Chapter 3.

The Digital Signal Processing module.

The two-blocks description introduced at the beginning of section 1.4 and the definition of a rigid but general interface between them presents the advantage of allowing a separate study of both processes, whether it was for their requirements, functionalities or results. Consequently, we shall now assume that top-quality information is delivered at the inputs of the Digital Signal Processing module, like that which would be directly extracted from a human reading.

Intuitively, the operations involved here are the computer analogue of dynamically controlling the articulatory muscles and the vibratory frequency of the vocal folds so that the output signal matches the input requirements. In order to do it properly, the DSP module should obviously, in some way, take articulatory constraints into account[1], since it has been known for a long time [Libermann 59] that phonetic transitions are more important than stable states for the understanding of speech. This, in turn, can be basically achieved in two ways :

- Explicitly, in the form of a series of rules which formally describe the influence of phonemes on one another;

[1]Even if, as already mentioned, the actual synthesis technique describes speech in terms of time-varying parameters that generally have no close relationship with articulatory ones.

- Implicitly, by storing examples of phonetic transitions and co-articulations in a speech segment database, and using them just as they are, as ultimate acoustic units.

Two main classes of TTS systems have emerged from this alternative, which quickly turned into synthesis philosophies given the divergences they present in their means and objectives : *rule synthesizers* and *segments concatenation synthesizers*[2].

3.1. Rule Synthesizers.

Rule synthesizers are mostly in favour with phoneticians and phonologists, for they constitute a cognitive, generative approach of the phonation mechanism [Pols 90]. The broad spreading of the Klatt synthesizer [Klatt 80], for instance, is principally due to its ability to study the characteristics of natural speech, by analytic listening of rule-synthesized speech [Klatt & Klatt 90]. What is more, the existence of relationships between articulatory parameters and the inputs of the Klatt model make it a practical tool for investigating physiological constraints [Stevens 90].

Rule based synthesizers are organized as in figure 3.1. In a preliminary phase, a large amount of words, generally Consonant-Vowel-Consonant (CVC) sequences, are read by a professional speaker, numerically recorded and stored. They are chosen so as to constitute a speech corpus, representative of the many transitions and co-articulations to study. Thanks to a speech analyser, the resulting digital data are then given a parametric form which typically separates the respective contributions of the glottal folds and of the vocal tract, and presents the latter in a compact way, more suitable for further study. A rule finding stage is then applied on source, which is generally performed by human experts. A first inspection of the speech data from all the speakers is done to get the rules form. The actual values of their parameters are then adjusted for one speaker, since rough inter-speaker means have little significance at this level[3]. Rules from already existing synthesizers, i.e. extracted from other speakers' voices and by different persons, may advantageously help as far as they model general articulatory features rather than speaker peculiarities[4] and as they do not correct system specific deficiencies. Finally, a long trial and error analysis work is undertaken to optimize the synthetic quality.

[2]not to mention intermediate approaches, like the one of [Bimbot et al 89], which we shall not develop here.

[3]The existence of such an inter-speaker inconsistency is also encountered at the prosody generation level.

[4]An important feature of rule-based synthesizers is that they allow to study speaker-dependant voice features so that switching from one synthetic voice into another can be achieved with the help of specialized rules in the rule database. Following the same idea, synthesis by rule seems to be a natural way of handling the articulatory aspects of changes in speaking styles (as opposed to their prosodic couterpart, which can be accounted for by concatenation synthesizers as well).

When a sufficient number of rules is gathered, synthesis can start. Rules are matched to the phonetic inputs, and a parametric speech signal is produced. It is finally transformed back into digital speech by a synthesizer which implements the model chosen in the analysis stage.

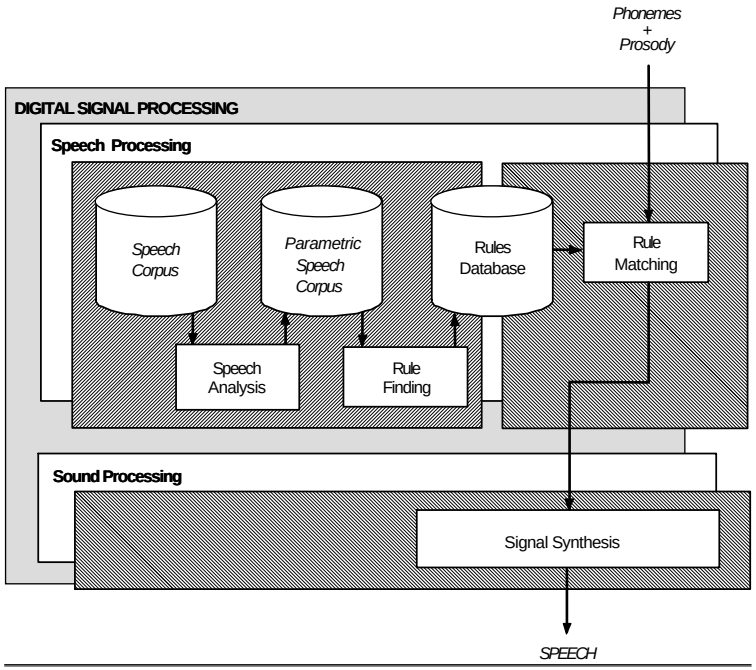

Figure 3.1 A typical Rule Synthesizer. Most of the work is effectively concentrated in the speech processing field.

