ``Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," Proceedings of European Conference on Speech Communication and Technology, Budapest, Hungary, vol.5, pp.2347-2350, Sep. 1999.

[18] Keiichi Tokuda, Heiga Zen, Alan W. Black, ``An HMM-based speech synthesis system applied to English," 2002 IEEE Speech Synthesis Workshop, Santa Monica, California, Sep. 11-13, 2002.

[19] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, ``Mixed excitation for HMM-based speech Synthesis," Proceedings of European Conference on Speech Communication and Technology, vol.3, pp.2259-2262, Aalborg, Denmark, Sep. 3-7, 2001.

[20] B. Gosselin, “Classification et Reconnaissance Statistique de Formes”, Faculté Polytechnique de Mons, notes de cours.

[21] A. W. Black and P. Taylor. The Festival Speech Synthesis System: system documentation. Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Scotland, UK, 1997. Avaliable at http://www.cstr.ed.ac.uk/projects/festival.html.

[22] A. W. Black , P. Taylor and R. Caley, ”The Architecture of The Festival Speech System”, In *The Third ESCA Workshop in Speech Synthesis*, pages 147-151, Jenolan Caves, Australia, 1998.

[23] J. Dines, PhD Thesis, "Model Based Trainable Speech Synthesis and its applications", Queensland university of Technology, Speech Research Laboratory, Australia, 2003.