Consequently, the final **segmental quality** of the synthetic speech delivered by a rule synthesizer depends :

- on its internal rules efficiency, i.e. their capacity to describe the basic parametric corpus with few audible errors;
- on the quality of the corpus, that is :
  - on the actual choice and recording quality of the utterances stored in it;
  - on the accuracy of the speech model used in the analysis block to effectively describe high quality speech : even if no rule-finding / rule-matching were applied, synthetic utterances could differ from the original ones due to *intrinsic* modelization errors;
  - on the actual algorithm that is used to fix up the (time-varying) values of the given model parameters, which can be responsible for *extrinsic* modelization errors, denoted as such because they result from the analysis algorithm inability to produce a correct parametric representation rather than from an incapacity of the model itself, even though this effect is often related to the peculiar model chosen;
- on the improvements gained by the trial and error analysis stage, which can eventually compensate for recording or algorithmic shortage.

For historical and practical reasons (mainly the need for a physical interpretability of the model), rule-based synthesizers always appear in the form of *formant synthesizers*. These describe speech as the dynamic evolution of up to 60 parameters [Stevens 90], mostly related to formant and anti-formants frequencies and bandwidths together with glottal waveforms[5]. As a result, they are almost free from intrinsic modelization errors. On the opposite, the large number of (coupled) parameters complicates the analysis stage. What is more, formants frequencies and bandwidth are inherently difficult to estimate from speech data. The need for intensive trials and errors in order to cope with extrinsic errors, makes them time-consuming systems to develop (several years are commonplace). Yet, the synthesis quality achieved up to now reveals typical buzzyness problems, which originate from the rules themselves : introducing a high degree of naturalness is theoretically possible, but the rules to do so are still to be discovered[6].

Rule synthesizers remain, however the most potentially powerful approach to speech synthesis. No wonder then that they have been widely integrated into TTS systems (MITTALK [Allen et al 87] and the JSRU synthesizer [Holmes et al 64] for English, [Santos & Nombela 82] for Spanish, the multilingual INFOVOX system [Carlson et al 82], and the I.N.R.S system [O'Shaughnessy 84] or [Bailly 88] for French).

---

[5]Since rule based synthesizers are not the object of our work, we invite interested readers to refer to [Holmes 83] and [Allen 87] for deeper descriptions of formant synthesizers.

[6]Naturalness problems are not peculiar to rule synthesizers. Even with segments concatenation methods, increasing the naturalness of a given synthesizer is far from being simple. Ideas such as incorporating stochastic components into speech models or rules most often merely result in additional noises...

3.2. Concatenation Synthesizers.

As opposed to rule-based ones, *concatenation synthesizers* possess a very limited knowledge of the data they handle : most of it is embedded in the segments to be chained up. This clearly appears in figure 3.2 : all the operations that could indifferently be used in the context of a music synthesizer (i.e. without any explicit reference to the inner nature of the sounds to be processed) were grouped into a *sound processing* box.

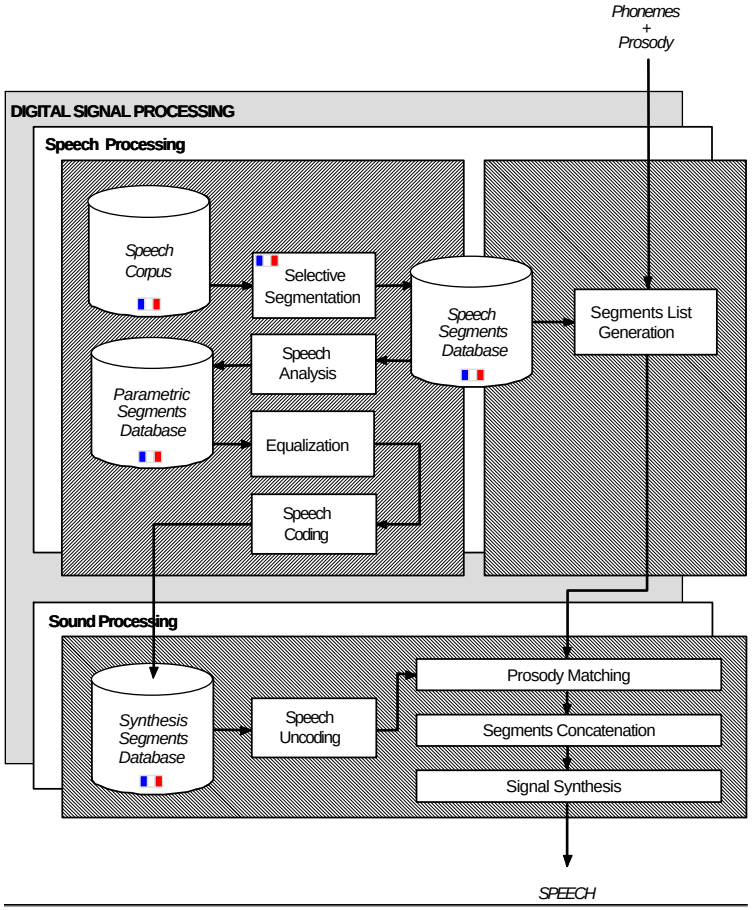

Figure 3.2 A general Concatenation Synthesizer TTS system. Language-dependent operations and databases are indicated by a flag.

3.2.1. Database preparation.

As before, a series of preliminary stages have to be fulfilled before the synthesizer can produce its first utterance. At first, segments are chosen so as to minimize future concatenation problems. A combination of diphones (i.e. units that begin in the middle of the stable state of a phone and end in the middle of the following one[7]), half-syllables, and triphones (which differ from diphones in that they include a complete central phone) are chosen as speech units, since they involve most of the transitions and co-articulations while requiring an affordable amount of memory. When a complete list of segments has emerged, a corresponding list of words is carefully completed, in such a way that each segment appears at least once (twice is better, for security). Unfavourable positions, like inside stressed syllables or in over co-articulated contexts, are excluded. A corpus is then digitally recorded and stored, and the elected segments are spotted, either manually with the help of signal visualization tools, or semi-automatically thanks to segmentation algorithms, the decisions of which are checked and corrected interactively. A segments database centralizes the results, in the form of the segments names, durations, and internal sub-splittings.

Segments are often given a parametric form, collected at the output of a speech analyser and stored in a parametric segments database. This operation recalls in many ways the analysis performed in rule synthesizers, but its objective is fairly different. Clearly, increasing the 'readability' of the data becomes useless : no more human interaction is expected. The use of a speech model, however, is often maintained[8] for two main reasons :

- Well chosen models allow data size reduction, an advantage which is hardly negligible in the context of concatenation synthesizers, as opposed to rule-based ones, given the amount of data to be stored here. Consequently, the analyser is often followed by a parametric speech coder.

- A number of models explicitly separate the contributions of respectively the source and the vocal tract, an operation which remains helpful for the pre-synthesis operations : prosody matching and segments concatenation.

---

[7]A consequence of this very imprecise definition being that diphones mostly remain obscure units when highly transient sounds are involved. This point is further discussed in Chapter seven.

[8]Even though it is generally not a formant model, as in rule synthesis.

Indeed, the actual task of the synthesizer is to produce, in real-time, an adequate sequence of concatenated segments, extracted from its parametric segments database and the prosody of which has been adjusted from their current value, i.e. the intonation and the duration they appeared with in the original speech corpus, to the one imposed by the Language Processing module. Consequently, the respective parts played by the Prosody Matching and Segments Concatenation modules are considerably alleviated when input segments are presented in a form that allows easy modification of their pitch, duration, and spectral envelope, as is hardly the case with crude PCM samples.

Since segments to be chained up have generally been extracted from different words, that is in different phonetic contexts, they often present amplitude and timbre mismatches. Even in the case of stationary vocalic sounds, a rough sequencing of parameters typically leads to audible discontinuities. These can be coped with during the constitution of the synthesis segments database, thanks to an *equalization* in which related endings of segments are imposed similar amplitude spectra, the difference being distributed on their neighbourhood. This operation, however, is generally restricted to amplitude parameters. Timbre conflicts are better tackled at run-time, by *smoothing* individual couples of segments when necessary rather than equalizing them once for all, so that some of the phonetic variability naturally introduced by co-articulation is maintained. In practice, amplitude equalization can be performed either before or after speech analysis (i.e. on PCM samples or on speech parameters).

Once the parametric segments database has been completed, synthesis can begin.

3.2.2. Speech Synthesis.

A sequence of segments is first deduced from the phonetic input of the synthesizer, in a Segments List Generation block which interfaces NLP and DSP modules. It queries the segments database for global information on the units it contains. Segments durations are chosen accordingly, given the desired lengths of the embedded phonemes, boundary ones being distributed proportionally to the initial respective contributions of the segments concerned. As an example, let us suppose the word 'le' is to be synthesized. After some language processing, the command '_ #120 l #70 @ #150 _ #100'[9], in which '#*nnn*' stands for 'with a duration of *nnn* ms', is presented at the input of the synthesizer. Successive accesses to the segments database inform the Segments List Generator that the following units are available :

[9]For the sake of clarity, prosodic events were not introduced here. They are positioned from phonemes to synthesis segments in a similar manner.

Segment names	subsegments durations	
_l	#100	#50
l@	#30	#90
@_	#90	#200

Clearly, durations have to be adapted. Assuming the *a priori* equal importance of the left and right parts[10] of any diphone, a constant shortening (or lengthening) ratio is applied throughout a given unit, which results in the final synthesis command : '_l #120 #44 l@ #26 #75 @_ #75 #100', where '#*nnn*' now denotes desired subsegments durations.

Once prosodic events have been correctly assigned to individual segments, the prosody matching module queries the synthesis segments database for the actual parameters, adequately uncoded, of the elementary sounds to be used, and adapts them one by one to the required prosody. A convenient way to do this is to make use of two *prosody transformation functions* which completely describe the prosodic changes to be made. The first one is a monotonous time alignment function $t'(t)$ which associates a synthesis instant t' to any analysis instant t. It is easily computed by linearly interpolating the (t,t') points corresponding to the limits of the subsegments of the current segment (Fig. 3.3), as found in the segments database (t) and in the synthesis command string (t'). In a like manner, the second function $\omega_0(t')$ provides frequency values at any synthesis instant t'. The way it is computed actually depends on the intonation description model implemented in the prosody generator module, as mentioned in section 2.2.1. In all our experiments, we have followed the IPO approach, for which ω_0 is described as a piecewise linear function.

[10]It is preferable to speak in terms of 'parts', rather than 'halves', since respective contributions are obviously not equal (see Section 7.1.3).

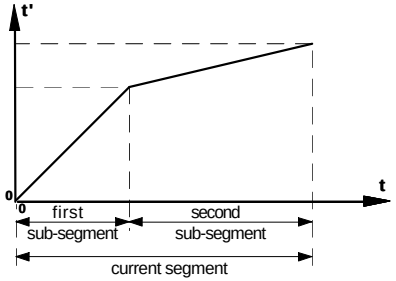

Fig. 3.3 Example of time alignment function for a segment with two subsegments (e.g. a diphone).

The segments concatenation block is in charge of dynamically matching segments to one another, by smoothing discontinuities. Here again, an adequate modelization of speech is highly profitable, provided simple interpolation schemes performed on its parameters approximately correspond to smooth acoustical transitions between sounds. Since such modifications are most often relevant to both ends of a segment, the concatenation block is inherently non-causal, as shown in figure 3.4, in which a one segment delay[11] has been introduced in order to force causality.

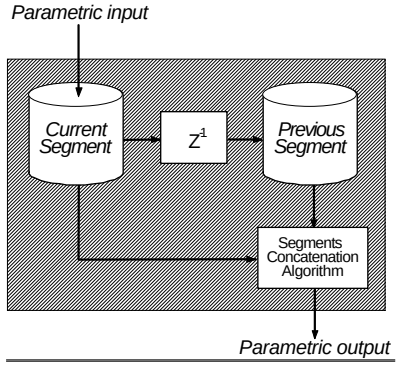

Figure 3.4 Inside the segments concatenation block.

[11]It should be emphasized that, as opposed to rule-driven transitions, the matchings performed by the concatenation block are independent of the type of segments processed. Segments are considered as sounds rather than speech units. Consequently, there is no reason, nor even means, to adapt a given segment according to any of its non-nearest neighbours.

Given the natural acoustical 'proximity' of the sounds to be chained, a simple linear smoothing is often applied, as shown on figure 3.5. Let us denote the left and right segments to be concatenated by L and R. If we consider a set $\mathbf{p}$ of parameters $\{p_1, p_2, ..., p_N\}$, the values of which are $\mathbf{p_L}^0$ at the end of L and $\mathbf{p_R}^0$ at the beginning of R, linear smoothing consists of distributing the difference $(\mathbf{p_R}^0-\mathbf{p_L}^0)$ amongst a number M_L of vectors $\{\mathbf{p_L}^{-(M_L-1)}, \mathbf{p_L}^{-1}, \mathbf{p_L}^0\}$ before and including $\mathbf{p_L}^0$, and a number M_R of vectors $\{\mathbf{p_R}^0, \mathbf{p_R}^1, ..., \mathbf{p_R}^{(M_R-1)}\}$ after and including $\mathbf{p_R}^0$. The values M_L and M_R are different, in all generality, since they can be imposed by the segments themselves, in order to restrain the interpolation process to their steady state parts only. If we denote as $\mathbf{p'}$ the parameters after smoothing, the interpolation laws are given by :

$$p'^{-i}_L = p^0_L + (p^0_R - p^0_L)\frac{i}{2M_L} \qquad (3.1)$$

$$p'^{j}_R = p^0_R + (p^0_L - p^0_R)\frac{(M_R - j)}{2M_R} \qquad (3.2)$$

for i=0...M_L-1 and j=0...M_R-1

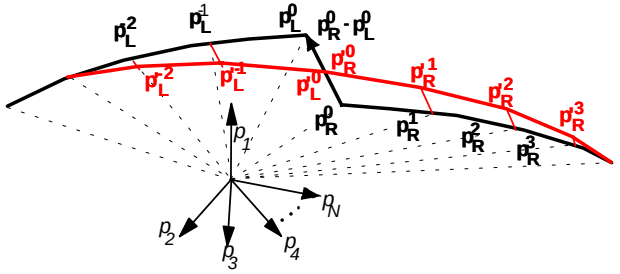

Figure 3.5 Parametric linear smoothing at the border of successive segments (M_L=2; M_R=3; $p_1 p_2 ... p_N$ are the axes of the multi-dimensional parametric space).

The resulting stream of parameters is finally presented at the input of a synthesis block, the exact counterpart of the analysis one.

3.2.3. Segmental quality.

Figure 3.2 naturally introduces a description of concatenation synthesizers in terms of segmental quality. Apart from the sampling frequency chosen (telephone quality voices require 8 kHz, while 10 kHz is a good choice for medium-quality, general purpose

systems, the difference with 8 kHz being quite noticeable, and 16 kHz or higher is a must for high-quality speech synthesizers), their efficiency to produce high quality speech is subordinated to :

- the type of segments chosen;
- the corpus they were extracted from;
- the segmentation quality;
- the speech signal model, to which the analysis and synthesis algorithms refer;
- the amount of degradation introduced by the speech coding phase;
- the prosody matching efficiency, which is strongly related to the model;
- the capabilities of the concatenation algorithm;

They are now reviewed.

3.2.3.1. The segments choice.

Segments should obviously exhibit some basic properties. Given the restricted smoothing capabilities of the concatenation block, they should be easily connectable, while keeping their number as low as possible[12]. On the other hand, longer units decrease the density of concatenation points, therefore providing better speech quality. This is clearly in contradiction with the limited memory constraint.

Diphones are not too numerous (about 1200 for French, including lots of phoneme sequences that are only encountered at word boundaries, for a little bit less than 3 minutes of speech, i.e. approximately 5.4 Mbytes of 16 bits PCM samples at 16 kHz) and they do incorporate most phonetic transitions. No wonder then that they have been extensively used. They imply, however, a high density of concatenation points (one per phoneme), which reinforces the importance of an efficient concatenation algorithm. Besides, they can only partially account for the many co-articulation effects of a language, since these often affect a whole phone rather than just its right or left halves independently : formant targets themselves tend to be changed, or a formant starts rising (or falling) during the realization of one phoneme and continues that movement smoothly throughout the ensuing one without any steady state (for a deeper insight, see the excellent [O'Shaughnessy 90], in which contextual effects are presented in terms of formant movements). Such effects are especially patent when somewhat transient phones, such as liquids and (worst of all) semi-vowels, are to be connected to one another.

[12] *Stricto sensu*,the information segments provide is more important, as far as storage is concerned, than their actual number, since quantization techniques can be applied on the parametric segments database, that tend to approach information theoretic limits. For small databases of equal size units, however, information is proportionnal to the number of segments.

Assimilations[13] are still a bigger source of concern. In French, sonorant consonants [R,l,m,n,j,H,w] become partially or totally unvoiced when they are immediately preceded or followed by a non-sonorant consonant (like in 'carte', 'plein', etc...). Partial assimilation cases are approximately covered by diphones, while total ones would require the introduction of several realizations (and a corresponding number of diphones) per phoneme in the database. In the case of sonorant consonants for example, this would necessitate the recording, segmentation, and storage of voiced and unvoiced versions of each phone (Fig. 3.6). This is sometimes denoted as *allo-diphones synthesis.*

Di-syllables (i.e. units that would start and end in the middle of vowels) could precisely obviate these difficulties, for only vowels would be concatenated[14]. Their number, however, is highly problematic and their size is naturally larger than diphones (more than 100000 theoretically possible units for French : about 300 minutes of speech and 600 Mbytes). Yet it is not clear whether restricting concatenation points to low energy sounds (i.e. consonants) would not precisely lead to a higher quality, in which case syllables (i.e. units of the form *Ci - V - Cf*, where *Ci* and *Cf* are consonant clusters and *V* is one vowel) would prevail. For the time being, neither solution has been adopted as such. It makes more sense to use them in combination with diphones.

Besides, triphones constitute a variant to the use of syllabic units as a complement to diphones. They can embody the strong dynamic effects quoted above, providing these only affect one phone, i.e. excluding sequences of sonorant consonants. To a larger extent, associating diphones with tri- and tetraphones to cope with a limited number of very specific contextual effects is often denoted as a *polyphone* approach [Aubergé 91].

[13] At least the ones that do not simply change a phoneme into something that is sufficiently close to another one for a substitution to be acceptable.

[14] Syllable protagonists also often argue that children start to babble syllables before they actually begin to talk. The hypothetical 'syllable-timed' nature of languages is still largely debated [Wenk & Wioland 82].

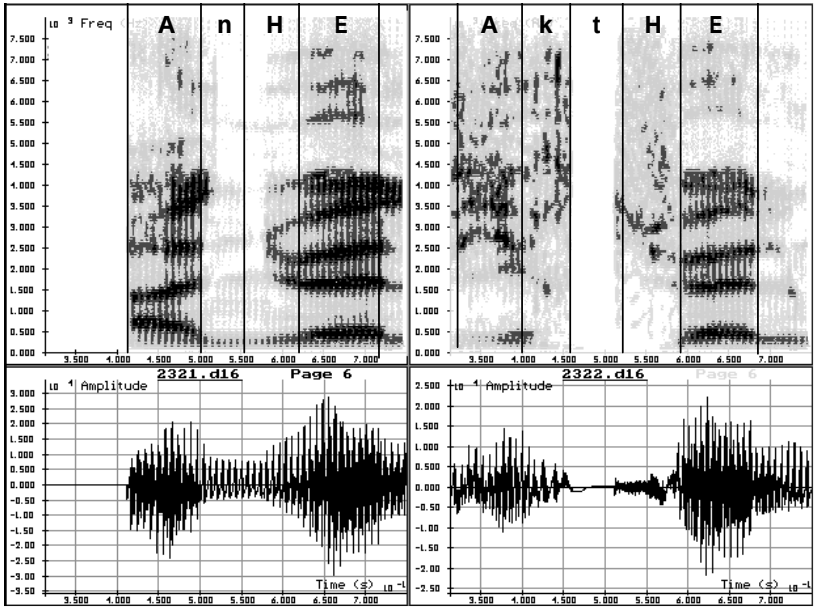

Figure 3.6 A case of sonority assimilation. On the left, the beginning of the word *annuellement*, in which [H] is placed in a voiced context. On the right, the beginning of *actuellement* : [H] is totally de-voiced, due to the preceding unvoiced plosive.

Another way of keeping the database size small while still somewhat turning syllables to account is to split them in two, at their vocalic nucleus, which acts as a co-articulation barrier in most cases. Such demisyllables (of the form *Ci - V* or *V - Cf*) have been fully adopted for the synthesis of German, in which long consonant clusters are manifold. A complete demisyllables database, containing about 2000 units, plus some high frequency functional words in order to decrease the concatenation density, is described in [Kraft & Andrews 92], while a combination of HAlf-syllables, DIphones, and sufFIXes provides the basis of the HADIFIX system [Portele et al 90].

Finally, even though segments should maintain a close relationship with phonemes, so as the segments list generator remains unmistakable, nothing truly impedes them from being directly chosen on the basis of connectivity requirements rather than on individual researchers' phonological knowledge. [Nakajima & Hamada 88] for

example, report the automatic extraction of 627 non-uniform, or hybrid synthesis units from a corpus of 432 words. It is further developed in [Takeda et al 89]. Even though no exceptional quality was reported, such an approach could speed up the constitution of the segment databases, for multilingual synthesis for instance.

To come to a conclusion, finding an optimal set of segments for a given memory capacity is still an open problem, mostly due to the impossibility of associating scores to sets of units, so as to compare them. Even if quantitative intelligibity measurements (such as the ones obtained by the Diagnostic Rhyme Test (DRT) or the Consonant-Vowel-Consonant (CVC) test : see Chapter seven) were collected for similar synthesizers addressing different segments databases for a given language, contrasting the results would still be highly debatable, since segmental quality has also a lot to do with the peculiar speech corpus from which segments originate.

3.2.3.2. The corpus.

Acoustic units are generally extracted from a specialized, tailored corpus, for their relative occurrence frequency in natural speech is highly variable. In the case of diphones, for instance, a list of 100 phonetically balanced sentences only covers 43% of the 1200 units required, with a redundancy of about 80 % [Tubach & Boë 86]. Moreover, as emphasized above, the shorter the segments (whether it was explicitly due to the type of units chosen, or implicitly, given the speaking rate adopted when reading), the more their realizations depend on their phonological context. The design and recording of the corpus should therefore be given special attention.

Just think for instance of the synthesis of [iSô] with diphones |iS| and |Sô|, respectively extracted from *dénicher* and *baluchon*. Clearly, both acoustic realizations of [S] are very different from one another, given their respective vocalic contexts, so that |iS| and |Sô| will generally not match exactly (this is illustrated in figure 3.7). As far as possible, such co-articulation effects should be avoided during the design of the synthesis corpus. At least, they should be kept small enough for the segments concatenation block in figure 3.2 to be able to smooth the resulting discontinuities.

Yet it is not clear whether segments should be extracted from nonsense syllabic sequences (also called *logatoms*), existing isolated words, or meaningful sentences. Even the question of best positioning units in the corpus is still widely debated. Stressed syllables are longer, thus less submitted to co-articulation, which results in easily chainable units; while unstressed ones are more numerous in natural speech, so that producing them efficiently could both increase segmental quality and reduce memory requirements. Likewise, co-articulations (and to a lesser extent assimilations) are also strongly subordinate to speaker 's fluency, so that imposing a slow speaking rate results in more intelligible but highly over-articulated units.

To a larger extent, the issues addressed above are part of a regrettable but necessary tradeoff between intelligibility and naturalness.

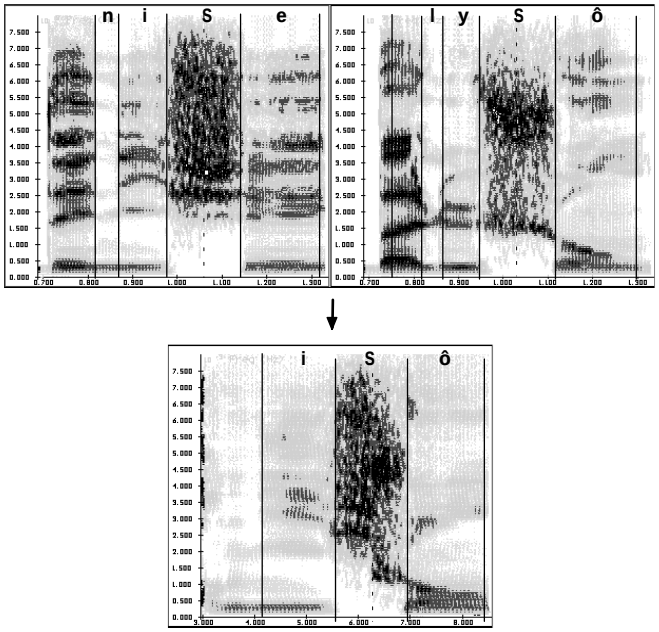

Figure 3.7 Co-articulation affects the realization of the [S] in [iSe] and [ySô]. As a result, a spectral discontinuity appears when synthesizing [iSô].

3.2.3.3. Segmentation.

Except for approaches like [Nakajima & Hamada 88], which explicitly banishes human intervention in the segments generation process, a fully automatic segmentation of corpuses is hardly conceivable in the context of concatenation synthesis. Indeed, most units presented in past discussions originate in phonological considerations rather than acoustic grounds. Isolating them therefore requires a deep prior knowledge of their specific features, so that unsupervised segmentation (i.e. segmentation on acoustic principles only, which [Van Hemert 91] calls *implicit segmentation*) can merely aspire to outline a solution (segments and sub-segments limits are misplaced, or just missing, while undefined ones appear), the refinement of which has to be performed by human experts.

Although going one step further, by enabling the computer to get additional information about the problem it is supposed to solve, is of a high interest to speed up the process, it is not likely to suppress manual corrections, at least for obtaining the highest quality achievable with a given corpus (see the excellent [Boeffard et al 92]). As a matter of fact, whatever means adopted (incorporating *a priori* information by designing an expert system, or allowing automatic learning with statistical or neural models), the final system should clearly exhibit a speaker-independent behaviour, since different corpuses are generally read by different people[15], so that the many speaker peculiarities will always introduce segmentation inconsistencies.

3.2.3.4. The model.

As for rule synthesizers, describing speech in terms of mathematical equations, the effect of which is revealed by the synthesis algorithm, generally introduces *intrinsic* modelization errors. Yet, *extrinsic* ones (i.e. analysis biases) can also occur, as the analysis algorithm does not always deliver the best achievable parameters, that is to say the parameters that would allow the synthesizer (thus the model) to produce its best quality speech.

Both modelization errors still have a major contribution in the somewhat limited naturalness provided by most actual TTS systems. They can be isolated by combining the analysis and synthesis algorithms alone into a *copy synthesizer* (fig 3.8), the aim of which is to generate the synthetic signal that best mimics the natural speech presented at its input. They can even be somewhat separated from one another by presenting some synthetic speech that was precisely produced by the given model at the input of the copy synthesizer, and contrasting its input and output signals : extrinsic errors are responsible for the difference.

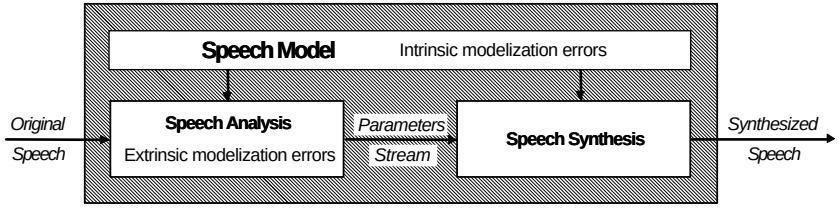

Figure 3.8 Copy Synthesis.

[15]Automatic segmentation is most interesting in the context of multi-lingual TTS systems, in which case corpuses are read by natives.

Models can be roughly divided into two classes.

Physiological ones are the foundation of *articulatory synthesis* (from the early two mass vocal folds model of [Ishizaka & Flanagan 72] to the recent region and mode theory of [Mrayati et al 88], through the very popular Maeda 's model [Maeda 79]). Given the high complexity of the (non-linear partial derivatives) mathematical equations underlying these models, both the analysis and synthesis operations remain open problems. Assessing articulatory parameters from rough speech data, often denoted as the *acoustic-articulatory inversion*, can no more be performed by some simple signal processing algorithms. Neural networks, for instance, have recently been proposed as a black-box type solution [Soquet et al. 90]. As a result, the interest of these models in the context of TTS synthesis is somewhat reduced. So far, they have mostly been used for cognitive research purposes.

As opposed to articulatory models, *terminal-analogue* ones originate, as their name literally infers, in the fact that the actual segmental quality of a synthesizer is related to its efficiency to produce natural sounding speech, no matter what the means to do it. They are again roughly classified into two groups depending on their relationship with the actual phonation process. *Production models* provide mathematical substitutes for the part respectively played by vocal folds, nasal and vocal tracts, and by the lips radiation[16]. Their most representative members are Linear Prediction Coding (LPC) synthesizers, and the formant synthesizers we mentioned in section 3.1. On the contrary, *phenomenological models* intentionally discard any reference to the human production mechanism. Among these pure digital signal processing objects, spectral and time-domain approaches are increasingly encountered in TTS systems.

3.2.3.5. The parametric speech coder.

It is well known by speech processing adepts that the strong inertia of the articulatory muscles is responsible for a large amount of redundancy in speech signals[17]. Data reduction techniques try to turn it into account. However, whatever the form in which speech is passed to a speech coder, whether it was as a sequence of samples or as a list of parameters referring to a given model, respectively addressed by *waveform* and *parametric coding algorithms*, compression brings distortion. The maximum compression ratio for which distortions remain inaudible, which is generally obtained by optimizing the given speech coder in the corresponding *vocoder* (Fig. 3.9), is an important figure in TTS synthesis. If it is too low, a tradeoff becomes necessary between quality and economical considerations.

[16]So that they could be called sub-components terminal-analog models.

[17]Yet only signal level redundancy is taken into account here. Additional superfluity is encountered in all languages at the linguistic level, as a means naturally adopted by humans to protect their message against misunderstandings, which are some form of channel distorsion.

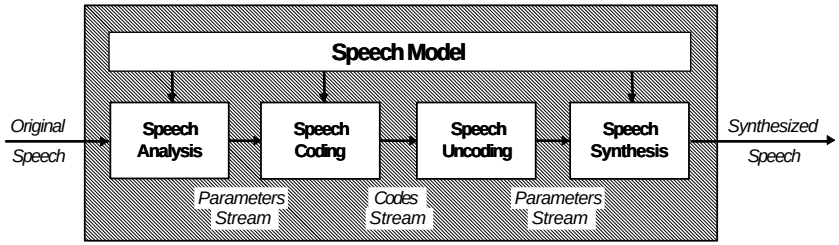

Figure 3.9 A general vocoder.

3.2.3.6. Prosody matching.

Being able to change the pitch and/or duration of segments while maintaining naturalness is of primary importance in TTS synthesis. It definitely is the main criterion in the choice of a model. Given the natural range of the laryngeal frequency in reading (yet with no emphasis), and assuming segments were recorded with an average frequency lying in the middle of this range, the prosody matching module should allow to apply a factor of 0.5 to 2 on the segments pitch, without affecting the position and bandwidth of formants[18,19]. Similarly, in order to account for the different lengths of stressed and unstressed vowels, a contraction/dilation of 0.5 to 2 should be possible. Most of the aforementioned model do it quite well.

3.2.3.7. Concatenation.

Transitions between speech sounds are theoretically defined in the articulatory domain. Whatever the speech model chosen, they are in all generality expressed in terms of non-linear relations (even in the case of formants description : see fig 3.9). They are consequently best accounted for by rule (as in the synthesis-by-rule approach), statistical, or neural models. However, when sounds to be concatenated are sufficiently 'close' to one another, linear relations give good results, provided they are performed in an adequately chosen representation space. Or, similarly, the efficiency of linear laws to describe simple transitions in such a way that they are still perceived as natural,

[18]As opposed to the effect obtained by varying the pitch of a cassette recorder, which results in well known bull-dog or mouse voices, since the whole spectrum is contracted/dilated.

[19]It should be noticed, however, that given the physical connection between the glottis and the laryngeal tract (the latter playing the role of a finite impedance on the former), pitch movements are somewhat related to formants variations. Consequently, wide range pitch changes result, with most speech models, in rather unnatural voices.

depends on the particular representation space they are expressed in[20] (see fig 4.10 in Chapter four for a striking example).

The importance of the concatenation operation for synthesis-by-concatenation has been previously underlined in the context of the natural mismatches between segments in a given database. Moreover, the availability of an efficient concatenation algorithm actually interferes with the database preparation, in the sense that over-articulation can be avoided during the recording phase, which results in faster and more natural synthetic speech.

References

[Aubergé 91] V. AUBERGE, *La synthèse de la parole : des règles aux lexiques*, Ph.D. thesis, ICP Grenoble, 1991, pp. 185-204.

[Allen et al 87] J. ALLEN, S. HUNNICUT, D. KLATT, *From Text To Speech, The MITTALK System*, Cambridge University Press, 1987, 213 pp.

[Bailly et al 88] G. BAILLY, G. MURILLO, O. AL DAKKAK, B. GUERIN, 'A text-to-speech system for French using formant synthesis', *SPEECH '88, 7th FASE Symposium*, Edimburgh, U.K., pp. 255-260.

[Bimbot et al 89] F. BIMBOT, G. CHOLLET, P. DELEGLISE, 'Speech synthesis by structured segments, using temporal decomposition and a glottal excitation', *Proc. Eurospeech 89*, Paris, vol. 2, pp. 183-186.

[Boeffard et al 93] O. BOEFFARD, L. MICLET, S. WHITE, 'Automatic Generation of optimized unit dictionaries for text-to-speech synthesis', *Proc. Int. Conf. Spoken Language Processing*, Alberta, 1992, pp. 1211-1214.

[Carlson et al 82] R. CARLSON, B. GRANSTRÖM, S. HUNNICUT, 'A multi-language Text-To-Speech module', *ICASSP 82*, Paris, vol. 3, pp. 1604-1607.

[Ishizaka & Flanagan 72] K. ISHIZAKA, J.L. FLANAGAN, "Synthesis of voiced sounds from a two mass model of the vocal cords", *Bell Systems Technical Journal*, n°51, pp. 1233-1268.

[Holmes 83] J. HOLMES, 'Formant Synthesizers - cascade or parallel ?', *Speech Communication*, Vol 2, 1983, pp. 251-273.

[Holmes et al 64] J. HOLMES, I. MATTINGLY, J. SHEARME, 'Speech synthesis by rule', *Language and Speech*, Vol 7, 1964, pp.127-143

[Klatt 80] D.H. KLATT, 'Software for a cascade /parallel formant synthesizer', *J. Acoust. Soc. AM.*, Vol 67, 1980, pp. 971-995.

[Klatt & Klatt 90] D.H. KLATT & L.C. KLATT, 'Analysis, synthesis, and perception of voice quality variations among female and male talkers', *J. Acoust. Soc. Am.*, Vol 87, 1990, pp. 820-857.

[Kraft & Andrews 92] V. KRAFT, J.R. ANDREWS, 'Design, evaluation, and acquisition of a speech database for German synthesis-by-concatenation', *Proc. SST-92*, Brisbane, pp. 724-729.

[Libermann 59] A.L. LIBERMANN et al., 'Minimal rules for synthesizing speech', *J. Acoust. SOc. Am.*, pp. 1490-1499, 1959.

[Maeda 79] S. MAEDA, , 'An articulatory model of the tongue based on based on a statistical analysis', *J. Acous. Soc. Am.*, vol. 65, 1979, S-22.

[20]Phenomena can be linear or not according to the particular space they are depicted in.

[Mrayati et al 88] M. MRAYATI, R. CARRE, B. GUERIN, 'Distinctive regions and modes : a new theory of speech production', *Speech Communication*, vol. 7, 1988, pp. 257-286.

[Nakajima & Hamada 88] S. NAKAJIMA, H. NAMADA, "Automatic generation of synthesis units based on Context-Oriented Clustering", *Proc. ICASSP 88*, 51.S14.2, pp. 659-662.

[O' Shaughnessy 84] D. O' SHAUGHNESSY, 'Design of a real-time French text-to-speech system', *Speech Communication*, Vol 3, pp. 233-243.

[O' Shaughnessy 90] D. O' SHAUGHNESSY, 'Spectral transitions in rule-based and diphone synthesis', *Proceedings of the ESCA workshop on speech synthesis*, Autrans, 25-28 sept 90, pp. 21-25.

[Pols 90] L. POLS, 'Does improved performance of a rule synthesizer also contribute to more phonetic knowledge ?', *Proceedings of the ESCA tutorial day on speech synthesis*, Autrans, 25 sept 90, pp. 50-54.

[Portele et al 90] T. PORTELE, W. SENDLMEIER, W. HESS, 'HADIFIX, a system for German speech synthesis based on demisyllables, diphones, and suffixes', *Proceedings of the ESCA workshop on speech synthesis*, Autrans, 25-28 sept 90, pp. 161-164.

[Santos & Nombela 82] J.M. SANTOS, J.R. NOMBELA, 'Text-To-Speech conversion in spanish : a complete rule-based system', *ICASSP 82*, Paris, pp. 1593-1596.

[Soquet et al 90] A. SOQUET, M. SAERENS, P. JOSPA, 'Acoustic-articulatory inversion based on a neural controller of a vocal tract model', *Proceedings of the ESCA tutorial day on speech synthesis*, Autrans, 25 sept 90, pp. 71-74.

[Stevens 90] K.N. STEVENS, 'Control parameters for synthesis by rule', *Proceedings of the ESCA tutorial day on speech synthesis*, Autrans, 25 sept 90, pp. 27-37.

[Takeda et al 89] K. TAKEDA, K. ABE, Y. SAGISAKA, H. KUWABARA, 'Adaptive manipulation of non-uniform synthesis units', *Eurospeech 89*, vol. 2, pp. 195-198.

[Tubach & Boë 86] J.P. TUBACH & L.J. BOE, 'Quantitative knowledge on word structure from a phonetic corpus, with application to large vocabularies recognition systems', ICASSP '86, Tokyo, pp. 61-64.

[Van Hemert 91] J.P. VAN HEMERT, "Automatic Segmentation of Speech", *IEEE Trans. on Speech Processing, vol. 39, n°4, 1991, pp. 1008-1012.*

[Wenk & Wioland 82] B.J. WENK, F. WIOLAND, 'Is French really syllable-timed ?', *J. of Phonetics*, vol. 10, pp. 193-216.